Proceedings in
Informatics and Information Technologies

Student Research Conference 2014

Mária Bieliková (Ed.)

Keynote by Jiří Matas

STU
FIIT

Proceedings in
Informatics and Information Technologies

IIT.SRC 2014
Student Research Conference

Mária Bieliková (Ed.)

IIT.SRC 2014: Student Research Conference

10th Student Research Conference
in Informatics and Information Technologies
Bratislava, April 29, 2014
Proceedings

IEEE Czechoslovakia Section

acm Slovakia Chapter

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
Faculty of Informatics and Information Technologies

Mária Bieliková (Ed.)

IIT.SRC 2014: Student Research Conference

10th Student Research Conference
in Informatics and Information Technologies
Bratislava, April 29, 2014
Proceedings

Czechoslovakia Section Slovakia Chapter

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
Faculty of Informatics and Information Technologies

Proceedings in
Informatics and Information Technologies

IIT.SRC 2014
Student Research Conference

Editor

Mária Bieliková
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2
842 16 Bratislava, Slovakia

Visit IIT.SRC on the Web: http://iit-src.stuba.sk/

Executive Editor: Katarína Mršková
Copy Editors: Andrej Fogelton, Tomáš Halagan, Dávid Chalupa, Peter Krátky,
František Kudlačák, Martin Labaj, Ján Laštinec, Dominik Macko,
Aurel Paulovič, Miroslav Siebert, Márius Šajgalík, Lukáš Turský,
Petra Vrablecová, FIIT STU
Cover Designer: Peter Kaminský

Published by:
Nakladateľstvo STU
Vazovova 5, Bratislava, Slovakia

ISBN 978-80-227-4153-8

Preface

This volume contains the keynote abstract and student papers selected for presentation and presented at IIT.SRC 2014, the 10th Student Research Conference in Informatics and Information Technologies, held April 29, 2014 at the Faculty of Informatics and Information Technologies of the Slovak University of Technology in Bratislava.

We included in this volume information on all 89 full papers presented at the conference, 77 of which are included in their full version, 15 extended abstracts, and information on accompanying events. 12 full papers of doctoral students presenting part of their research on ongoing projects were submitted for publication elsewhere and published in this volume in form of one page abstracts. Similarly to previous years when these internal publications in most cases were a first step towards later publishing on national or international established conferences or journals, we suppose that also other papers presented in this volume will evolve into their new (amended/extended) versions and submitted elsewhere.

Research has been one of the main priorities of the university education since its very beginning. It is the case also for our university – the Slovak University of Technology in Bratislava and its faculty – the Faculty of Informatics and Information Technologies. Close connection of research and education leads very naturally to a participation of students in research. This holds not only the students of doctoral study, where research is a substantial part of their study and one of their principal activities. A participation of students in research is now common also for students of master, and even bachelor study.

Universities of technology have a long tradition of students participating in a skilled labour where they have to apply their theoretical knowledge. The best of these results were usually presented at various students' competitions or exhibitions. There were also combined with student research works. Our university has a long tradition in such competition named ŠVOČ (abbreviation of the Student Scientific and Technical Activity). Ten years ago our faculty, FIIT STU, decided to transform former ŠVOČ into the Student Research Conference covering topics of Informatics and Information Technologies (IIT.SRC). Participants are students of all three levels of the study – bachelor (Bc.), master (Ing.) and doctoral (PhD.) study. The conference adopted a form of reviewing as at any other scientific conference, and presenting internally the papers in a form of printed internal Proceedings and this open access e-version of Proceedings.

IIT.SRC 2014 attracted 107 university student papers from which 89 was accepted as full papers (23 bachelor, 51 master, 15 doctoral) and 15 as extended abstracts. The number of papers slightly varies each year. This year we have noticed significant increase in bachelor and master categories and decrease in doctoral category comparing to IIT.SRC 2013.

Full papers are organized in seven sections:

- Intelligent Information Processing,
- Web Science and Engineering.
- Computer Science and Artificial Intelligence,
- Computer Graphics, Multimedia and Computer Vision,
- Computer Networks, Computer Systems and Security,
- Software Engineering,
- Innovative Designs and Applications.

The conference was opened by Jiří Matas followed by a keynote titled *Computer Vision – What Can and Cannot Be Done Yet*. Jiří Matas is currently full professor at the Center for Machine Perception, Czech Technical University in Prague. His main research interests are in computer vision, pattern recognition, image processing and machine learning. In the lecture, he presented selected algorithms that lead to successful applications and demonstrate the recent shift in computer vision towards method relying heavily on machine learning.

Besides the 104 students' projects presented at the conference several accompanying events were organized. The RoboCup Exhibition is organised as a part of IIT.SRC from 2005. RoboCup is an attractive project with free participation, designed to support education and research in artificial intelligence, robotics and information technologies. Through several years, our students achieved interesting results, which were presented during the conference. RoboCup exhibition presented both the way the RoboCup simulated league is played and also the progress of current students' research in this field. Five years ago a new RoboCup league – three-dimensional (3D) robotic simulation was added. The extension of the simulation to the third dimension shows the continuous progress in RoboCup and in our students' skills.

This year we organized for the sixth time as part IIT.SRC a showcase of TP-Cup projects. TP-Cup is a competition of master students' teams aimed at excellence in development information technologies solutions within two semester long team project module. The competition has four stages. 13 teams managed to achieve this stage and presented their projects during the TP-Cup showcase. Extended abstracts of their projects are included in these proceedings.

Accompanying events included for seventh time also our programming contest. It follows a long tradition at the Slovak University of Technology in Bratislava and our faculty in organizing programming contests, especially the ACM International Collegiate Programming Contest like competitions. This year we have organized for the third time the final round of the ProFIIT programming contest for high school students in parallel with IIT.SRC. Our aim was to show our potential future students exciting research opportunities awaiting them at our university.

We continued this year with FIITApixel exhibition. FIITApixel brings together both students and staff of the Faculty as well as its potential students and alumni in an effort to create, share and judge pictures. It is organized as an ongoing event, where anyone can contribute pictures. The IIT.SRC FIITApixel exhibition presented the best pictures of this year contest.

For the third time we organized this year Junior IIT.SRC. It provides a room for presenting inventive high school student projects within the topics of the conference. Three high school students' submissions were selected. All these projects are also presented as extended abstracts for more detailed explanation of proposed ideas and realized prototypes in these proceedings.

The student research conference is the result of considerable effort by a number of people. It is our pleasure to express our thanks to:

- the IIT.SRC 2014 Programme Committee who devoted effort to reviewing papers and awards selection,
- the IIT.SRC 2014 Organising Committee and accompanying events coordinators (mentioned in particular reports in these proceedings) for a smooth preparation of the event,
- the students – authors of the papers, for contributing good papers reporting their research and their supervisors for bringing the students to research community.

Special thanks go to Katarína Mršková and doctoral students who did an excellent job in the completion of the proceedings, Zuzana Marušincová and the whole organizing committee for effective support of all activities and in making the conference happen. Finally we highly appreciate the financial support of our sponsors which helped the organizers to provide excellent environment for presentation of the results of student research and valuable awards.

Bratislava, April 2014

Pavel Čičák and Mária Bieliková

Conference Organisation

The 10th Student Research Conference in Informatics and Information Technologies (IIT.SRC), held on April 29, 2014 in Bratislava, was organised by the Slovak University of Technology (and, in particular, its Faculty of Informatics and Information Technologies) in Bratislava.

General Chair

Pavel Čičák (dean, Faculty of Informatics and Information Technologies,
Slovak University of Technology in Bratislava)

Programme Chair

Mária Bieliková

Programme Committee

Michal Barla
Vanda Benešová
Anna Bou Ezzeddine
Michal Čerňanský
Peter Drahoš
Elena Gramatová
Ladislav Hudec
Daniela Chudá
Katarína Jelemenská
Peter Kapec
Michal Kompan
Gabriela Kosková
Margaréta Kotočová
Ivan Kotuliak
Tomáš Kováčik
Alena Kovárová
Tibor Krajčovič
Vladimír Kvasnička
Peter Lacko
Michal Laclavík
Ján Lang
Marián Lekavý
Mária Lucká
Peter Magula
Ľudovít Molnár
Pavol Návrat
Ivan Polášek
Jiří Pospíchal
Viera Rozinajová
Petr Šaloun
Jakub Šimko
Marián Šimko
Juraj Štefanovič
Jozef Tvarožek
Valentino Vranić

Organising Committee

Alexandra Bieleková
Mária Bieliková
Ivan Kotuliak
Zuzana Marušincová, *Chair*
Michal Michel
Katarína Mršková
Ľubica Palatinusová
Branislav Steinmüller
Roman Stovíček

all from FIIT STU in Bratislava, Slovakia

General Sponsors

– Capco Slovakia
– Hewlett-Packard Slovakia
– IBM Slovakia

Financial Sponsors

- Regional Card Processing Centre
- Softec
- Soimco
- Tempest

Supporting Professional Societies and Foundations

- ACM Slovakia Chapter
- Czechoslovakia Section of IEEE
- Slovak Society for Computer Science
- Informatics Development Foundation at FIIT STU

Medial Partner

- PC Revue

Accompanying Events Sponsors and Medial Partners – TP CUP

- Accenture
- Ditec
- Enprovia
- QBSW
- PosAm
- Softec
- Tempest
- Unicorn
- PC Revue
- robime.it

Accompanying Events Sponsors – FIITAPIXEL

- Fotolab

Table of Contents

Keynote

Intelligent Information Processing

Bachelor Degree Program Students

Master Degree Program Students

Doctoral Degree Program Students

Web Science and Engineering

Bachelor Degree Program Students

Master Degree Program Students

Doctoral Degree Program Students

Computer Science and Artificial Intelligence

Bachelor Degree Program Students

Master Degree Program Students

Doctoral Degree Program Students

Computer Graphics, Multimedia and Computer Vision

Bachelor Degree Program Students

Master Degree Program Students

Computer Networks, Computer Systems and Security

Bachelor Degree Program Students

Master Degree Program Students

Doctoral Degree Program Students

Software Engineering

Bachelor Degree Program Students

Master Degree Program Students

Doctoral Degree Program Students

Innovative Designs and Applications

Bachelor Degree Program Students

Master Degree Program Students

Extended Abstracts

Accompanying Events

Computer Vision – What Can and Cannot Be Done Yet

Jiří MATAS

Czech Technical University in Prague
Center for Machine Perception
Technická 2, 166 27 Prague, Czech Republic
matas@fel.cvut.cz

Abstract. In the last 20 years, computer vision has been transformed from a purely theoretical discipline producing algorithms operating well on a few images hand-picked by their authors to a field where the path from a conference paper to a start-up company is very short. Companies like Google, Microsoft, Adobe and Amazon employ dozens or even hundreds of computer vision specialists. In the lecture, I will present selected algorithms that lead to successful applications and demonstrate the recent shift in computer vision towards method relying heavily on machine learning. Examples include the Viola-Jones sliding window object detection and large-scale image retrieval based on the Bag-of-words method. Finally, I will discuss open problems like categorization and long-term object tracking.

Intelligent Information Processing

Dynamic Score as a Mean for Motivation of Students in an Educational System

Richard FILIPČÍK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`richard@filipcik.sk`

Abstract. While motivating students to learn, it is important to employ appropriate method for student activities evaluation and correspondent presentation of this evaluation in an educational system. We propose in this paper a method for dynamic regulating score value. Score computation based on teacher's decisions on priority of particular activities and by the student actual activities. It is important to provide effective mean for score regulation to gain expected activities from students and thus support learning process. Our method aims to motivate students to perform all preferred activities equally considering whole group of students. To keep attention and activity of the student we propose personalized stream of announcements on actual state of activities performed by the whole class and on the value of particular activities in time. We integrated the method into Adaptive Learning Framework ALEF. The stream is presented on Facebook to keep students aware of their educational activities. We present an analysis of historical data which allowed to set up parameters for our score computation method.

1 Introduction

There are many ways on how to motivate student during the use of web-based learning system. It is score and pointing systems which belong in most frequent and used ways to support motivation. Most educational systems use various methods of score computation – from the simplest which just count amount of performed activity to the advanced which take many factors into account. However, there are much more ways to classify pointing systems. Some systems do not change their „conventions“ and same activities rate by same amount of points each time. In other words, this kind of systems use static way of score computation. On the other side, some systems take user activity into account, change „weights“ of activities on the basis of it, so they compute scores following current system status. This is way how users behaviour can affect score computation.

In this paper we introduce dynamic score computation method based on users activity, so system can modify „weights“ of particular activities and induce students to perform all of them more or less equally. We are also making an effort to implement this method into learning system

[*] Bachelor degree study programme in field: Informatics
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

ALEF [5]. By this method we aim at increasing students‘ motivation thanks to smarter score computation.

The paper is structured as follows. First we discuss related work. We concentrate on existing methods for score computation and systems using it. In the chapter 3 we propose our own method for dynamic score computation we've been implementing into educational system ALEF. Experiment consisting of two phases is described in the Evaluation chapter. The content of this paper is concluded in chapter 5.

2 Related work

The idea of a score as a part of pointing system is not new. It is often used in various educational systems, but also in systems not primarily focused on learning. Even though with different names such as reputation or karma, its main goal is always the same. The general use of score is to motivate a user of a system [7], therefore in case of educational systems to motivate a student. However, it also may serve as a way for collaboration between system users/students [4].

Score or reputation, method for its calculation, and way of its use can take many forms [1,7]. Probably the most common form of use is to receive points for actions performed in the system, while the total amount of points received for the same type of action is constant and it does not change over time. This type of score has common use in many systems. One of the most popular non-educational system with that score type is Stack Overflow[1]. Here users, mostly programmers, get reputation points for various activities, such as answering a question or receiving positive feedback for their comments. Another example is Feedback Forum [1] used by popular auction site eBay[2]. On eBay, each successful transaction is rewarded by positive points for seller by reputation system.

In [2] authors described an experiment about impact of using negative types of "rewards". Experiment showed that possibility of receiving a negative badge motivated students more than possibility of receiving positive ones. This could also be easily applicable on score ratings. Threat of score losing can be interesting option in some cases.

An example of more advanced method for score calculation may be exercise sharing tool SOCIALX [3,4]. Here, a reputation earned by student is used to determine the weight of actions performed by a student. To calculate actual student reputation there are also several facets affecting it, such as usefulness and competency.

Educational system ALEF previously used a simpler method for score calculation, which took only number of activities performed into account, but it did not count their relative ratio. Weights of these activities were set to constant values and did not change over time. As a student activity any action performed by the student on a learning object (questions, exercises) or metadata (tags, comments) is considered. Each activity which is also connected to the ALEF domain model through metadata defined on particular learning objects [5].

Systems mentioned above propose various ways on how to calculate score and how to use it. We consider as important source for score regulation activity of the group of students as a whole based on preferred activities instead of considering only individuals in computing score. In such case the system would automatically regulate the score and appropriately inform the students about the changes.

3 Dynamic score computation in web-based learning

We proposed a new method for score regulation which takes into account current system status and changes it is affected by. This method relies on several factors important for score regulation.

[1] http://www.stackoverflow.com/
[2] http://www.ebay.com/

While some of them are constant and do not change over time, others are not and they can vary depending on the actions being performed.

- activity weight, which is value estimating effort necessary to perform this activity,
- activity preference set by the teacher, defining how important is the activity in the certain time period and
- activity priority, computed by the system itself, it is based on current status of activity of all students that is amount of activities performed and their relative ratio.

Activity weight is a constant value representing difficulty of the activity to perform for an average student. It is predefined in the system and it does not change over time. When determining weights, we first sorted activities by difficulty following opinions of the expert (teacher). Then we adjusted them considering historical data. By this, we showed, for instance, that text highlighting is less difficult than searching and inserting new external source. Based on the historical data of the use of ALEF we can compare amount of activities performed and thus determine their approximate difficulty, since we can expect less difficult activities are performed more frequently by the students than more difficult ones.

Activity preference is based on teacher requirements for students in a given period of time and it usually depends on the current requirements for learning process. Teacher can set which activities will be preferred and the level of the particular activity preference as well. The preference level is represented by a number within the range expressed by select list, where higher number means higher level of preference.

Activity priority is a dynamic factor and it is computed exclusively by the system itself, with no options to modify it from the outside. Priority depends mainly on the amount of activities performed by students, taking activity weight and activity preference into account as well. Based on the weights of activities we can expect relative ratio between these activities which should be present in the system. Activity priority expresses the relationship between the expected ratio and the actual ratio. The higher actual ratio is than expected, the lower score addition for the current activity will student receive by performing it and vice versa.

Proposed method calculates and regulates score for the student s at time t following expression below:

$$score(s,t) = score(s,t-1) + \sum_i partial\_score(C_i,s,t) \tag{1}$$

where *score(s,t)* is score regulation function returning score for the student *s* at time *t*. We can see it sums up student's previous score and score additions for all activities performed at time *t*.

The value of score addition for the particular activity is computed by *partial_score*(C_i,*s*,*t*) function which takes three arguments - activity C_i, student *s* and time *t*. It can be expressed as below:

$$partial\_score(c,s,t) = weight(c) * pref(c,t) * prior(c,t) * add(c,s,t) \tag{2}$$

where *weight(c)* is the weight of the current activity, *pref(c,t)* is activity preference at time *t*, *prior(c,t)* is activity priority at time *t* and *add(c,s,t)* is regular score addition computed as difference between activity score at time *t* and time *t-1*. The amount of score addition returned by *add(c,s,t)* is also logarithm-based so additions are decreasing in time. Thus, we gain amount of score addition by multiplication of activity weight, preference, priority and regular addition.

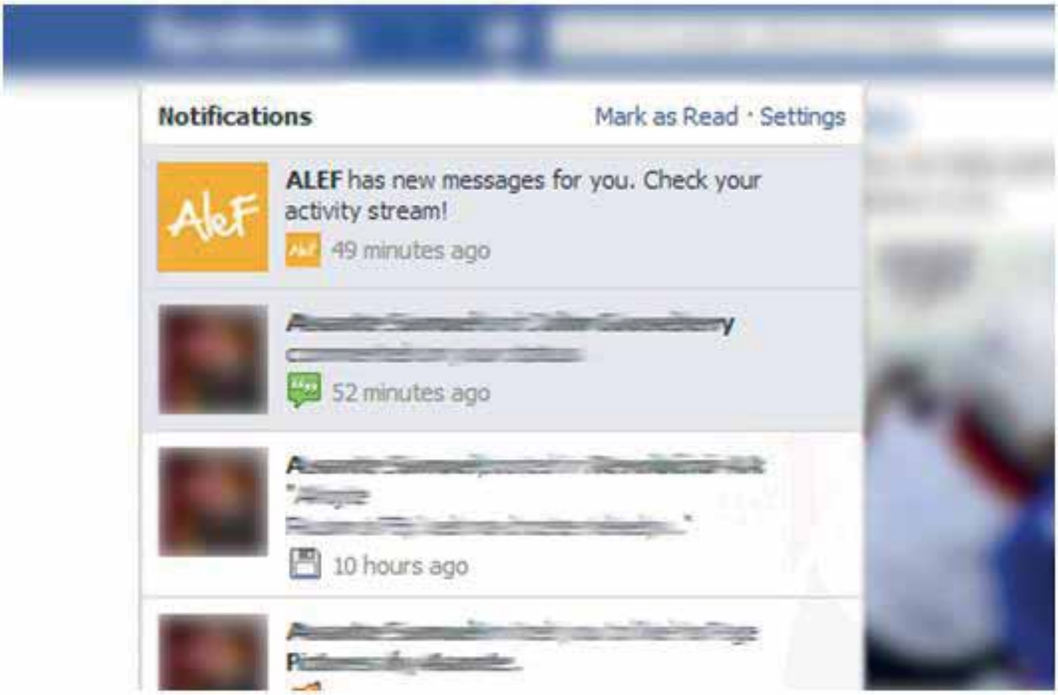
However, it should be noted that this algorithm is applicable only on activities which have preference level set by a teacher. In the case teacher is not preferring (with any level of preference) particular activity, the amount of the score addition will be the same as it would be if there was no dynamic score regulation.

In the algorithm presented above, we can expect the amount of particular activity performed by all students can occasionally be much higher (or lower) than other activities. That would lead to score addition overflow because of too high activity priority computed by the system. For this case, we have set interval limiting activity priority level. By implementing this interval, we can assure the priority will never be higher as 1.5 times of its regular value as well as it will never be lower 0.5 times of its regular value.

To ensure students can observe changes in activity priorities and thus they are motivated by this changes to perform activities with increased priority value, we have also implemented time interval to handle priority computation actions. With this limit changes in activity priorities are not so smooth and are easily visible in the eye of the student.

To give the student feedback from the system to know when and how factors affecting score regulation change, part of our method is also a feature called activity stream, which is similar to news streams popular on many social networks. In the stream student can see messages about preference or priority changes. To avoid case stream is overloaded with many similar messages, system regulates content and time interval of generated messages and also provide simple personalization.

To bring messages from the stream right to the student, there is an option to connect his ALEF account with account on Facebook, so he/she can see what's new in the notifications bar. Thanks to rising popularity of various social networks it is likely that many students have their own accounts too. Bringing ALEF stream messages into Facebook notifications bar should motivate student to use ALEF more frequently. Notifications visualization is shown in Figure 1.



ALEF has new messages for you. Check your activity stream!

49 minutes ago

52 minutes ago

10 hours ago

Figure 1. Visualization of ALEF message in Facebook notifications bar.

4 Evaluation

We have integrated our method for dynamic score regulation into the environment of Adaptive Learning Framework ALEF[3]. ALEF is being developed on Faculty of Informatics and Information Technology of Slovak University of Technology in Bratislava and is used on multiple subjects as

[3] http://alef.fiit.stuba.sk/

teaching support, so it is used by dozens of students during a semester. ALEF was previously using simple method for score calculation based on points accumulation when any activity was performed. By implementing our method we want to achieve more balanced relationship between activities and motivate students to use ALEF more frequently while learning.

In order to set initial weights for particular activities we analysed students' activity in last two years. We used analysed data so weights now reflect effort necessary to perform the activity. The more effort is needed, the bigger weight the related activity has. Weight values can be found in Table 1. Each activity category is also divided into particular subcategories with their own weights.

Table 1. Weights of particular activities used by ALEF.

	Activity category	Activity weight
1	Bug reports	1.0
2	Comments	1.0
3	Highlights	0.5
4	Exercises	1.0
5	Explanations	1.0
6	External sources	1.0
7	Questions	1.0
8	Tags	1.5
9	Summaries	0.5

We plan an experiment consisting of two phases. In the first phase we will observe the behaviour of students when there will be no explicit feedback from the system about activity priority nor activity preference changes. Thus students will use system just as before, only method for score regulation will be different. By evaluation of this phase we will determine impact of new score regulation to student decision which activity perform as next.

The second phase will continue with new score regulation method, however, this time students will be explicitly informed about changes in preference or priority. This will be achieved by activity stream already mentioned above. By messages, students will be informed not only about preference/priority changes, but the size of this changes as well, so they will know which activity is worth to perform at the moment and which not. Thanks to simple message personalization student is informed about other preferred activities if he/she continue to perform only one type of activity.

We will compare results obtained from the both phases of experiment with each other and will determine how decisive was providing feedback by activity stream for students decisions on which activity to perform as next. By comparing this data with historical ones we will also try to estimate whether our method provided more balanced relationship between different activities.

Our evaluation will also be integrated with some more experiments based on ALEF which are aiming on external source adding and question-answer learning objects [6].

5 Conclusion

The aim of the method proposed by us is to make score computation and regulation more dynamic and perhaps fairer, and to achieve the ratio between activities performed in the system to be more balanced. At the same time we want to use it to motivate students to use learning system as a support for their studies more frequently. The key in our method is an algorithm that uses three different factors affecting the score - activity weight, activity preference and activity priority. While for the activity preference is important what teacher need at the moment, activity priority value depends only on system calculations. Priority activity is growing in parallel with reducing its ratio in relation to other activities performed by the students and vice versa.

However, the aim of our method is not only to propose a new method for score calculation, but to get this calculation closer to the student too. It is important for score calculation to be transparent so student knows reasons which activity do he or she receives new points for and why he/she receives that amount of points. For this purpose we have designed a feature called activity stream, which informs student about activity preference and activity priority changes via short messages. We plan to evaluate effectiveness of our method by the experiment consisting of two phases. In the first phase the student uses system as before, however with new score regulation method implemented. Student has no feedback from the system about changes in activity preference nor activity priority. This feedback is introduced in the second phase of the experiment via activity stream. The experiment will take place in the ALEF learning system on the Principles of Software Engineering and Functional and Logic Programming courses.

Acknowledgement: This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. 009STU-4/2014.

References

[1] Resnick, P., Ko, K., Zeckhauser, R., Friedman, E.: Reputation systems. In: *Communications of the ACM*. ACM, New York, NY, USA, 2000. pp. 45-48

[2] Santos, J. L., Charleer, S., Parra, G., Klerkx, J., Duval, E., Verbert, K.: Evaluating the Use of Open Badges in an Open Learning Environment. In: *Scaling up Learning for Sustained Impact*, Springer Berlin Heidelberg, 2013. pp. 314-327

[3] Sterbini, A., Temperini, M.: Learning from peers: motivating students through reputation systems. In: *International Symposium on Applications and the Internet, Social and Personal Computing for Web-Supported Learning Communities (SPeL)*. Turku, Finland, 2008. pp. 305-308

[4] Sterbini, A., Temperini, M.: Social Exchange and Collaboration in a Reputation-Based Educational System. In: *Proceedings of 9th International Conference of Information Technology Based Higher Education and Training (ITHET)*. Cappadocia, Turkey, 2010. pp. 201-207

[5] Šimko, M., Barla, M., Bieliková, M.: ALEF: A framework for adaptive web-based learning 2.0. In: *Key Competencies in the Knowledge Society* [online]. Springer Berlin Heidelberg, Berlin, Germany, 2010. pp. 367-378.

[6] Šimko, J., Šimko, M., Bieliková, M., Ševcech, J., Burger, R.: Classsourcing: Crowd-Based Validation of Question-Answer Learning Objects. In: *5th Int. Conf. of Computational Collective Intelligence Technologies and Applications*, ICCCI 2013, Springer LNAI. Vol. 8083, pp. 62-71, 2013

[7] Zichermann, G., Cunningham, C.: Gamification by design : implementing game mechanics in web and mobile apps. Sebastopol, Calif: O'Reilly Media, 2011, pp. 35-66.

Enhancing Keyword Map Visualisation for Educational Content Management

Matej KLOSKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
matej.kloska@gmail.com

Abstract. In present, visualisation of data is becoming more and more commonly used. We want to focus on a special type of visualisation – keyword map visualisation. Any type of a document could be described by a set of keywords and relationships between them, which can be easily interpreted as a graph or a keyword map. This paper presents a method for keyword map visualisation for educational system ALEF based on an analysis of existing solutions in the field of graph visualisation and navigation. The method is later on evaluated via live experiments.

1 Introduction

Nowadays, people are creating more digital documents – data elements – than ever before. If we want to search and navigate in those documents, we need efficient way how to interpret relations between documents. Information visualisation [6] has become a large field and various “subfields” begin to emerge. The question is, whether there is an inherent relation among the data elements to be visualised. Proper relations creation and information visualisation implies high success rate in looking for desired information in documents.

If we work with a large number of data elements, we often need efficient representation of content – e.g., to describe each document with an appropriate set of keywords. Keyword sets alone do not guarantee quality of maps, i.e. the ease of navigation in information space. Quality in this sense strongly depends on interconnections of keywords between documents. There are several techniques how to create keyword connections. Most of them are based on clustering techniques.

Another problem is, how to properly visualise keywords and relations between them. The importance to support users and provide adequate user experience is also crucial problem, which we have to keep in mind. The more fail proof interface, the more valuable maps.

We propose a method for keyword map visualisation for educational system ALEF with support of content managing system COME^{2}T (Collaboration-and Metadata-oriented Content Management EnvironmenT; see Figure 1). Our method will be implemented and evaluated using existing system COME^{2}T [1]. The COME^{2}T allows easy administration of lightweight semantics

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 9–14.

for the provided content – digital documents and user-created annotations. It was designed as is being used a content management system for educational system ALEF.

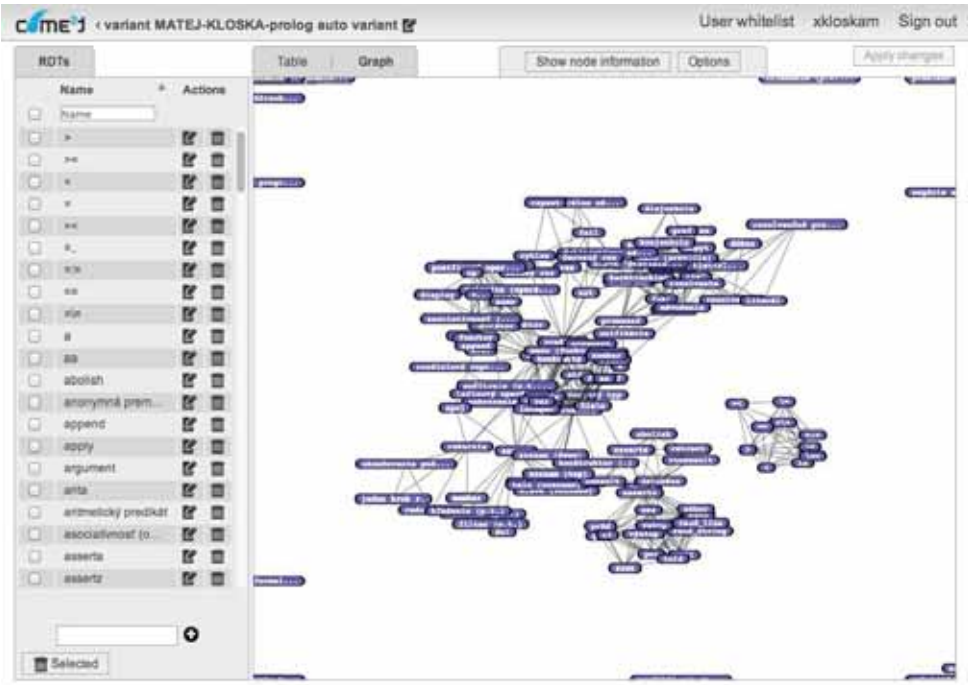


Figure 1. Screenshot of COME^{2}T showing keyword map of existing data from "Prolog course" [7].

2 Related Work

To our best knowledge, we are not aware of many similar works to our method except for method, which is already implemented in system COME^{2}T. Complexity of method leads to many approaches to solve specific parts of method such as layouting, clustering, navigation and creation of graphs. In last three decades, there were many surveys [9] and case studies [10] in field of information visualisation focusing on general techniques of visualisation. With respect to our point of interest, there were also made more specific works focused on graph and domain visualisation techniques. Study of algorithms for drawing graphs [11] provides comprehensive list of well-known algorithms and techniques. Introduction to typical application areas and navigation in information with respect to graph visualisation is discussed in a survey with proposed solutions to navigation-interaction, layouting and clustering [2,15]. Aesthetic visualisation of information as key issue with proposed optimizations of graph representations is mentioned in recent work of Meiller [12]. Clustering with existing proposals – structural and semantics clustering - was mentioned in many works with pros and cons of each one [13,14]. Other work related to visualising knowledge domains by Border [8].

Creating and maintaining adaptive educational applications is a hard work for teachers and developers. In order to help the author to perform these tasks the e-learning systems must provide authoring and management tools. One of existing solutions in this field is system AHA! [16] Other solution for managing e-learning content, metadata and social annotations is COME^{2}T [1]. AHA! Graph Author tool provides easy to use interface for creation of concepts and concepts relations. Functionality of this tool in comparison to COME^{2}T has in our opinion advantage for user in hierarchical representation of concepts. COME^{2}T has advantages over AHA! Graph Author in administration of relation types and graph visualisation.

In the context of graph visualisation and system COME^{2}T, we have also looked at existing solutions of client libraries providing API for data manipulation, transformation and visualisation in web browsers. *JavaScript InfoVis Toolkit*[1] is easy to use monolithic library for interactive data visualisations. D3.js[2] is library primarily designed for manipulating documents based on data. yWorks[3]as example of commercial solutions represents solution for data visualisation on various platforms including client web browsers, server and desktop applications.

It is obvious from Table 1, all libraries provide almost the same functionality with slightly different additional functionalities. D3.js is cost-free and compared with yWorks HTML has huge community of users with various plugins and modifications ready to use for developers. Library has well documented[4] API with many tutorials and examples. JIT has almost the same features like D3.js, but it is not maintained on regular basis. JIT was previously used as default library for visualisation in COME^{2}T.

All mentioned works discuss different approaches to data visualisation on different levels of abstraction from general visualisation to specific domain visualisation. State of the art in educational content management is for us COME^{2}T.

Table 1. Comparison of key attributes of selected client libraries.

Feature	**JIT**	**D3.js**	**yWorks HTML**
Custom nodes	yes	yes	yes
Edges operations	create/delete	create/delete	create/delete
Labels	yes	yes (plugin)	yes
Tooltips	yes	yes (plugin)	yes
Context menu	no / only tooltip	yes	yes
Pan	mouse drag	mouse drag	mouse drag/scrollbar
Zoom	yes	yes	yes
Selection	no	no (plugin)	yes
Morphing the graph into another one	yes	no	no
License	free	free	commercial

3 Keyword map for domain conceptualization visualization

Based on the analysis of several participants habits during keyword relationship authoring in previous experiments with content management system COME^{2}T [7], we have identified several key issues necessary to involve in the design of our method. They can be divided into two main types of related issues:

- *graph structure visualization*: our goal is to adjust algebraic properties such as density of graph and graph size,
- *navigation support*: our goal is to support user experience in navigation by adding several navigational upgrades.

[1] http://philogb.github.io/jit/

[2] http://d3js.org/

[3] http://www.yworks.com/en/products_yfileshtml_about.html

[4] https://github.com/mbostock/d3/wiki

3.1 Graph structure visualisation

Graph structure highly affects visual quality of output map. Basically, there are two key issues targeted to a structure. First of them is *graph size*. The higher the count of vertices in map, the more difficult to work with such graph. To solve this issue, we introduce structural clustering in order to virtually reduce number of visible vertices on screen (see Figure 2). Structural clustering is more suitable in comparison to traditional semantics clustering that is based on clustering of keywords' semantics. In other words, we take advantage of the keywords and explicit relationship between them (not necessarily semantic) and do not consider other non-explicit relationships (e.g., similarity of keywords that can be induced from keyword-document associations). The reason of this benefit results from no need to acquire knowledge base of domain related to input data. Semantic clustering is more sophisticated method. However, it is tightly coupled to knowledge base therefore method is less variable [2].

In the context of clustering it is important to mention fundamental approaches to visualisation and behaviour of clusters:

– *grouping*: representation of clusters with more general superior node,

– *ghosting*: emphasizing of selected node with close neighbourhood, all other nodes fade out,

– *hiding:* similarity to ghosting, the difference is in behaviour of not related nodes - they are completely hidden [2].

Our method is be based on combination of grouping and ghosting. In grouping, superior node will be represented by three or four keywords with the most outer-going edges. Our goal is to make graph more "user-friendly" by reducing number of visible graph elements on screen.

Creation of relations between cluster and node or two clusters is not allowed in our method. Relations are defined only for pairs of keyword nodes. Clusters' outgoing relations are calculated from relations between nodes encountered in cluster and nodes from the outer space. If node is moved out of cluster, relations are recalculated according to remaining nodes inside cluster. Map authoring is also supported by automatic re-layouting of graph's visualisation on user decision.

Second issue related to graph structure is *graph density*. Dense graph is a graph with the number of edges close to the maximum number of edges. On the other hand, sparse graph is the opposite – low number of edges. The higher the number of graph edges the lower the readability for the user. We propose two solutions. The first solution is to apply clustering with an appropriate type of zooming. The other one is based on idea of spanning trees [4]. Before constructing a spanning tree, we traverse graph and analyse all nodes in order to define exploration function [5] based on the structure of the graph. Considering node and edge types in exploration function would result in more suitable spanning trees than general maximum or minimum exploration function [4].

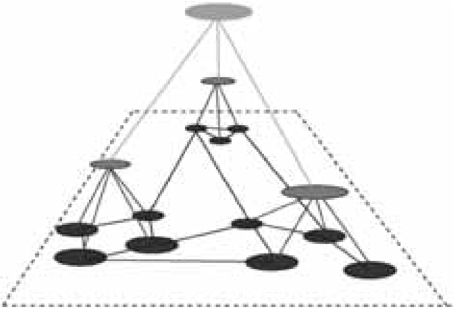


Figure 2. The principle of clustering (hierarchical) [17].

3.2 Graph navigation support

We have identified three features that would improve user experience and make navigation easier:

- keyword map overview,
- navigation bar,
- colouring of nodes and edges.

The keyword map overview will help user to know where in map (s)he exactly is. It would be particularly useful in the case of zoom and pan where only a small part of map is shown on screen.

Currently, there is no other simple way how to provide user simple feedback on position in map in COME^{2}T, especially when clustering is applied. Navigation bar would enhance track of navigation in map when user wade in any cluster. It is similar to well-known address bar in web browsers or file explorers. In our case, bar provides information about users' map exploration path with respect to clusters hierarchy.

The third proposed feature is colouring of map elements. Visual feedback is key feature for every user. We would like to provide interface for custom colouring of nodes and edges in addition to automatic one based on defined rules and analysis of graph. Automatic colouring is easy to implement and use because of predefined edges types.

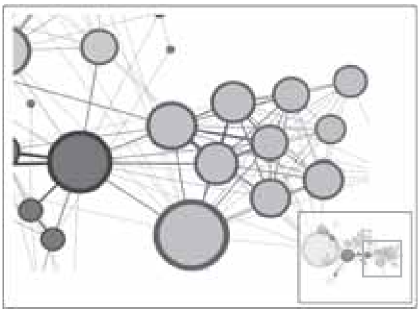


Figure 3. Example of keyword map overview in bottom right corner. Taken from Gephi[5].

4 Evaluation

Since we are still in a phase of implementing the proposed utilities to the system COME^{2}T, we here describe the evaluation plan. Our method would be evaluated in several phases.

In first phase we will evaluate the impact of user interface and visualization library changes on time required for creation defined maps for provided courses. We expect, that time required for creation will be smaller than for the early method.

In second we will evaluate the impact of supporting features like navigation bar and keyword map overview when a user comes back to previously created maps. We expect that supporting features will help a user orientate in map easier and recognize work already done.

Final evaluation of method will be evaluated using the same tasks as in evaluation [7] made before implementation of our method.

5 Conclusion

In this paper we proposed a method for enhanced keyword map visualisation. We proposed several advancements in visualisation and user interface. All of them are based on research in field of

[5] https://gephi.org/

graph visualisation and navigation. Our visualisation and navigation method is currently being implemented in content management system COME^{2}T and will be evaluated against existing method via live experiments in creating and managing keyword maps (as a form of conceptual metadata) for educational system ALEF.

Acknowledgement. This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Franta, M., Habdák, M., Šimko, M. et al.: Managing content, metadata and user-created annotations in web-based applications. In: *DocEng '13*, ACM, (2013). pp. 201.

[2] Herman, I., Society, I. C., Melanc, G. et al.: Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, (2000), vol. 6, no. 1, pp. 24-43.

[3] Bastian, M., Heymann, S.: Gephi: An Open Source Software for Exploring and Manipulating Networks. In*: ICWSM,* (2009), pp. 361-362.

[4] Graham, R. L.: On the History of the Minimum Spainning Tree Problem. *Annals of the History of Computing*, (1985), vol. 7, no. 1, pp. 43-57.

[5] Fisher, D., Dhamija, R., Hearst, M.: Animated exploration of dynamic graphs with radial layout. In: *INFOVIS '01*, IEEE Computer Society, Washington, (2001), pp. 43-50.

[6] Schulz, H.-J., Schumann, H.: Visualizing Graphs - A Generalized View. In: *IV'06*, (2006), pp. 166-173.

[7] Vrablecová, P.: Experiment with educational content authoring. Technical report. Slovak university of Technology in Bratislava, (2013).

[8] Borner, K., Chen, C., Boyack, K. W.: Visualizing Knowledge Domains. *Annual Review of Information Science & Technology*, vol. 37, no. 1, (2003), pp. 179-255.

[9] S. K. Card: Visualizing retrieved information: A Survey. *IEEE Computer Graphics and Applications*, (1996), vol. 16, pp. 63-67.

[10] Card, S. K., MacKinlay, J.: The Structure of the information visualization design space. In: *IV 1997*, (1997), pp. 92-99.

[11] Battista, G. Di, La, R., Salaria, V., Tamassia et al.: Algorithms for Drawing Graphs: an Annotated Bibliography. *Comput. Geom. Theory Appl.*, (1994), vol. 4, no. 5, pp. 235-282.

[12] Meiller, D., Hemmje M., Klas, C.: Aesthetic Visualisation of Information: Optimization of Graph Representations. In: *AVI '12*, (2012), pp. 653-656.

[13] Du, K.-L.: Clustering: a neural network approach. *Neural Networks: The Official J. of the Int. Neural Network Society*, (2010) vol. 23, no. 1, pp. 89-107.

[14] Rástočný, K., Tvarožek, M., Bieliková, M.: Web Search Results Exploration via Cluster-Based Views and Zoom-Based Navigation. *J.UCS*, (2013), vol. 19, no. 15, pp. 2320-2346.

[15] Frishman, Y., Ayellet, T.: Dynamic drawing of clustered graphs. In: *INFOVIS 2004*, (2004), pp. 191-198.

[16] De Bra P., et al.: Authoring and management tools for adaptive educational hypermedia systems: The AHA! Case study. In: *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, Springer, Berlin Heidelberg, (2007), pp. 285-308.

[17] Eades, P.; Fend, Qing-Wen. Multilevel visualisation of clustered graphs. In: *Graph drawing*, Springer, Berlin Heidelberg, (1997), pp. 101-112.

Word Sense Disambiguation Targeting Slovak Language

Jaroslav LOEBL*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`jaroslavloebl@gmail.com`

Abstract. Word sense disambiguation (WSD) is an important part of natural language processing. There are many approaches to maximize the precision of disambiguation process. However, most of them are focused primarily on English language with use of English lexical resources such as WordNet. In this paper we present unsupervised method for WSD, which makes use of Kratky Slovník Slovenského jazyka to obtain sense definitions and to set level of granularity, and of Slovnik Slovenského Jazyka and Wikipedia to enrich contextual information. In case of Wikipedia we also use keywords extracted from English version of an article translated by the web service. The disambiguation algorithm itself is based on the Lesk algorithm counting overlaps between context of a to-be-disambiguated word and glosses obtained from three abovementioned resources.

1 Introduction

Internet is overflowing with documents and text written down in natural language. This means a great amount of data and information that are unstructured. During the information retrieval process, it is often important to understand the content. The understanding includes disambiguation of polysemous words.

Word sense disambiguation is a process of marking a polysemous word with correct sense tag, in other words, the ability to tell which sense of polysemous word is used in a given context. While this process is almost automatic for humans, in computer analysis of natural language it is not straightforward at all, yet very important for many disciplines, including machine translation, part-of-speech tagging, content analysis and basically any other field which requires deep understanding of text.

There is a lot of effort going on in word sense disambiguation research. However, most of it is focused primarily on English language. This is natural, as it is widely used around the world and offers large amount of knowledge resources, such as machine readable dictionaries or annotated corpora with semantic relations between words. We focus on Slovak language and its resources, as there is no publicly available service nor methodology (as we do not have many required resources, which are for English language considered as basic) offering any kind of word sense

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

disambiguation. We believe it can be helpful in some end-point applications, for example machine translation, as stated in [1] where disambiguating word senses improved the translation from 29.73 to 30.30 (BLEU score), representing a significant improvement.

In the rest of the paper we provide short overview of basic approaches and existing methods and focus on our own dictionary-based algorithm with context enrichment from Wikipedia.

2 Related work

There are four main approaches to word sense disambiguation:

1. Supervised – methods that are dependent on sense tagged corpora, which is usually manually created. Disambiguation rules are extracted from these corpora in learning phase by one of the machine learning algorithm (Naive-Bayes, Support Vector Machines and many others). These rules are then applied to untagged word in unannotated corpus, to disambiguate the correct meaning of the polysemous word. As far as precision goes, these are the best performing algorithms, when they significantly outperform most frequent sense baseline as shown in [2] where LazyBoosting algorithm accuracy is by 17.89 higher than most frequent sense baseline. State-of-art algorithms perform even higher, for example 88 percent recall in [3]. However, crafting manually sense tagged corpora is expensive and very time consuming process and where there is no such corpus available, these methods are useless. Corpora are also often aimed on certain domain.
2. Semi-supervised – to avoid bottleneck caused by the need of a large dataset tagged manually, David Yarowski proposed an algorithm, which worked with only small amount of tagged data [4]. Then, the rest of the polysemous words are tagged in a bootstrapping process, until a whole corpus is tagged or algorithm stops at stable number of untagged words. Even though the human participation is greatly reduced, it is still needed for learning phase.
3. Knowledge-based - methods that relies on dictionaries, thesauri and other lexical resources and word disambiguation is based on context similarity. This approach was firstly presented by Michael Lesk [5]. Basic lexical source, which is used in almost every knowledge-based algorithm, is WordNet. These methods do not require sense tagged corpora to learn from and can be applied to any domain.
4. Unsupervised - also called Word Sense Induction (or Discrimination) - methods that works with context of ambiguous word only. Word senses are inducted directly from unannotated corpora and words are clustered into groups which represent one meaning of word [6]. These methods have the potential to overcome the knowledge acquisition bottleneck.

Since we want our service to be available as a public web service which assigns a sense label to a polysemous word, the process of disambiguation has to be fully automatic. Therefore the most suitable are knowledge-based methods, as they do not rely on any kind of human participation. As mentioned above, the main source of senses for English is WordNet, which has been the main source of fine grained word senses for about 20 years [7]. A question could be raised, whether WordNet glosses (gloss - short definition of particular sense of the word) provide enough contextual information to match the correct sense in target context. Ponzetto and Navigli proposed methodology to overcome this problem by extending WordNet with large amount of semantic relations from Wikipedia [8]. This approach has also achieved performance comparable to supervised systems – about 81.7 % precision on SemEval-2007 task 07 dataset[1] (comparing to 83.2% of best performing supervised algorithm).

Another approach was presented by Chen, Ding, Boews and Brown [9] - use of popular search engines (namely Google and Yahoo) to create a huge knowledge base from web documents represented by graph. Afterwards, disambiguation was performed by syntactic similarity and graph

[1] Available at http://nlp.cs.swarthmore.edu/semeval/tasks/task10/data.shtml

similarity algorithms. On dataset derived from SemEval-2007 Task 07 it achieved 73.65% precision, comparing to 83.21% precision of best performing supervised system.

3 Disambiguation algorithm and methodology

Before we proceed to disambiguation itself, we have to set some prerequisites. At first, we have to set our sense inventory and level of granularity. For example, if we look up word “knife” in WordNet, we will see three possible senses:

- S: (n) knife (edge tool used as a cutting instrument; has a pointed blade with a sharp edge and a handle),
- S: (n) knife (a weapon with a handle and blade with a sharp point),
- S: (n) tongue, knife (any long thin projection that is transient) "tongues of flame licked at the walls"; "rifles exploded quick knives of fire into the dark".

There is a fine distinction between uses of knife as a tool and as a weapon, whereas in Short Dictionary of Slovak Language (Krátky Slovník Slovenského Jazyka - KSSJ), which is codifying for Slovak language, we will see only one sense:

- (ručný) nástroj na rezanie a krájanie pozostávajúci z čepele a rúčky.

This could be applied to both uses of word “knife”. So far there is no better source of word sense inventory than KSSJ, so we will stick to coarse grained senses provided by KSSJ.

3.1 Lexical resources

As we can see in example above, provided gloss does not offer many words describing particular sense. Therefore we will try to extend it by two other resources: Slovník slovenského jazyka and Wikipedia. From these sources we will map contextual information to glosses from Krátky Slovník Slovenského Jazyka and from Wikipedia, we will even try to use English version of articles.

Short Dictionary Of Slovak Language (Krátky Slovník Slovenského Jazyka – KSSJ) – as only available codifying resource, this will be our sense inventory. Getting particular sense definitions will also include modification of glosses, which contains many abbreviations which in form they are provided are useless, i.e. short hud. will be rewritten to its corresponding full form hudba (music in English). These short tags will also provide part-of-speech information, in case the word is not standalone.

Slovak Language Dictionary (Slovník Slovenského Jazyka - SSJ) – structure of glosses from this source is same as in KSSJ. Therefore the process of obtaining glosses is identical. Senses obtained from this source are used only to extend word senses obtained from KSSJ, as they are not codifying.

Wikipedia – for further extending of glosses we will also use Wikipedia. At first, we will take short descriptions of senses from disambiguation page (if there is any). Next we will check if there is a link to an article for particular sense. If so, we will test two variations: extracting the overview of article and extracting keywords of an article. In case we find out that there is corresponding English version of article for particular word sense, we will translate its contents using Yahoo Query Language[2] and extract keywords from translated text by Metallurgy web service[3]. Extracted keywords will be added to glosses.

[2] http://developer.yahoo.com/yql/

[3] http://metallurgyapi.eu/

3.2 Algorithm

Algorithm itself works in the following steps:

1. Input of algorithm is polysemous word and context (surrounding words, or whole article) in which this word occurs
2. Check if the word is standalone – i.e., that part-of-speech is either noun, verb or adjective. For this purpose we will use Slovak POS Tagger[4].
3. Get sense inventory from each available lexical resource, respecting the specifics of particular resource listed above. KSSJ is processed at first, as its glosses for certain word are fundamental sense inventory. Glosses from other resources are only used to extend these glosses, not to add new senses.
4. Lemmatize all obtained words - for this purpose we will use LemmatizerWebService[5]
5. Extend glosses provided by KSSJ by glosses obtained from other available lexical resources. For this purpose we will use simplified version of Lesk algorithm [10] - counting word overlaps between glosses. Glosses are assigned to each other by number of same word lemmas.
6. After these steps, we should have enough contextual information associated with each gloss. We will therefore proceed to comparing glosses to target corpora containing the polysemous word. Again, we will use Lesk algorithm to count overlaps between corpora and each gloss. Note, that today graph similarity algorithms are gaining more attention and shows promising results, however, their input usually is sentenced parsed into a tree and for Slovak language, this task is non-trivial and out of scope of this work. Nevertheless, Lesk performance is sufficient and still is used in many top-scoring disambiguating systems.
7. Result of algorithm is list of possible senses, ordered by possibility of the correctness of the particular sense.

There are lot of customizable attributes, which can impact the accuracy of the algorithm. These include:

- size of context window in target corpora (how many words surrounding target polysemous word will be taken into account),
- Wikipedia context - there are two variations: first is already described above and consist of using the only perex of the article and second variation uses only keywords from article extracted by Metallurgy web service,
- English resources - whether adding contextual information obtained from translated English resources are improvement or not.

4 Evaluation

4.1 First experiment

Since there are no WSD systems targeting Slovak language to which we can compare our performance, our main target is to beat threshold set by most frequent sense baseline. When first experiment took place, we have implemented only basic prototype, which included Slovak glosses obtained from KSSJ, SSJ and Wikipedia (with information only from disambiguation page). We set up two testing scenarios, both aimed on disambiguating polysemous word *hlava* (head in English):

[4] Available at http://morpholyzer.fiit.stuba.sk:8080/PosTagger/

[5] Available at http://text.fiit.stuba.sk/lemmatizer/

1. evaluation on web documents - we picked up selection of articles from news sites (namely sme.sk, hnonline.sk and aktuality.sk) containing word *hlava*. In this case, context window was the whole article,
2. evaluation on Slovak National Corpus[6] – using its web interface we picked 100 word occurrences from various resources. Context window was limited to 40 tokens to left and 40 to right side (token = word or punctuation mark).

In the first scenario, the performance was rather poor. Most frequent sense and our algorithm scored equally 36% (meaning that in 36% of cases the disambiguated word sense was the same as sense assigned by human annotator). After closer examination, we discovered, that too large context window had strong negative impact on disambiguation process, as words included from another paragraphs were not necessarily related to to-be-disambiguated word.

Performance in the second scenario was much better - 61% score by MFS and 58% score by our algorithm. There we could observe other issues causing negative impacts on performance. Frequent occurrence of pronouns and other not standalone words often caused choosing wrong sense. Also after mapping algorithm some senses contained much more lemmas, resulting in bigger chance of overlaps (this is also related to first mentioned issue, as many of mapped lemmas were not standalone words). Although results after first set of experiments were not very good, it gave us useful information about drawbacks of our algorithm.

4.2 Further evaluation

As the development continued, we felt that more sophisticated methodology of evaluation is needed. For word sense disambiguation systems, there is a well-established evaluation method, proposed by Adam Kilgarriff [11] called SENSEVAL, which was later renamed to SemEval. It provides training datasets, tagged and untagged corpora for many languages, however, none of them are for Slovak Language, which means, that if we want to imitate this evaluation, we have to build our own testing dataset. So far we extracted 285 lexical samples from Slovak National Corpus project containing one of the three chosen polysemous word (hlava, kráľ, list - head, king, letter in English) and we manually assigned a sense tag to each one of the occurrences.

Perception of word sense is a subjective manner and even human annotators are not able to achieve 100% correctness of sense tagging [11]. Percentage of words that are tagged with the same sense tag by two or more annotators is called interannotator agreement and is taken as an upper bound of performance, which can word sense disambiguation system achieve [6]. Therefore at least two human judges will have to determine whether our system assigned correct sense tag.

Evaluation on this dataset with advanced prototype of our service was more successful, than the first experiment. The most frequent sense baseline was 50.7%, while our algorithm chose the right sense in 54.2% of all instances. The biggest impact on successfulness of disambiguation algorithm was the POS tagger, which allowed us to filter out not standalone words. Not all features are implemented yet (translations from English Wikipedia) and there are still lots of attributes of algorithm which can have impact and which need to be tested.

5 Conclusion and future work

There is a great amount of research going on in the field of word sense disambiguation, however, when it comes to Slovak language, we see minimal (if any) effort. Also any potential researcher will have to face the lack of important resources. Situation is furthermore complicated by complexity of Slovak language (or any other Slavic language), where part-of-speech tagging, syntactic and semantic analysis of sentence are often much more complicated than in English language. We proposed a dictionary based algorithm for word sense disambiguation and tested the

[6] http://korpus.juls.savba.sk/

early prototype of it. The resulting performance was not as expected, however, experiment provided us with important information that can point our attention to right direction. We need to note that Slovak language itself constitute a challenge due to its inflectual nature. There is a lot of work ahead, including fixing existing issues and implementing missing features of using the translated English resources, and we believe that in the end, our method will significantly outperforms the most frequent sense baseline. Successful WSD service can improve performance of many end-user applications, not only machine translation but also text-processing applications, such as information retrieval from medical documents, for example. We also hope that this work will also encourage other researchers to participate in this field and in building missing resources and improving existing ones.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Chan, Yee Seng; NG, Hwee Tou; Chiankg, David. Word sense disambiguation improves statistical machine translation. In: *Annual Meeting-Association for Computational Linguistics.* 2007. p. 33.

[2] Escudero, Gerard; Marquez, Lluís; Rigau, German. A comparison between supervised learning algorithms for word sense disambiguation. In: *Proc. of the 2nd workshop on Learning language in logic and the 4th conf. on Computational natural language learning-Volume 7*. Association for Computational Linguistics, 2000. p. 31-36.

[3] Chan, Yee Seng; NG, Hwee Tou; Zhong, Zhi. NUS-PT: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In: *Proc. of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007. p. 253-256.

[4] Yarowsky, David. Unsupervised word sense disambiguation rivaling supervised methods. In: *Proc. of the 33rd annual meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 1995. p. 189-196.

[5] Lesk, Michael. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: *Proc. of the 5th annual int. conference on Systems documentation.* ACM, 1986. p. 24-26.

[6] Navigli, Roberto. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR),* 2009, 41.2: 10.

[7] Navigli, Roberto. A quick tour of word sense disambiguation, induction and related approaches. In: *SOFSEM 2012: Theory and practice of computer science.* Springer Berlin Heidelberg, 2012. p. 115-129.

[8] Ponzetto, Simone Paolo; Navigli, Roberto. Knowledge-rich word sense disambiguation rivaling supervised systems. In: *Proc. of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010. p. 1522-1531.

[9] Chen, Ping, et al. A fully unsupervised word sense disambiguation method using dependency knowledge. In: *Proc. of Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2009. p. 28-36.

[10] Vasilescu, Florentina; Langlais, Philippe; Lapalme, Guy. Evaluating Variants of the Lesk Approach for Disambiguating Words. In: *LREC*. 2004.

[11] Kilgarri, Adam. Senseval: An exercise in evaluating word sense disambiguation programs. In: *Proc. of the first int. conf. on language resources and evaluation.* 1998. p. 581-588

Search Resources in Unstructured Peer-to-Peer Networks

Lukáš PUTALA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova2, 842 16 Bratislava, Slovakia
`l.putala@gmail.com`

Abstract. Searching for information in unstructured peer-to-peer networks is a difficult task because of the decentralization of nodes. Network can contain millions of randomly connected nodes which do not have the knowledge of whole the network. Proposed solution is inspired by insects with non-centralized management. We are solving searching problem by using the metaphor of bees gathering food, which we partially modified because of additional needs of our work with the metaphor of ants looking for food. We compare our solution with flooding algorithm. The goal of our work is to achieve more efficient searching in unstructured peer-to-peer networks and at the same time minimize the amount of used resources and usage of network.

1 Introduction

In this paper we dedicate ourselves to searching resources in unstructured peer-to-peer networks by using the metaphor of bees gathering food [5]. Searching of information in unstructured peer to peer networks is a difficult task because of the decentralization of nodes. Network can contain millions of randomly connected nodes which do not have the knowledge of whole the network. We have inspired ourselves by the sociology of insects with non-centralized management. Our solution rests in using the metaphor of bees gathering food, which we partially modified because of additional needs of our work with the metaphor of ants looking for food [1]. Colonial species of insect, such as bees and ants, require the fastest and the most efficient system to interchange information. The source of the food is an information (data source), that is saved in individual nodes of the network. Bees looking for food represent agents looking for data sources specified in the user query.

2 Peer-to-peer networks

In our work we represent unstructured peer-to-peer network as an undirected graph [2]. Each node in the network knows only a small amount of nodes from the whole network [3]. The established connection between two nodes is bidirectional, but with different transmit delays. Transmit delay

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 21–25.

is a positive integer number, which represents the amount of milliseconds that are needed to transfer the packet between two nodes.

In our work we are using our own peer-to-peer network event based simulator. All packets in the network are stored in a list by the arrival time (time stamp) to next node. The simulator always picks the packet with smallest time and executes searching algorithm. When a new packet is send, it is inserted into the list with arrival time to the next node (current time + transfer time). Each node contains information about data it holds and a cache/memory used by simulated algorithms (e.g. flooding algorithm stores information about processed search request on each node).

3 Searching in unstructured peer-to-peer networks

Searching in unstructured peer-to-peer networks is a very difficult task from the effectiveness viewpoint, and also the load of the network. There are many solutions to a given problematic.

3.1 Flooding algorithm

The search using flooding algorithm is one of the least efficient and simplest search methods. If a given node does not contain the searched information, the request is being sent to all neighbouring nodes. Because we cannot stop its spread in the moment when we find the requested data, the search will end after all of the nodes have been visited. The disadvantage of this algorithm is in overloading the network with packets.

In our simulation the search begins with choosing the initial node and requested data. The bee is initialized from the initial node for each neighbour, the time of arrival is calculated and then the bee is inserted into the simulator. Every node does the same, if it gets a new (not seen before) request it forwards it to all her neighbour. After the requested data is found a new bee is initialized, which returns back using the same path to the initial node. However, the simulation continues until all of the nodes are visited.

3.2 Bees algorithm

At first we start the search by sending bees to the exploration graph. After the bee explores some parts of the graph of the network, the bee returns back to the beehive following the same path as it used to get there. By defining time to live parameter, we determine how many nodes bee visits before it returns back to the beehive.

To make sure each node is visited equally often, each node keeps its own data table that stores the number of times it has been visited. When the bee visits a node, algorithm calculates using an uniform distribution, the probability of what neighbouring node will be visited next. This procedure ensures that nodes that have been visited less often, have a higher chance of being visited by the next bee.

If the node does not contain data that the bee is looking for, the time to live of a bee is decremented. After that Bee’s route is updated and the node is determined by the next neighbouring node and the bee is inserted into the simulator. If bee’s time to live expires, it returns back to the beehive and updates the cache on the visited nodes. After the bee arrives to the beehive, its own primary time to live is increased by a constant.

If a bee finds the sought data, it returns to the beehive by the same way as if it's time to live has expired. At the same time bee update record to every node´s cache on the way to the beehive. Bee update information about the way from the actual node to the source node and the maximum bandwidth from the actual node to the source node. This information will be used from next bee, which goes to this way with some probability. If the bee decides to go their own way, it can finds the best way or finds other sources nodes. If the bees find more sources nodes, we can parallel download data from all nodes. In our work we consider that the every node is connect to 1GB LAN network.

4 Results

We have tested our solution on networks with 100 and 1000 nodes. We measured whole time of the simulation run, the number of transferred packets and with the bees algorithm we also alternated the number of used bees. We run the simulation 15 times on different network topologies for each algorithm with the same parameters.

Total time of the simulation is represented by the time from the start of the request until the last bee arrives to the beehive (bees algorithm) or after it finishes searching the whole network(flooding algorithm).

In the Table 1., we can see the results of the testing for a small network of 100 nodes. In this test we can see that bees algorithm is not faster than the flooding algorithm but number of packets transmitted can by considerably lower.

Number of bees represent how many bees we are used in the test. *Avg. sim. time* is average simulation time. *STDEV time* is a standard deviation for the simulation time. *Avg. time data found* is an average time when the bee returns back to the beehive with a way to the source node. *STDEV time data found* is a standard deviation this value. *Avg. packets* is how many packets algorithm used in the whole simulation. *STDEV packets* is a standard deviation this value.

Table 1. Search simulation results for 100 nodes network.

Algorithm	*Number of bees*	*Avg. sim. time*	*STDEV time*	*Avg. time data found*	*STDEV time data found*	*Avg. packets*	*STDEV packets*
Flooding alg.	-	*627*	0	627	0	858	0
Bees alg.	5	3316	1996	1862	1365	*273*	179
Bees alg.	50	1432	659	467	198	944	313
Bees alg.	100	1087	37	337	54	1565	11
Bees alg.	150	1258	181	342	57	2360	19
Bees alg.	200	1078	42	312	44	3135	8

In the Table 2., we can see the result of the testing for larger network of 1000 nodes. We can see that flooding algorithm found data in the average time a bit faster than bees algorithm but uses a lot more resources than bees algorithm. We can notice that with the increasing number of bees the simulation time is decreased. Even though the number of packets is increasing, with the number of bees increasing, it is still less than what was used in the flooding algorithm. When we select a suitable number of bees, the average search time is approaching the average time of the flooding search algorithm, but load on the networks is much lower.

Table 2. Search simulation results for 1000 nodes network.

Algorithm	*Number of bees*	*Avg. sim. time*	*STDEV time*	*Avg. time data found*	*STDEV time data found*	*Avg. packets*	*STDEV packets*
Flooding alg.	-	*922*	0	922	0	18557	0
Bees alg.	10	7586	3088	4697	2450	1296	538
Bees alg.	100	2366	916	983	479	3289	1354
Bees alg.	500	1242	206	*299*	83	7990	35
Bees alg.	1000	1285	161	*273*	32	15960	23
Bees alg.	1500	1180	24	*243*	4	23933	21

As a second we tested the download speed on networks only for 100 nodes. We used 100 bees in bee´s algorithm and 10% of the duplicates source data in nodes. We measured time to find all sources nodes and the download speed. We run simulation 5 times on equally network and parameters as previous test for 100 nodes. In this test we consider only one way to download data from every source node. If the bees found the better way from some source, the way was update for this way. We consider for better way when the total bandwidth from this source is higher.

In the Table 3., we can see the results of the testing for a small network of 100 nodes. In this test we can see that bees algorithm found first source faster than flooding algorithm. Second source bees algorithm found just a little latter. When flooding algorithm found first source, bees algorithm already has twice more download speed as the flooding algorithm. Bees algorithm found all 10 sources faster than flooding algorithm and download speed is about 19 mb/s higher.

Time found represent time, in milliseconds, wherein the algorithm found sources. *Download speed* represent speed, in megabytes peer second, which the algorithm download data in the time.

Table 3. Download speed simulation results for 100 nodes network.

Sources	*Time found Bees alg. [ms]*	*Time found Flooding [ms]*	*Down. speed Bees alg. [mb/s]*	*Down. speed Flooding [mb/s]*
1	*164*	180	1	4
2	167	240	8	9
3	180	286	14	13
4	210	362	18	17
5	235	392	23	20
6	244	407	27	25
7	326	430	38	30
8	388	497	41	32
9	485	644	50	34
10	*509*	745	*55*	36

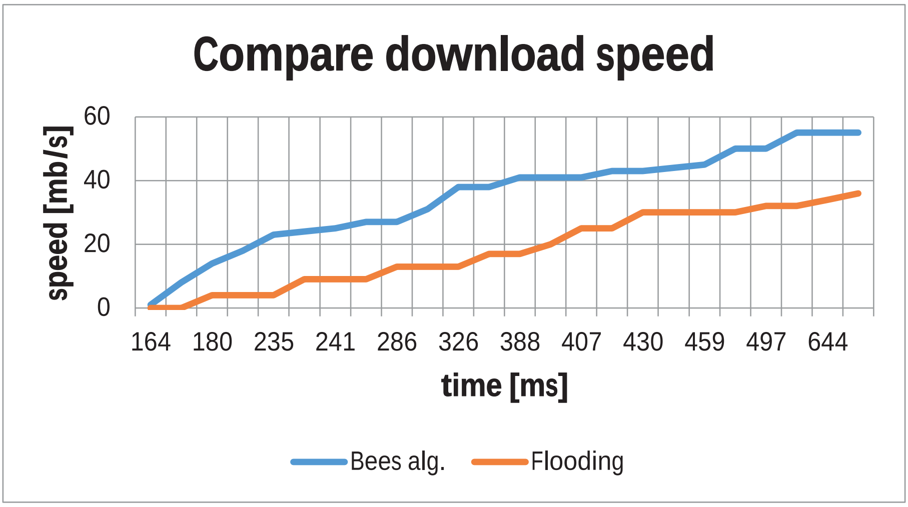


Figure 1.Compare download speed for bees algorithm and flooding algorithm.

In this graph we can see progress to the download speed depending on the time. The download speed for bees algorithm is more better, because bees updated way more times and found more optimal way.

5 Future work

We will further focus on optimization of the bee search algorithm. After returning back to the beehive, bee might subsequently decide whether to explore the graph or whether to do the celebration dance (share information with other bees), based on the data it has found before. This can help us find better (with more bandwidth) routes to data sources.

6 Conclusion

In our work we have proven that using the bees algorithm and by choosing the optimal amount of bees we can find the data faster than with the flooding algorithm. We have also managed to decrease the amount the packets used to find requested data and by that even decrease load on the network. We used the caches on each node to store local area information. The search is no longer be random search because bees used this information for the next searching. Using this knowledge we could improve the download speed compared with flooding algorithm.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0233-10.

References

[1] Ismaili, F.: *Bio-inspired algorithms for P2P overlays*. [Online; accessed February 19, 2014]. Available at: https://diuf.unifr.ch/main/pai/sites/diuf.unifr.ch.main.pai/files/PAI_Seminar-_2013_Ismaili.pdf

[2] Kvasnička, V.: *Teória grafov I*. [Online; accessed February 19, 2014]. Available at: http://www2.fiit.stuba.sk/~kvasnicka/DiskretnaMatematika/Chapter_10/transparencies10.pdf

[3] Ledvina, J.: *Strukturované a nestrukturované P2P sítě*. [Online; accessed February 19, 2014]. Available at: http://www.kiv.zcu.cz/~ledvina/Prednasky-DS-2009/DS-12-P2Pa.pdf

[4] Navrat, P., Kovacik, M.: *Web Search Engine as a Bee Hive*. IEEE/WIC/ACM International Conference, Hong Kong, (2006).

[5] Sabo, Š.: *Online získavanie informácií z textových zdrojov*. [Online; accessed February 19, 2014]. Available at: http://is.stuba.sk/lide/clovek.pl?id=17053;zalozka=7;studium=72038

Authentication on Smartphone Using Keystroke Dynamics Together with Hardware Sensors

Štefan Šmihla[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`stefan.smihla@gmail.com`

Abstract. In current time, when technology is being evolved in direction of smartphones and tablets, we realize importance to improve security for authentication of user. To achieve this goal we propose to implement multifactor authentication through one of biometric methods. In this work we focus on improvement of authentication through keystroke dynamics. We would like to enhance common time based metrics with hardware sensors like device orientation and accelerometer. The aim of this work is to explore metrics based on hardware sensors and to evaluate accuracy after implementation of these sensors.

1 Introduction

In recent years smartphones and tablets have become very popular and for many people they became important part of everyday life. People tend to use smartphones to entertain themselves, for work purposes or to easily access information. Also smartphones are often used as a storage to keep valuable private or business data. Together with these growing opportunities there is increase of security risk that unauthorized person is able to gain access to services and data. Password alone is not good enough to ensure expected security. For unauthorized person it might not be a problem to guess, log or gain password from weak secured place. In addition users tend to use easy-to-remember phrases as passwords. Therefore need to improve authentication is growing and could be achieved with combination of password and some specific biometric method.

Keystroke dynamics is biometric method based on that every human has unique typing style on which base he could be identified. If unauthorized person manages to obtain password, system still can forbid access if typing style does not match with authorized person. An advantage of this method is that it could be low cost implemented with basic hardware. Research on keystroke dynamics dates from 1980 with computer's keyboard and continues today. However research on smartphone platform with touch based display is still behind research on computers with physical keyboard and mouse.

[*] Bachelor study programme in field: Informatics
Supervisor: Assoc. Professor Daniela Chudá, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 26–31.

In this work we focus on smartphone platform and by enhancing authentication security through keystroke dynamics. Most of smartphone devices provides several different hardware sensors. We use some of them to monitor behavior during password authenticating process. We evaluate these logged data provided from hardware sensors together with time based vectors provided from keystroke dynamics and we conclude if these metrics can enhance accuracy of authentication.

2 Related work

Research on keystroke dynamics has begun since 1980 by Gaines et al [3]. In their research they started with T-statistics method to evaluate flying times between pressed keys on written phrases and sentences.

Although the research on keystroke dynamics on computers started in 1980, keystroke dynamics on mobile devices has begun in 2006 by Clarke and Furnell [2]. In their research they used 12-key physical keyboard and they measured flying times between keys and hold times for pressed keys. They evaluated three experiments, first for authentication based on PIN, second as alphanumerical password and third experiment as phone number. In first experiment they reached FAR with accuracy 3%, however accuracy of FRR was 40%. In second experiment they reached FAR with accuracy 15% and FRR with accuracy 28%. Last experiment ended with accuracy of 18% in FAR and 29% in FRR.

In 2008 Clarke with Buchoux continued on research and they evaluated 4-digit PIN and strong alphanumeric password [1]. They also compared Mahalanobis and Euclidean distance vector algorithm. Mahalanobis algorithm showed to be more benevolent in FAR, but more accurate in FRR. They also proved that 4-digit PIN cannot be used with keystroke dynamics due to high acceptance rate, which was more than 50% on both distance based algorithms.

Interesting results were reached by researchers Zahid, Shahzad et al. in 2009 [6]. They designed dynamical continual method which evaluates user after login and they combined several advanced algorithms. They reached FAR with accuracy 2,07% and 1,73% accuracy in FRR.

In 2011 Maiorana et al. prepared research based on several password lengths [4]. They represented dependence on EER as complexity of password has grown. As assumed, EER was decreasing with raising password's complexity. In 4-digit password EER had 20% accuracy and on 10-digit password EER decreased to accuracy around 10%.

Very interesting research for this work was done by Trojahn and Ortmeier in 2012 [5]. They noticed that besides standard metrics like digraphs and trigraphs for flying times and hold times, another metrics can be used like error rate, pressure and finger size. In their work they used Euclidean distance based algorithm and they compared entered password on 12-key virtual keyboard layout and QWERTZ virtual keyboard layout. During authentication process they evaluated their solution particularly on numerical and particularly on alphabetic password on different keyboard layouts. Alphabetic password brought slightly more accurate EER than numerical password and 12-key virtual layout showed to be more accurate with EER than QWERTZ virtual keyboard layout.

Summary of all results are shown in Table 1. Most of these works evaluates authentication on 12-key based physical keyboard and measures only digraphs for flying times and hold times. Only Trojahn and Ortmeier used another valuable features provided from smartphone in their solution.

3 Authentication by keystroke dynamics and hardware sensors

The principle of multifactor authentication through keystroke dynamics is based on creating biometric template consisted from multiple biometric samples. Every biometric sample contains values measured during authentication process. During registration, user is tasked to enter password several times to create biometric template. Then during login biometric template is compared with a new biometric sample and depends on their similarity, user is accepted or rejected. This principle is shown in Figure 1.

Table 1. Overview of related works focused on mobile devices and smartphones.

Author	Year	FAR	FRR	Application	Keyboard
Clarke & Furnell	2006	3% 15% 18%	40% 28% 29%	PIN Password Phone num.	12-key 12-key 12-key
Buchoux & Clarke	2008	15% 2, 5%	57, 5% 20%	4-digit PIN Password	12-key 12-key
Zahid & Shahzad	2009	2, 07%	1, 73%	Password	QWERTZ
Maiorana a kol.	2011	EER 10% EER 20%		4-digit password 10-digit password	12-key 12-key
Trojahn & Ortmeier	2012	12, 13% 9, 04% 9, 53% 8, 31%	8, 75% 6, 66% 5, 88% 5, 26%	Numerical password Numerical password Alphabetic password Alphabetic password	QWERTZ 12-key QWERTZ 12-key

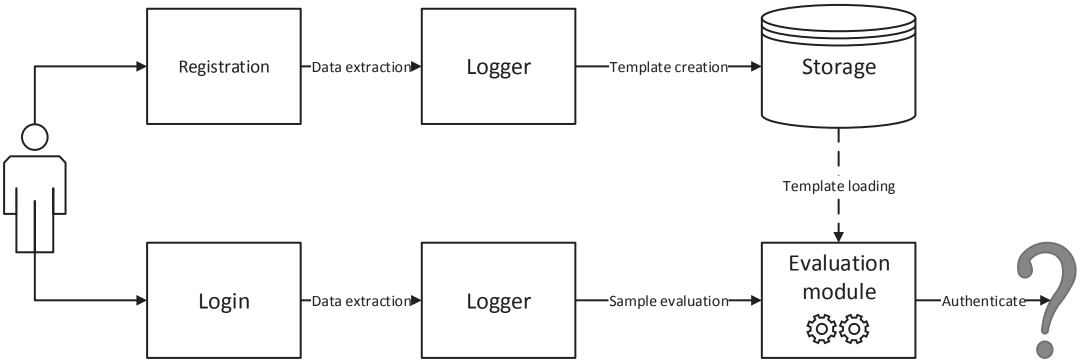


Figure 1. Authentication process - registration and login.

Keystroke dynamics is based on several biometric characteristics. Two main measured characteristics are flying times between pressed keys (digraphs) and hold times for every pressed key. Also it is common to measure time between three pressed keys (trigraphs). Error rate can be also evaluated and additionally for typing non-alphabetic characters it is possible to check, if user tend to change keyboard or rather perform long press.

3.1 Statistical T-test

In comparison between biometric samples and templates, several algorithms can be used. In this work we evaluate statistical T-test. Basic principle starts with iterating through stored biometric samples and computing average values $\overline{x}_k$ for every pressed key. Standard deviation σ_k is also computed.

During evaluation, values from test sample are compared against average values from template extended with threshold P. Standard deviation is also added. This give us min_k and max_k values (eq. 1 - 2). Test sample x_k from correct user should be ranged between min_k and max_k (eq. 3).

$$min_k = (1 - P) * (\overline{x}_k - \sigma_k) \tag{1}$$

$$max_k = (1 + P) * (\overline{x}_k + \sigma_k) \tag{2}$$

$$min_k \leq x_k \leq max_k \tag{3}$$

At the end, all tested values are counted and compared to values which passed this test. This gives us value T which represents probability whether evaluated user is appropriate or not. In this work we consider user as correct if $T \geq 0.8$.

3.2 Hardware sensors

Smartphone devices provide several sensors which can enrich keystroke dynamics on smartphone platform. Most of them are accessible and easily implemented. Touch pressure is computed by touched area. With raising pressure a greater area is touched, however there is limitation of how many pixels are touched. These pixels are standardized to range $<0 - 1>$ with 1 as maximum area. Display orientation can be also checked. Several users tend to use vertical direction and some users tend to use horizontal direction. Horizontal direction can be directed to right or left. Authorized user can have preferred orientation. Orientation of device in space can be also measured in degrees on X, Y and Z axis. Accelerator measures motion of device in space and it might be used to evaluate how much user shakes with device during authentication and to measure emotions or environment dynamics. Smartphone devices also contain atmospherics pressure, light and temperature sensors, but in this work we rather focus on touch pressure, orientation and accelerator.

4 Data logger

We developed logger on Android device which measures most characteristics mentioned above. However there are several limitations. Together with flying times, logger counts how many times delete key was pressed and stores it as error rate. It also logs for non-alphabetical keys if user tend to change keyboard or rather do a long press on key and stores it as long press rate.

Although flying times are measured correctly through *TextWatcher* Android class, for logging hold times custom keyboard should be developed. However custom keyboard would confuse users and costs development time, so we kept logger without custom keyboard. Due to that, hold times are not measured as planned.

Without custom keyboard we have also limitation for touch pressures, which can be obtained only on Android devices with version lower than 4.0.2. On newer devices support for touch pressure through service was removed due to security reasons. Orientation in space is measured same way as accelerometer, however some devices are not equipped with orientation sensor and do not fully measure orientation in space. Fortunately simple display rotation is logged without limitation together with accelerometer.

5 Evaluation

Proposed system was evaluated on 15 users. They could use their own device and they were asked to create two templates with two specific phrases as password. First password was simple and common phrase *"vcelimed"* with specific length of eight characters and without non-alphabetical characters. Second password was more complex with two non-alphabetical characters and length of sixteen character. We used phrase *"l3kvarov@strudla"* as a second password. To create templates from both phrases, users were asked to enter password twenty times, which stored twenty biometric samples for each user.

In analysis, first five samples were ignored. They were used for users to train password. Another ten samples were used to create biometric template. Last five samples were used as test samples to evaluate our solution. As shown in Tables 2 and 3, we split evaluation process for three different characteristics to show difference between them. We evaluated all of them in error rate and also

Table 2. Results for simple phrase.

Characteristic	Error rate	FAR	FRR
Flying Times	No	18.48%	6.67%
	Yes	**18.19%**	**6.67%**
Accelerance	No	40.67%	50.67%
	Yes	39.24%	50.67%
Orientation	No	45.62%	38.67%
	Yes	44.95%	38.67%
All above	Yes	19.33%	28.00%

Table 3. Results for complex phrase.

Characteristic	Long press rate	Error rate	FAR	FRR
Flying Times	No	No	10.44%	5.71%
	Yes	No	**7.03%**	**5.71%**
	No	Yes	10.11%	7.14%
	Yes	Yes	6.70%	7.14%
Accelerance	No	No	33.41%	50.00%
	Yes	No	16.81%	50.00%
	No	Yes	32.75%	51.43%
	Yes	Yes	16.48%	51.43%
Orientation	No	No	37.91%	28.57%
	Yes	No	19.01%	28.57%
	No	Yes	37.58%	30.00%
	Yes	Yes	18.68%	30.00%
All above	Yes	Yes	6.04%	18.57%

for complex phrase we evaluated long press rate. On the last row, we combined flying times, acceleration and orientation together. In Table 2, we got relatively high FAR for every measured characteristics. To more equal FAR to FRR we need to scale threshold. In results from complex phrase in Table 3, flying times show relative good accuracy. As expected long press rate led to more accurate results. FAR was reduced by error rate only slightly, however it reflected on increased FRR. Accelerometer and orientation in space showed to be useless on both phrases. However they might be used in different way, e.g. to measure emotions or to measure environmental influence. To confirm that, additional research is needed. We evaluated additional scaling to get more balanced FAR and FRR. We used flying times and long press rate, which showed to be most accurate. As shown in Table 4, we obtained equal error rate (EER) around 9-11% for simple phrase and EER around 6-7% for complex phrase. Also as shown in Table 4, threshold might be scaled together with raising password's complexity.

6 Conclusions

In this work we designed solution to evaluate flying times, accelerometer and orientation in space. We used statistical T-test in evaluation and enhanced results with long press rate and error rate. As expected long press rate can improve accuracy of authentication. Error rate also increases possibility, that system falsely reject authorized person. We show that accelerator and orientation alone are not

Table 4. Results depended on sensitivity of threshold.

Threshold	-	0.05	0.1	0.15	0.2	0.25	0.3
Simple phrase	FAR	3.52%	6.10%	**11.24%**	18.48%	27.43%	33.90%
	FRR	45.33%	24.00%	**9.33%**	6.67%	6.67%	2.67%
Complex phrase	FAR	0.55%	1.32%	3.63%	**7.03%**	12.31%	16.70%
	FRR	67.14%	42.86%	11.43%	**5.71%**	2.86%	2.86%

accurate enough to authenticate user, however they might be used in different way e.g. to identify emotions or to measure environmental influence. To confirm that, additional research is needed and might be subject of future work.

We reached best accuracy by evaluating flying times with long press rate. With additional threshold scaling we designed system with accuracy ranged from around 89-91% (in simple phrase) to around 93-94% (in complex phrase). However to confirm that results it would be correct to let users log in after some time and check if accuracy of authentication has changed.

In future work we plan to evaluate our solution on more samples and to implement Manhattan, Euclidean and Mahalanobis distance vector algorithm and compare them with statistical T-tests. We also consider possibility to use accelerator and orientation in space for authentication in different way as we used in this work.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

References

[1] Buchoux, A., Clarke, N.: Deployment of Keystroke Analysis on a Smartphone. In: *Australian Information Security Management Conference*, 2008, p. 48.

[2] Clarke, N., Furnell, S.: Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 2006, vol. 6, no. 1, pp. 1–14.

[3] Gaines, R., Lisowski, W., Press, S., Shapiro, N.: *Authentication by Keystroke Timing: Some Preliminary Results.* RAND Corporation, Santa Monica California, 1980.

[4] Maiorana, E., Campisi, P., González-Carballo, N., Neri, A.: Keystroke Dynamics Authentication for Mobile Phones. In: *Proceedings of the 2011 ACM Symposium on Applied Computing*, New York, 2011, pp. 21–26.

[5] Trojahn, M., Ortmeier, F.: Biometric authentication through a virtual keyboard for smartphones. *International Journal of Computer Science & Information Technology*, 2012, vol. 4, no. 5, pp. 1–12.

[6] Zahid, S., Shahzad, M., Khayam, S., Farooq, M.: Keystroke-based User Identification on Smart Phones. In: *Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection*, Berlin, Heidelberg, 2009, pp. 224–243.

Personalized Expert Food Recommendation System

Peter TRUCHAN*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
truchanpeter@gmail.com

Abstract. In this paper we describe the problem of personal recommendation in context of alimentation and healthy food. We propose a recommendation system and the metrics based on Guideline Daily Amounts of nutrients. In this paper, there are analysed tools that support healthy eating by making recommendations based on expert knowledge. The output would be product, which will be using these personalized recommendations. It will also try to help in general knowledge about health and it will encourage the users to follow guides about daily amounts of the most valuable nutrients, which could influence their performance and physical and mental health.

1 Introduction

Every human being in this world has unique organism and metabolism. People live different lifestyles. We need to supply our body with enough vitamins, minerals and other nutrients. The main function of our body is to gain energy from food. However, we need lots of other nutrients to complete this process. Our food needs to be sufficiently diverse. Nowadays, one of the biggest problems is the fact that food contains less and less essential nutrients. Scientists found out that apple nowadays contains 50 % less vitamins than 20 years ago [2]. In the antique world people knew about the influence of food to our life and health. Today, we have several methods how we can track what we eat, but they are not very easy to use for common people. We tried to find a scientifically based method, that would best suit the nutritional needs of contemporary people, and make it better and easier to implement into one's life.

Daily Guide Amount is system created to cover almost all nutrients the human body needs. European Union and also other national and international institutions try to reduce number of ill and obese people. One of the measures is that, almost all food contains informations about composition on its cover. It is help for people on diet, but it includes a few information for people with eating disorders. It contains only informations about energy value and amount of lipids, proteins and carbohydrates.

[*] Bachelor study programme in field: Informatics
Supervisor: Professor Pavol Návrat, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 32–37.

1.1 Related work

At the beginning we analysed different approaches to healthy eating. One of it was system based on ratio between different groups of food. It is called the food guide pyramid. The food guide pyramid is a pyramid shaped guide of healthy foods divided into many sections to show the recommended intake for each food group. The first food pyramid was published in Sweden in 1974 [9].

The main disadvantage of this approach is its small accuracy. Better way is to watch all essential nutrients and its amounts. To do this, we need lot of information about composition of food. In 100 years there were several attempts to collect these data. One of the best public accessible source of data is National Nutrient Database for Standard References. There are nutrient information over 8,000 foods. You can search by food name, group, or list to find the nutrient information about food. This functionality is great, but if you want to calculate everything you eat through the day, you have to spend a lot of time calculating and searching. This database is created and administrated by nutrition specialists. For this reason we used this system and made it more accessible, user-friendly and more popular among population.

2 Meal menu building

2.1 Data preparation

Firstly, we prepare database for the needs of our system. We focused on easy searching and speed.Then we tried to automatically localise the database to other languages. This task was completed but the results was not as good as we expected. For this reason we use English version of database for the English mutation of system and we use poorer Slovak database for the Slovak language mutation. This database contains only around 1400 foods. On first look this number is not big enough, but these are specially chosen foods, which are mostly used by Slovak consumers. The main function of our system is to create personalized meal menus based on science. It should be designed for users and provide calculations to reach their goals.

After downloading and editing the national nutrient database we focus on efficiency and usability. We implemented an advanced searching algorithms as a Double Metaphone sound-alike search [7] and Trigram method [1]. When a user wants to search some food in our database, we try to find it and the results are then collected and sent back to user. After that, he can choose from the group of foods the one he was searching for.

2.2 Data collection

The main task for user is to create a menu for eating which will be balanced and healthy. Our tool provides functionality for fast creating, sharing, ranking and using daily menus.

After user finds out his admired food he will just add information about amount and the food will be added to his daily menu. He can now lookup information about composition of his daily intake. Tool also provides well-arranged information about foods with highest amount of concrete nutrient. If user wants to increase daily intake of proteins, he can easily find out food with the highest amount of desired nutrient. He can also change his guided daily amounts and the tool will adjust. When the menu is created, there is a function for finding best matching additional food to balance menu. We used a modified Singular Value Decomposition recommendation tool. The tool is ready for using different algorithms to gain the best afford for user.

2.3 Calculation of composition

This is the most important part of the tool. In fact, we do not need special algorithms for this part. In our database there is data for 100 grams of every food. We only need to get data from system and calculate amount of nutrients from height of the food. After that, the amount of nutrients is displayed in well-arranged graphs. We designed colour-full graphs where user can see when something is wrong at the first look. There is only one problem in calculation, because not every food has all information about composition. We will ignore this cases and when the information is missing we will assume that this food does not have any amount of missing nutrient.

3 Automatic generation and recommendation

3.1 Meal menus generating

Our main goal is to provide user a tool which will keep him healthy. The other goal is to save him as much time as possible. We invented functionality for generating meal menus. After some time, when we collect enough data from user, we will be able to find out his favourite foods and food combinations. With the help of logistic regression [4] we are able to find similar users (based on similar eating habits). This will help us in generating several meal menus. We take all foods from all similar users and combine it. After that, the user will be able to choose and modify generated menus. In the future, there is also possibility to connect system with intelligent fridge and create shopping list.

3.2 Food recommendation

We are using singular value decomposition algorithm to recommend missing foods in meal menus. Algorithm choose foods primary from set of previously used foods in meal menus. This is because of too large database combined with fact that every person has own preferences in eating. There are too many foods, and there is big chance that user cannot buy or he does not like the recommended food. We will find set of foods from user menus and from menus of other users which we will find through collaborative recommendation. We will include similar users in the same set based on foods they have been eating the in past. This approach has main disadvantage in cold start, but this can be solved by finding similar users at start based on their age, sex, height or hobbies. After we have set of proper foods, we can find the one or two which fits the most to our daily menu. Firstly, we calculate how much nutrient we need to make daily menu perfect. Than we use our algorithm to find the best matching food to this complement of daily menu.

3.2.1 Linear Algebra

SVD methods are a direct consequence of a theorem in linear algebra:

Any MxN matrix A whose number of rows M is greater than or equal to its number of columns N, can be written as the product of an MxM column-orthogonal matrix U, and MxN diagonal matrix W with positive or zero elements (singular values), and the transpose of an NxN orthogonal matrix V [6].

Assume that we have a matrix where column represents a food, and row represents a nutrient. With M foods and N nutrients, we are looking at an MxN matrix. The theorem simply states that we can decompose such a matrix into three components: MxM, MxN, and NxN. More importantly, we can use this decomposition to approximate the original MxN matrix. By taking the first k values of the matrix S, we can effectively obtain a compressed representation of the data.

3.2.2 Machine Learning

One of the most fundamental properties of Machine Learning is its close correlation to the concept of data compression. If we can identify significant concepts, then we can represent a large dataset with smaller. SVDs allow us to compress a large matrix by approximating it in a smaller-dimensional space.

SVDs found wide application in the field of Information Retrieval where this process is often referred to as Latent Semantic Indexing. In these applications the columns of the matrix are the documents, and the rows are the individual words. Running SVD allows us to collapse this matrix into a smaller- dimensional space where highly correlated items (for example, words that often occur together) are captured as a single feature. In practice, programmers usually collapse their enormous matrices to 100, 200, or 300 dimensions (from original 10000+) and then perform similarity calculations. This same method has also found many uses in image compression and computer vision applications [8].

3.2.3 Reduction of dimensions

For the sake of brevity we will use a very simple example with only 4 foods, and 6 nutrients.

Table 1. Exemplar source data.

	apple (182g)	pear (178g)	lemon (6g)	banana (118g)	strawberries (147g)
Energy (kcal)	95	101	3	105	47
Carbohydrates	18.91	17.36	0.25	14.43	7.19
Iron	0.22	0.32	0.05	0.31	0.6
Vitamin C	8.4	7.7	7.7	10.3	86.4
Water (ml)	155.72	149.45	4.9	88.39	133.7

Table 2. Exemplar missing nutrients in daily menu.

	complement #1	complement #2
Energy (kcal)	100	50
Carbohydrates	20	8
Iron	0.5	1
Vitamin C	10	90
Water (ml)	140	150

Cranking this matrix through the SVD yields three different components: matrix U (6x6), matrix S (6x6), matrix V (4x4). Now, we collapse this matrix from a (6x4) space into a 2-dimensional one. To do this, we simply take the first two columns of U, S and V [3].

Now, we can plot results. We can treat the first column of U as x , and the second column as y. These are the foods. Process is then repeated for matrix V. These are the nutrients.

Because we are working with a small example it is hard to call two foods a ’cluster’ but you will notice that apple and pear are located very close to each other. When we compare their amount of nutrients in our original matrix, we can see that it is right. Our dimensionality reduction technique effectively captured the fact that apple and pear seem to have similar amount of nutrients. Now, we only need to calculate similarity.

3.2.4 Finding similar foods

If we add next meal menu to our set of properly foods (in this example complement #2), we want to recalculate similarity. To do this we perform the following calculation.

$$complement_{2D} = complement^T x U_2 x S^{\overset{-1}{k}} \tag{1}$$

First line is the general formula to project a new food into our space. More information about this process is available [6]. The important result is that we have the x, and y coordinates for complement. Let us add them to our earlier graph in the Figure 1.

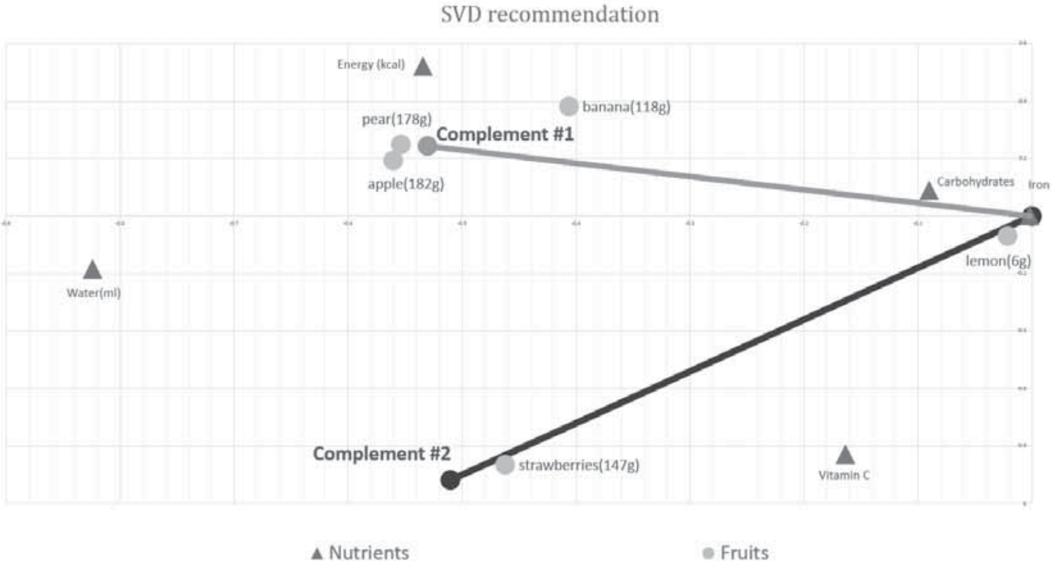


Figure 1. Graph of similar foods and nutrients.

4 Conclusion

We created prototype of our application and we analysed users opinions and their expectations from the system. The application was tested on three subjects. Subject has 3 scenarios to complete. Two of them is successful and one did not finish two scenarios. The main reason was problem with computer and internet connection.

4.1 Scenarios

The scenarios are designed for different types of persona. First one is the body-builder who has extraordinary requirements for daily amounts of nutrients. First scenario for this persona is to change default daily amounts of nutrients. This user has already created account and he is using our application. Second persona is obese person whose goal is to lose weight. He wants to check his weight and modify daily menu to lower amount of energy. Third scenario is about choosing the existing daily menu. User should choose existing menu which has his favourite foods in it and which contains right amounts of nutrients.

4.2 Results

Our results are presented in Table 4. We used Likert measurement of attitudes (0 means I totally disagree, 7 means I totally agree) [5].

Table 3. Information about testing subjects.

Name	Sex	Age	Goal
Lukas	M	21	body-building
Zdena	F	43	health
Martina	F	37	lose weight

Table 4. Likert measurement of attitudes.

Question	Lukas	Zdena	Martina	Average
The website is intuitive, nice and clean.	7	3	5	5
I found the right foods.	6	2	5	4
I want to use this application in the future.	7	5	7	6

5 Future work

In the future, we will present our results of the used algorithms and recommendations. We would like to present system influence to users health, and more sophisticated algorithms for recommendation. The main metrics of success of our system is number of daily users. We will also measure how many users will add recommended menu or meal to their daily menu. There can be also interesting data about personal characteristics of our users.

References

[1] Angell, R.C., Freund, G.E., Willett, P.: Automatic spelling correction using a trigram similarity measure. *Information Processing & Management*, 1983, vol. 19, no. 4, pp. 255–261.

[2] Davis D, Epp M, Riordan H: Changes in USDA Food Composition Data for 43 Garden Crops, 1950 to 1999. *Journal of the American College of Nutrition*, 2004, vol. 2004, no. 669-682.

[3] Grigorik, I.: SVD Recommendation System in Ruby. *http://www.igvita.com/2007/01/15/svd-recommendation-system-in-ruby/*, 2007, [Visited 19.02.2014].

[4] Hosmer Jr, D.W., Lemeshow, S.: *Applied logistic regression*. John Wiley & Sons, 2004.

[5] Likert, R.: A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[6] M.W.Berry, S.T. Dumals, G.: Using Linear Algebra for Intelligent Information Retrieval, 1994.

[7] Philips, L.: Hanging on the metaphone. *Computer Language*, 1990, vol. 7, no. 12 (December).

[8] Singhal, A.: Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2001, vol. 24, no. 4, pp. 35–43.

[9] Wills, D.J.: Food-Based Dietary Guidelines in Europe. *EUFIC REVIEW*, 2009, vol. 10.

User's Model Characteristics Relevancy and Weight for User Emotions Identification

Ivana BOHUNICKÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
bohunicka.i@gmail.com

Abstract. Researches focused on human emotions identification at work with computer reach increasing care in the area of human computer interaction and artificial intelligence. Researchers aim to recognize user´s emotions the most exactly. Condition of solution is that emotions influence work and behavior of person during the time of using computer. In this paper we are focused on user´s model biometric characteristics and its influence determination for recognition of emotions at work with computer. We propose method for influence determination of each particular characteristic inspired by calculating information gain used in classification. On the base of influence and according to properties of these characteristics we have defined own weighting coefficient, which we apply in comparative methods for emotion´s recognition. We show influence of user characteristics and how our approach increases success of emotion recognition.

1 Introduction

In this paper we are focused on the user´s model for the aim of his emotions and emotional status identification. Mood, emotions, feelings and states of human are changeable in various situations, also during the time of using computer. Occur of negative emotion at work with computer is highly frequent. This fact was confirmed by symposium that we realized. Our respondents said that they have felt states as tiredness and stress periodically. But computer systems are not able to understand and adapt to user. Because of this reason computer programs react unsuitable, provide incorrect feedback, applications interrupt work and it lead to user increasing frustration. So, after successful recognition of actual user emotion in computer environment, solutions want to offer better and more effective conditions of his work.

Our general goal is recognition of emotions by modeling user according his biometric characteristics. In solution we come out from fact, that emotions influence human work and behavior during using computer. User behavior and manners affect biometric characteristics such as keystroke dynamics, mouse dynamics and work with applications. These characteristics we can monitoring and then create model from characteristics values. We understand user model as mathematical formula of person behavior that allow us to recognize user or user emotions.

[*] Master degree study programme in field: Software Engineering
Supervisor: Assoc. Professor. Daniela Chudá, Institute of Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Identification is realized by comparing created model with data captured from real time work with computer. In this paper we are mainly focused on selection of the most relevant user characteristics and their weighting, that lead to more exact emotion recognition.

2 Related work

The ability to express and recognize emotions plays the important role in human communication and once more in human computer interaction. Researches has far long time ago proved, that human has tendency to communicate and work with computer in the natural social way reflecting interactions of human at usual social events [6]. In area of human computer interaction the production of emotional intelligent system´s increases, these systems will react on human´s emotions and behavior at time of working with computer.

Solution of given assignment consist of four basic steps:

1. collect user behavior characteristics,
2. create model from collected characteristics,
3. identification of emotion by comparing models,
4. execute adequate action according recognized emotion implies recommendation or feedback.

For creating model it is necessary to collect sufficient amount of data about user and his work. For this, there are various tools, such as [2] which process data from keystroke dynamics and mouse dynamics, monitoring human´s behavior at time of using applications and development tool.

In article [1] for distinguish the emotions authors have been interested by two analyzed approaches: dimensional and categorical. In studied works of user´s emotions modeling the authors [1], [3], [4] are focused on categorical approach, where the user has the possibility to choose from concrete emotions. Dimensional approach divides emotions into groups by two dimensions: arousal and valence. Besides emotions, there is the possibility to deal with identifying states such as stress, tiredness or diseases [5].

In terms of identification, authors [1], [3], [4] promise successful results of emotions recognition by using classification and statistical methods. In article [5] mentioned advantage of using statistically pre-processed data.

3 Our method for user’s model metrics valuation

We proposed method consists of user’s characteristics selection and weighting. These characteristics called the metrics. We choose these metrics:

- keystroke dynamics including elapsed time between keys press and duration of a key press,
- mouse dynamic including mouse speed, mouse acceleration, scroll speed, left button clicking and right button clicking,
- work with applications including count of running applications and window status of running applications,
- computer system usage including charging of memory and processor.

Our proposed method aims to define influence of each metric on emotion recognition and this method includes two steps:

- determination of metrics relevancy,
- determination of metrics weight.

3.1 Determination of metrics relevancy

Importance of each metric is different in emotion recognition, because values of characteristics are differently impacted by emotion affecting. Determination of metrics relevancy means determination of their importance. We propose method for determination metrics relevancy inspired by calculating information gain. Information gain is usually used in classification method decision tree for setting the order of attributes in tree nodes. We propose using of this method for setting the order of metrics.

For information gain we have to know value of entropy, which describes a measure of disorder and homogeneity. Entropy is defined as (see Formula 1):

$$Entropy(S) = \sum_{i=1}^{c} -p_i * \log_2 p_i \tag{1}$$

where S is a collection of all examples, p_i is the proportion of S belonging to class i. Classes in our solution are emotions. Then information gain is defined as (see Formula 2):

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2}$$

where *Values(A)* is the set of all possible values of attribute A. Attributes in our solution are metrics. S_v is the subset of S for which A has value v.

On the base of information gain we find out metrics relevancy. The most relevant are keystroke dynamics metrics and order of relevancy from the most relevant is:

1. keystroke dynamics,
2. mouse acceleration,
3. mouse right button clicking, mouse left button clicking,
4. charging of memory,
5. running applications,
6. mouse speed, mouse scroll speed,
7. charging of processor.

We use metrics relevancy in following determination of metrics weight.

3.2 Determination of metrics weight

In our solution we recognize emotions by statistical methods Euclid distance and Manhattan distance, which we apply on created model of user metrics. It is advisable to use weighting in these methods. Because of that purpose we suggest own weighting coefficient. By iteration and experimental process we get final form of coefficient. Our suggested coefficient is defined as (see Formula 3):

$$weighting\ coefficient = \frac{R * x^2}{|mean - median|} \tag{3}$$

where x is multiplicity of current metric appearance, R is metric relevancy and |*mean-median*| express subtraction of mean and median of current metric values.

Influence on metrics weight has a frequency of their appearance. So value x express multiplicity of metrics appearance. More frequent metric provide more information about user's behavior influenced by emotion affecting.

But, range of metrics values, which are highly frequent, can be huge. It can deface and make worse result. This case we can recognize from values of mean and median. Bigger range of metrics values caused bigger difference between mean and median. So metric's multiplicity is divided by absolute value of mean and median subtraction.

At last we add metrics relevancy described in chapter 3.1 to weighting coefficient. Relevancy is express as variable R.

4 Experiments

For experiments we had to get data of user behavior and emotions during work with computer. We get data form project EmLog [3] so we have possibility to compare with their results. Provided data contains activities of 5 users. From these data we can create user's model. In solution we create user's model that contains 5 vectors for 5 groups of emotions. For each particular emotion group is created one vector. Vectors consist of statistically pre-processed values of metrics. We recognize 5 dimensional groups of emotions: neutral, negative excited, negative calm, positive excited, positive calm. We recognize emotions by comparing vectors with statistical method Manhattan distance.

Each recognizing emotion can be detected with certain probability. In first experiment we measure probability value of detecting correct emotion. Probability for each particular emotion is defined as (see Formula 4):

$$probability = \frac{correct\ emotion\ distance}{\sum_{e \in Emotions} distance_e} \qquad (4)$$

where *correct emotion distance* is value of distance between two vectors of one, correct emotion and denominator express sum of all distances between one emotion vector and vectors of all emotions.

In this experiment we recognize emotions by Manhattan distance, which is variously weighting. We compared probability of correct emotions detecting by using various parts of our weighting coefficient. This leads to comparing several coefficients and declares our suggested coefficient as the most successful. It is visible from following graph (see *Figure* ***1***).

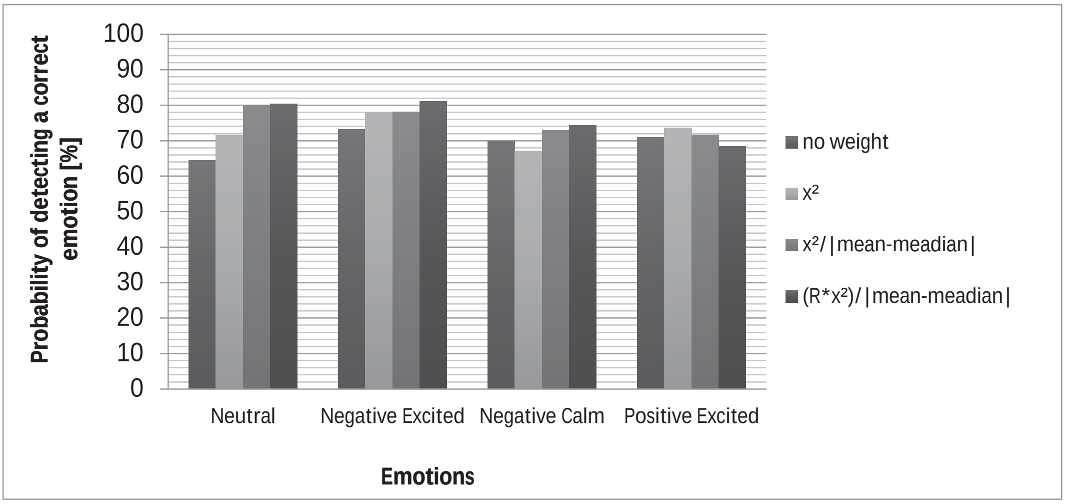


Figure 1. Increasing probability of detecting a correct emotion by using various weights.

Because of that we use data from EmLog, we have opportunity to compare our results with their. EmLog evaluate solution by precision. Precision of each particular emotion is defined as (see Formula 5):

$$precision = \frac{|correct\ recognition\ of\ particular\ emotion|}{|all\ recognition\ of\ particular\ emotion|} \qquad (5)$$

Comparison of precision results in our and EmLog project is visible in following graph (see *Figure 2*). Compared works use different methods for particular steps of solution and it caused differences in results.

In EmLog project they recognized 5 concrete emotions: normal, happy, stressed, tired and frustrated. So we had to toggle their emotions to our dimensional groups of emotions. It is also important to mention that EmLog project use Manhattan distance (in graph specified as VDC) and Cosine similarity (in graph specified as CSC) for emotion recognition. For now, our solution use only Manhattan distance. In our solution, model consisting of vectors contains preprocessed values of metrics from limited count of user activities. EmLog model contains all unprocessed metrics values and their count in structure like histogram.

Our solution using Manhattan distance reach better results than EmLog solution using Manhattan distance. Problem caused only neutral emotion state, where EmLog is better. It shows that problem is influenced by amount of collected activities data preprocessed into model. We plan to realize another experiment with aim to discover the most suitable amount of data.

In case of Cosine distance, EmLog reach more successful results, which ensure their model and Cosine distance method. Our success in negative emotions recognition is influenced by emotion partitioning. EmLog recognized emotions: stressed, tired and frustrated, which are negative and these states can have similar character and can be exchange to each other. Our solution recognizes groups of emotions that can be more expressly.

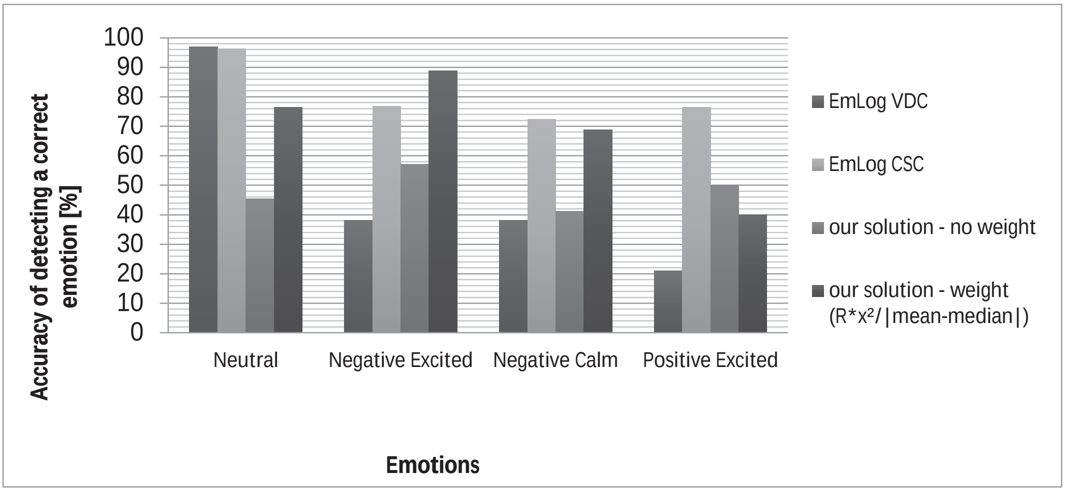


Figure 2. Comparing our solution with project EmLog.

5 Conclusion and future work

In this paper we described user´s modeling for the aim of his emotions identification. We mainly focused on users characteristics that fill user's model. We analyze properties of these characteristics that can influence emotions recognition. After that we set weighting coefficient for improving recognition. We realize experiments, where we compare our solution with another solution. Experiments results prove our contribution on successful emotion recognition.

Other possible improvement in discussed area can be reached by different model structure and various comparing methods.

In the future work we will get other users data and prove our solution on them. By applying solution on bigger dataset we will reach more precise results. Also we will apply another comparing method for emotions recognition, such as Euclid distance.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

References

[1] Epp, C., Lippold, M., Mandryk, L.R.: Identifying Emotional States using Keystroke Dynamics. *SIGCHI Conference on Human Factors in Computing Systems*, (2011), pp. 715-724.

[2] *PerConIK*. Slovak university of technology in Bratislava, Faculty of informatics and information technologies [Online; accessed February 16, 2014]. Available at: http://perconik.fiit.stuba.sk

[3] Gajdoš, J. et al.: Odhaľovanie emocionálneho stavu používateľa. Team project, Slovak university of technology in Bratislava, (2013).

[4] Lv, H.R. et al.: Emotion recognition based on pressure sensor keyboards. *IEEE International Conference on Multimedia and Expo*, (2008), pp. 1089-1092.

[5] Vizer, L. M., Zhou, L., Sears, A.: Automated stress detection using keystroke an linguistic features. *International Journal of Human-Computer Studies*, (2009), vol. 67, no. 10, pp. 870- 886.

[6] ZIMMERMANN, P., et al.: Affective computing – a rationale for measuring mood with mouse and keyboard. *International Journal of Occupational Safety and Ergonomics*, (2003).

Enhancing MapReduce Using Hash Tables and Optimized Data Exchange

Michal DORNER*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`dorner.michal@gmail.com`

Abstract. MapReduce is a programming model for simplified parallel data processing, mostly used for large-scale data sets on clusters of commodity computers. In MapReduce, chunks of input data are mapped to list of key/value pairs, then records with same key are grouped together and reduced to final output. In this report, we propose several MapReduce optimizations, mostly aimed on recursively reducible job types. Proposed methods includes using hash tables for faster grouping and reducing disk I/O by incremental reduction and optimized data exchange. Our experimental prototype shows promising results for two tested applications: WordCount and computing Term Frequency–Inverse Document Frequency.

1 Introduction

Processing large-scale data sets is nowadays needed in many domains such as data mining in web services, astronomy and bioinformatics. Volume of available data is increasing exponentially and sequential algorithms are no longer capable to process such volumes in acceptable time. Hence, the role of parallel and distributed computing models is becoming more important than ever before.

The MapReduce programming model simplifies large-scale data processing on commodity clusters by having users specify a map function that processes input key/value pairs to generate intermediate key/value pairs, and a reduce function that merges and converts intermediate key/value pairs into final results. Many real world tasks are expressible in this model. Apache Hadoop is currently the most known and used MapReduce implementation. Prominent companies such as Facebook and Yahoo runs Hadoop clusters of thousands nodes. While Hadoop excels at scalability, infrastructure and tools, its core suffers from inefficiency in specific task types.

We have identified grouping of records in shuffle phase and high usage of temporary files as main sources of inefficiency in Hadoop. The method described in this paper targets at solving identified problems by using hash tables for faster grouping and incremental reduction and optimized data exchange for reduced disk usage.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

1.1 MapReduce programming model

Input data are first split into separate chunks of typically 16 to 64 megabytes. Each chunk is processed independently in parallel by user provided Map function. Map function takes key (chunk ID, line number, etc.) and input chunk and produces a set of intermediate key/value pairs. Map function have the following general form [8]:

$$map(K1, V1) \rightarrow list(K2, V2)$$

The MapReduce framework in shuffle phase groups together all intermediate values associated with the same intermediate key and passes them to the user provided Reduce function. Reduce function takes an intermediate key and a set of values for that key and produces final output. Reduce function have the following general form [8]:

$$reduce\big(K2, list(V2)\big) \rightarrow list(K3, V3)$$

Reduce invocations are distributed by partitioning the intermediate key space into R pieces using a partitioning function (e.g., hash(key) mod R) [2]. Execution overview of MapReduce job is shown in Figure 1.

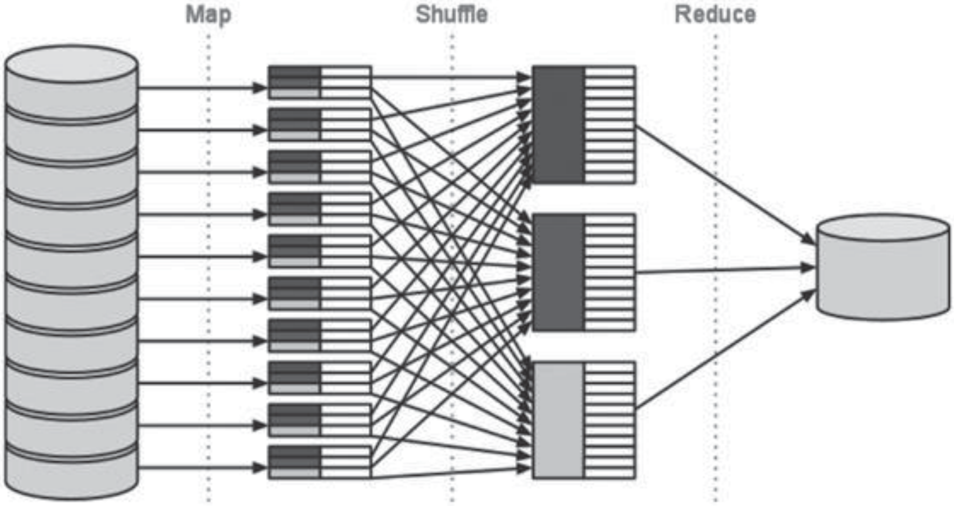


Figure 1. Execution overview of MapReduce job.

1.2 Recursively reducible jobs

Recursively reducible jobs are class of MapReduce jobs where a portion of the map results can be reduced independently, and the partial reduced results can be recursively aggregated to produce global reduce results [3]. In typical MapReduce implementations, there is an optional Combiner function used to reduce network bandwidth by reducing output of single map task before it’s send to reducer.

2 Related work

One of the first usage of hashing in MapReduce was project Mars [4]. Mars uses hashing for speed up sorting. It computes single 32-bit integer hash value for each key (possible more complex data structure) and then it sorts records according to their hash values. Fetching and comparing keys are evaluated only when their hash values are the same. Given a good hash function, the probability of comparing keys is low.

Condie et al. in their paper [1] proposed method allowing data to be pipelined between operators. This extends the MapReduce programming model beyond batch processing, and results into shorter completion times and improved system utilization for batch jobs as well.

Shinnar et al. created new implementation [7] of the Hadoop Map Reduce (HMR) API for in-memory jobs providing significantly better performance than the Hadoop but dropping Hadoop capabilities of resilience and ability to handle more data that fit into ram.

Mohamed et. al implemented MapReduce model using MPI and optimized data exchange policy. In their method reducers starts at the same time as mappers, targeting synchronization barrier between map and reduce phase [5].

3 Grouping records using hashing

Traditional implementations of MapReduce, e.g. Hadoop, are mostly using sort-merge algorithm to guarantee the computation model. Sort-merge method consist of in-memory sorting and either in-memory or on-disk merge of already sorted records. While this works well in terms of handling massive amounts of data which may not fit into memory, sorting is relatively expensive operation when only grouping is needed.

Alternative way to group records with same key together is using hash table. Hash tables allows arbitrary insertions of key-value pairs at constant average cost, which makes grouping of n items cost O(n). In theory, hash tables provides significantly better performance than traditional sorting, which makes on average O(n log n) comparisons to sort n items. However real results may vary depend on number of table resizing, additional cost of table traverse in case of many empty buckets, number of hash collisions and complexity of computing hash compared to cost of key comparison.

Grouping using hash table also results in unsorted output. Unsorted output is not problem if either record order doesn’t matter or if reduction significantly reduces number of records and cost of sorting reduced output is negligible in comparison with cost of less effective grouping using sorting. Limiting factor for using hash table is available main memory. Good performance is only achievable when whole table is loaded into RAM. However, as is shown in next chapter, for recursive reducible job types it’s possible to avoid this limitation while still getting performance benefits of this method.

4 Incremental reduction

Our proposed method is aimed on optimized processing of recursive reducible job types. It’s scalable, effectively handling smaller amount of data that fit into memory and also bigger data sets that don’t. With little modifications it’s usable for both types of reduction: reduction of single map output in map tasks (combiner in Hadoop) and reduction of collected map outputs in reduce tasks. Better performance is achieved by using hash tables for grouping while solving problem of RAM limitation by fallback to traditional sort-merge method on already reduced data if needed. Execution overview of our method is shown in Figure 2. Key ideas of our method are:

1. Using hash table for faster grouping of records.
2. Incrementally reduce records in hash table for keeping RAM usage in acceptable bounds.
3. Sort and flush content of hash table to temporary files when needed.
4. In the end, if there are written any temporary files, process them with rest of data placed in memory same way as in traditional sort-merge method.

Our presumption is this method will perform very well on datasets where map function will produce many records with duplicated keys. Our method will achieve better performance by both faster grouping method and reduced disk I/O.

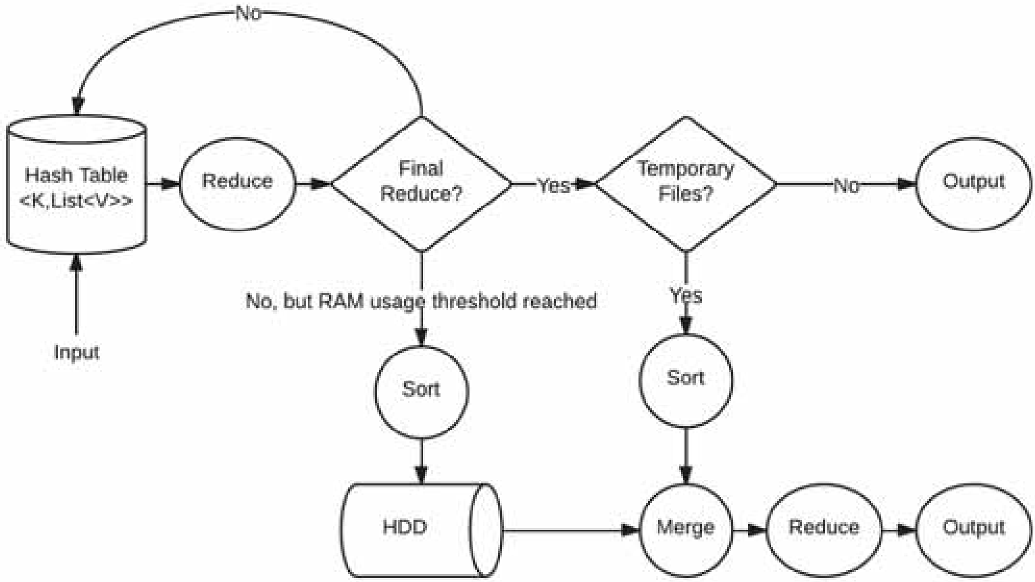


Figure 2. Execution overview of incremental reduction.

4.1 Processing input that fit into memory

Processing input that fit into memory is the most simple case. All input key-value pairs are inserted into hash table (buffer) and whole table is then reduced producing final output. In this case no temporary files are written at all. This scenario is most likely to happen when processing single map output in map task.

4.2 Incrementally reduce buffer

Incremental reduction traverses over whole hash table, replacing list of values associated with key by list with one element – result of values reduction. Incremental reduction of buffer will be triggered when ram usage reach specified limit and not all input is already processed. This could happen in map task when map function produces too much data and in reduce task after collecting multiple map outputs from map tasks. Incremental reduction (and sort-merge of temporary files if there are any) will be also triggered in reduce task after map task in shared computation slots has no input to process.

4.3 Fallback to sort-merge

When RAM usage after reduction reach another specified limit, records will be sorted and written to disk as temporary file. Those files will be later processed with sorted result of incremental reduction in same way as in traditional sort-merge method. Our method should still behave better than traditional sort-merge because sorting is done on data that was already reduced using optimized grouping by hash tables. If this scenario happens on map task, final sorted, merged and reduced map output send to reducer will be not inserted into reducer buffer, but directly stored as it’s temporary file for later processing using sort-merge method.

5 Optimized data exchange

There are two slow components in distributed computing systems composed from commodity computers: network and hard drives. Network bandwidth can be reduced using compression and in case of recursive reducible jobs by applying reduction before transmitting data. This is already done in Hadoop and there is no much space for further improvements.

On the other side, Hadoop uses many temporary files during job execution. Each output of map task is firstly materialized onto disk and later read and send to reduce task. Writing each map output to disk is necessary for effective handling of node failures, but we don't always need such level of robustness and re-reading data that was already in memory before is not necessary at all.

Disks are slow and there may be many simultaneous read/write requests from multiple tasks on same node, multiple remote tasks fetching map outputs and usually node is also in HDFS cluster serving inputs for other map tasks. Each disk operation also consumes CPU cycles which could be better used for map or reduce computations.

Our approach targets mostly on reducing amount of data read from or written to disk and effective data exchange between tasks on same node. Main points where our method differs from Hadoop are:

1. Reduce tasks will run simultaneously with map tasks, sharing the same computation slot. In one time, only map or reduce task will be running CPU sensitive processing. Merging and optionally reducing collected map outputs will have higher priority than processing input chunks in map tasks.
2. Map task will send its output to reduce tasks immediately after computation and writing map output to local disk for node failure handling will be optional.
3. All tasks on single node will run in same virtual machine or execution environment. This will allow efficient in-memory data exchange between mappers and reducers.

6 Evaluation

Our first experiment evaluates performance impact of used grouping method to processing single input chunk in WordCount application. We have tested traditional sorting method, hashing method and sorting after reduction with hashing (fallback mode). Map function tokenize input into words and for each yields {*word*, 1}. Reduce function sums word occurrences and yields {*word*, *sum*} as final output. Since performance of hash table may vary depend on internal implementation, we have implemented our prototype using standard libraries in C++, Java and Erlang. Results for C++ are shown in Figure 3. Results of other implementations were almost identical.

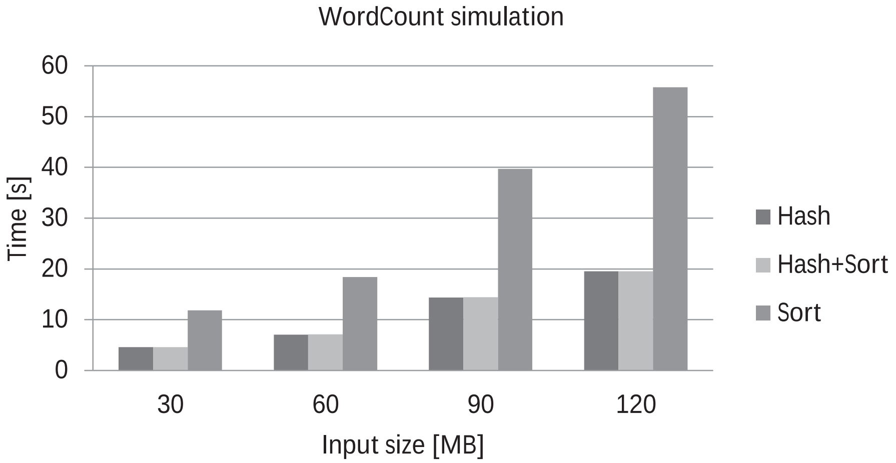


Figure 3. Simulation of processing single input chunk in WordCount application.

7 Conclusions and further work

In this paper we have proposed several optimization methods for MapReduce programming model. Our approach is using hash tables for faster record grouping with combination of incremental reduction and optimized data exchange for reduced disk usage. Main benefits of our method is possible shorter computation time while it still preserves resilience and ability to handle input that not fit into memory.

First experiments show significant performance boost, when grouping using hashing results into more than 2 times shorter computation times in WordCount application. Gap between traditional sorting method and hashing becoming bigger with more data. This behaviour follows their amortized complexity O(n) vs. O(n log n).

Results also shows if reduction significantly reduces number of keys, cost of sorting records for fallback mode is negligible. This proves our hypothesis of performance benefits even in fallback mode for inputs that not fit into memory.

Our further work on this topic will be evaluating performance impact of our optimizations on real cluster solving TF-IDF computation and testing impact of various chunk sizes and threshold settings for triggering incremental reduction. Our current prototype of optimized distributed MapReduce framework is written in Erlang. Erlang was created with distributed computing in mind and many problems of distributed computing are solved in its core libraries and language itself. While this was helpful for creating specialized prototype from scratch, we have discovered Erlang performance and memory usage characteristic not suitable for MapReduce framework in practice. Since Hadoop now implements experimental interface for pluggable sort and shuffle logic, we will evaluate possibilities of implementing our method directly into Hadoop.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0233-10.

References

[1] Condie, T., Conway, N., Alvaro, P., Hellerstein, J. M., Elmeleegy, K., Sears, R.: MapReduce Online. In: *NSDI'10 Proceedings of the 7th USENIX conference on Networked systems design and implementation*. USENIX Association Berkeley, (2010).

[2] Dean, J., Ghemawat, S.: MapReduce: Simplifed Data Processing on Large Clusters. In: *Proc. of the 6th Symposium on Operating Systems Design and Implementation*. USENIX Association Berkeley, (2004), pp. 137-150.

[3] Elteir, M., Lin, H., Feng, W.: Enhancing MapReduce via Asynchronous Data Processing. In: *ICPADS'10: Proc. of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems*. IEEE Computer Society Washington, (2010), pp. 397-405.

[4] He, B., Fang, W., Luo, Q.: Mars: a MapReduce framework on graphics processors. In: *PACT'08 Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM New York, (2008), pp. 260-269.

[5] Mohamed, H., Marchand-Maillet, S.: Enhancing MapReduce using MPI and an optimized data exchange policy. In: *ICPPW '12 Proceedings of the 2012 41st International Conference on Parallel Processing Workshops*. IEEE Computer Society Washington, (2012), pp. 11-18.

[6] Rehmann, K.-T., Schoettner, M.: An In-Memory Framework for Extended MapReduce. In: *Parallel and Distributed Systems (ICPADS)*. IEEE, (2011), pp. 17-24.

[7] Shinnar, A., Cunningham, D., Saraswat, V., Herta, B.: M3R: increased performance for in-memory Hadoop jobs. In: *Proceedings of the VLDB Endowment*. VLDB Endowment, (2012), vol. 5, no. 12, pp. 1736-1747.

[8] White, T.: Hadoop: The Definitive Guide, 3rd Edition. O'Reilly Media, (2012).

Finding and Utilizing Experts in Crowdsourcing Game

Peter DULAČKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
dulacka@gmail.com

Abstract. Crowdsourcing games proved to be great tool for utilizing human computation using fun and challenge as motivation instead of monetary means. Through redundant task of solving and collaborative filtering, all incorrect, biased or invalid artefacts created by players are ruled out during validation process. However, among these, there are also expert-generated correct, and often above-average quality artefacts, we want to preserve. We propose a crowdsourcing game with custom expert finding method to find and utilize experts not only to preserve advanced artefacts, but also to speed up validation process. Conducted experiment proved method to recognize expertise of players in specific domains.

1 Introduction

To get most out of online multimedia services (search engines, personalized recommenders, etc.), proper metadata for each entity need to be generated. Such generation can be provided by: (1) domain expert, (2) automated means or by (3) crowdsourcing. One of the means of crowdsourcing are crowdsourcing games (also known as games with a purpose – GWAP) providing fun as a motivation leaving expensive crowd funding behind.

One of the biggest crowdsourcing games problem is a need to validate all user-generated artefacts (i.e. useful results of the game, e.g. metadata, created by its players). However as the validation needs to be done by another person, whole generation process (until artefact can be used) is slowed down leaving validation to a few options:

– Multi-player game design. Validation is performed by other players right when the artefact is generated by input/output players' agreement [1]. Such games suffer from cold start problem and need to have at least two players online simultaneously or automatic bot present.

– A-posteriori evaluation. Validation is performed later by aggregation results during time period and rating of each generated artefacts – if threshold is met, artefact is validated. Such evaluation can be performed in very same game or passing aggregated data into separate – validation-only oriented – crowdsourcing game [2].

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Jakub Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

These validation methods introduce limitation into the process. Not only they correctly remove incorrect artefacts generated by non-experts, they also remove expert artefacts as the crowd cannot validate their correctness due to lack of expertise in the given domain. The only option to preserve expert artefacts is to find experts in the crowd and alter the validation process to favour their outputs.

In this paper we present method for expert finding in crowdsourcing games based in HITS algorithm [3], being able to recognize experts among players. We present related work in crossover of domains of crowdsourcing games and expert finding, expert finding method, crowdsourcing game it is incorporated into and our evaluation results. We conclude our findings in the end of this paper.

2 Related work

2.1 Games with a purpose

Many games with a purpose deal with artefact validation immediately thanks to multiplayer game design. They generally use [1]: (1) output agreement, (2) input agreement or (3) inversion problem. As multiplayer games suffer from cold start problem, not many were successful in this area.

Other single player games depend on independent output agreement (artefact generation) of players given the same task, such as in [4]. In our previous research we focused on game designed solely for purpose of artefact validation [2], however it succeeded only in recognizing incorrect artefacts and wasn't able to help in validation of expert artefacts.

All of mentioned approaches suffer from not being able to validate expert outputs due to lack of such knowledge of general player base (regardless the game design).

2.2 Authority recognition

Experts in crowd are generally being searched for in: (1) closed corporate environment and (2) open online communities (CQA systems such as Yahoo! Answers[1] or Stack Overflow[2]).

In closed corporate environment there is a handful of expert finding means being used [5]: (1) document changes tracking, (2) domain specific heuristics or (3) expert databases. In [6] Balog, Azzopardi and De Rijke present heuristic method based on the person-document relation. Even when their experiment resulted in precision (of expert recognition) value ~ 0.316, their approach ranked in top 5 proposed approaches for expert finding in TREC[3] conference.

In the open communities, mostly statistical metrics and graph analysis are being used [7]:

- Simple statistical metrics: Such as number of answers.
- InDegree metric: Counting number of edges entering the node.
- Z-score. Counting number of edges entering and leaving the node.
- Pagerank / Expertise rank / HITS algorithm

Expert finding in crowdsourcing games is represented by "capability aligned matching" [8] which extends ESP Game [9] with the capability of evaluation the player's expertise and image annotation difficulty. Game is based on economic model rewarding players according to quality and amount of their previous annotations. However game doesn't solve the problem of removal of expert artefacts.

[1] http://answers.yahoo.com/
[2] http://stackoverflow.com/
[3] http://trec.nist.gov/

3 Expert finding in crowdsourcing game

We created a crowdsourcing game focusing on music metadata generation and validation. Our primary goal was to recognize player's level of expertise (possibly also with expertise threshold determining expert) and utilize it in the process of annotating music. The game was incorporated into online radio Woodstock.FM[4], created for the purpose of evaluation of our method. Radio's playlist was divided by expert into 4 different little overlapping music domains (genres), so we were able to test expertise of players in different domains. Domains are defined by set of representative artists, which were chosen by expert. Playlist is based on this list – one of the core artist is randomly picked and send to online recommendation service for "next song" recommendation.

Game module consist of multiple mini games which are designed to test player's expertise in domain of song the radio is playing. Correctness of most of the answers is directly evaluable as all questions are about facts related to the song or artist. As all players are listening to the same song, we are able to directly evaluate and compare their expertise for given song (artist, genre). Player's goal is to answer correctly as many questions as she is able to.

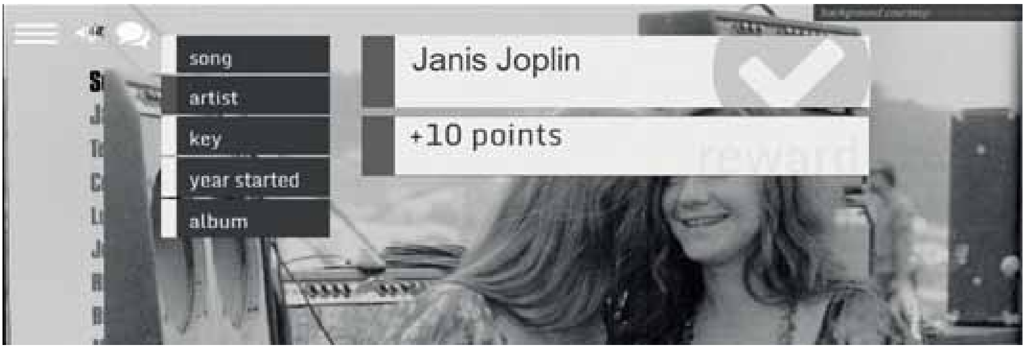


Figure 1. Successful mini game answer in game mode of Woodstock.FM.

Some mini games are designed not to test expertise, but to generate artefacts (metadata) for given song. When song ends, we are able to calculate player's expertise based on current and previous results and predict correctness and need for further validation of generated artefacts. Standard and simplest game scenario goes as follows:

1. Player logs into the radio, which is already streaming some random song. There is no information about song title or artist available for the player. The only hint for player is background, which displays first high resolution result from search engine when artist's name is queried – not always the artist itself, e.g. *The Eagles* displays picture of eagle.
2. Players chooses mini game from the list: artist name, song title, album title, origin country of the artist, year the artist started, musical key, etc. To remove luck factor in authority calculation, we don't provide options to choose from and player has only input box with autocomplete available. Successful answer is displayed on Figure 1.
3. If the player is correct, she receives points for the answer (based on the difficulty of question determined by preliminary experiment results) and token for adding custom song into the radio playlist – all other players would listen to the song and answer the questions related to it. No points are being subtracted. Player front-end points scoring is not related to his expertise score explained in the following section.

[4] http://woodstock.fm/

3.1 Domain classification

To evaluate player's expertise for given domain (not just the artist), we introduced artist-genre distance, which numerically stands for probability that artist belongs to given genre – in our case one of the 4 introduced musical domains.

Our musical domains were defined by lists of core artists. Let each musical domain be represented by a set of weighted annotations. Most of the songs and artists have already available list of validated annotations (we used LastFM API[5] and dataset). We obtained list of weighted annotations for these artists and used them to train Naïve Bayes classifier. To classify the song to one of our domains, we acquire its annotations and pass it to the classifier. Classification is calculated as follows (*C* – class/genre, *F* – classifier/annotations, *p* – probability):

$$p(C|F_1, \ldots, F_n) = \frac{p(C) * p(F_1, \ldots, F_n|C)}{p(F_1, \ldots, F_n)} \tag{1}$$

3.2 Expertise evaluation

Our expert finding method is based on HITS algorithm [3] due to its higher success rate in online communities analysis over other methods. As HITS is based on homogenous entities (users helping users, website referring websites), adaptation of algorithm for heterogeneous entities is necessary (users solving tasks). We introduced set of limitations (minimal requirements) to meet our goal:

- There are two types of nodes in graph: users and tasks.
- Only user-task relation is allowed.
- The relation can be created only when user answers the task correctly.

Meeting this limitations we are able to calculate *authority* and *hub* based on HITS accordingly:

$$aut\boldsymbol{h} = \frac{\sum_{i=1}^{n} \frac{1}{\boldsymbol{h}ub(i)}}{\boldsymbol{h}ub_{\max\square}};\ \boldsymbol{h}ub_{max} \in T: \boldsymbol{h}ub_{max} > \boldsymbol{h}\ \forall\ \boldsymbol{h} \in T \tag{2}$$

After the algorithm converges, there are two values that represent our case:

- Authority of user. Value represents expertise of player in given domain. Higher the better.
- Hub of task. Value represents difficulty of task. Higher the easier.

Considering the result as an expertise of player for given artist, to calculate player's expertise in the domain, we need to calculate weighted arithmetic mean of his expertise for set of artists of domain (*auth* – authority, *D* – artist-genre distance):

$$E_d = \frac{\sum_{artist}\left(aut\boldsymbol{h}(artist) * \frac{1}{D}\right)}{artist_{count}} \tag{3}$$

Such evaluation brings numerous advantages: no need to manually evaluate difficulty of each task (for the game purposes), method is not bound to be used only with music (yet the minimal requirements have to be met), and it can be used outside of crowdsourcing games domain. Method usage does not require simultaneous users' interaction and can be used in single user applications.

The biggest disadvantage of method is need for a valid question-answer set. As the answer is deterministic and verifiable, it needs to be acquired from public source or generated by expert. Incorporating public source into the game makes cheating more available and introduces risks that game design needs to handle.

[5] http://www.last.fm/api

4 Experiments

We picked 20 songs (5 for each of our musical domains) and created questionnaire/quiz for our experiment participants. Musical education of participants varied – from the student of musical studies to laymen. Participants were instructed to fill in the questionnaire without cheating – questionnaire included personal questions such as musical education, favourite artists and their personal evaluation of their skills, and quiz questions such as song title, artist title, year the song was released, similar artists, more songs from given artist and more. Participants did not have option to choose from couple of options and needed to generate answer on their own. They did not know title of song or name of the artist – same as in our game module.

As we evaluated results, we realized that the difficulty of quiz questions were too high (all participants left the answer field in questionnaire blank) and decided to ignore some question types from evaluating. Authority score (numeric representation of expertise) of each participant for each musical domain was calculated as follows in Table 1. Score effectively represents number of correct answers combined with difficulty with successfully completed tasks. Maximum score 1 is achievable only when all questions are answered correctly. Participant #1 was a-priori considered as expert based on his domain education.

Table 1. Authority score of users after questionnaire evaluation.

	#1	*#2*	*#3*	*#4*	*#5*	*#6*
pop	0.5132	0.2812	0.2852	0.3217	0.3073	0.3353
rock	0.4477	0.2155	0.3483	0.3328	0.2778	0.3835
alternative	0.5278	0.1553	0.2798	0.3478	0.0225	0.2451
old/soundtrack	0.2441	0.0497	0.1838	0.0000	0.0256	0.1307

To verify correctness of authority score, we handpicked pairs for further validation in genres *pop*, *rock*, and *alternative*. These users were presented a song of a genre and list of song annotations. Their task was to validate these annotations – to pick the correct and the wrong for given song. We compared their answers with ours expert-generated golden standard created by 3 experts (majority decision). Results are presented in Table 2 (*score* – success rate in annotation validation, FP – false positives, *FN* – false negatives, *FPx* – false positives considering only full agreement of experts, *FNx* – false negatives considering only full agreement of experts).

Table 2. Authority validation of experiment participants.

	score	*FP*	*FN*	*FPx*	*FNx*	*authority*
alternative						
#4	0.83	0.17	0.17	0.11	0.22	0.35
#5	0.70	0.25	0.33	0.11	0.22	0.02
pop						
#2	0.57	0.47	0.36	0.17	0.67	0.28
#4	0.73	0.47	0.00	0.17	0.00	0.32
#5	0.67	0.53	0.09	0.50	0.33	0.31
rock						
#4	0.80	0.38	0.12	0.00	0.00	0.33
#5	0.67	0.31	0.41	0.00	0.33	0.28

In general, users with higher authority score achieved better results in validation music annotations. We observed lower false positives and false negatives score for subjects with better authority score.

We also observed negative impact caused by difference of type of given tasks (need for knowing the answer in questionnaire vs. true/false choice in the validation task). Experiment participant with low authority score for *alternative* genre was able to compete with other participants in validation tasks. As a result, we were forced to alter future experiments not to mix task types when validating subject's authority score.

5 Conclusions and future work

We presented domain expert finding method designed for crowdsourcing game. Game is designed to utilize the method together with artefact generation. In our experiment we were able to recognize players with higher expertise and prove it in metadata validation task confronted against golden standard. Even though we see potential in the method, its greatest disadvantage is need to have dataset all tasks are validated against. As these datasets are publicly available (or expert generated), probability of cheating is higher. However proper game design might remove such risk – e.g., time limiting or competitive question assignment.

After we evaluated experiment result we decided to alter roadmap of the game development to add more lower-difficulty mini games, proposed team confrontation integration into game module and alter all future experiments to prevent mixing different kind of user tasks (custom input vs. choice) to prevent noise in measured results. This was mostly caused by unexpected high difficulty of mini games and lower knowledge of music of crowd in general resulting in unsatisfactory comments during experiment evaluation.

Our future work consist of developing features in radio and game module, and evaluation of change in speed of artefact generation process which we believe will be affected positively.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] L. von Ahn and L. Dabbish, "Designing games with a purpose," *Commun. ACM*, vol. 51, no. 8, pp. 57, Aug. 2008.

[2] P. Dulačka, J. Šimko, and M. Bieliková, "Validation of music metadata via game with a purpose," in *I-SEMANTICS '12*, 2012, pp. 177.

[3] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999.

[4] J. Šimko, M. Tvarožek, and M. Bieliková, "Little search game: term network acquisition via a human computation game," *ACM Hypertext 2011*, pp. 57-61, 2011.

[5] A. Mockus and J. D. Herbsleb, "Expertise Browser: a quantitative approach to identifying expertise," *Proc. 24th Int. Conf. Softw. Eng. ICSE 2002*, pp. 503-512, 2002.

[6] K. Balog, L. Azzopardi, and M. De Rijke, "Formal models for expert finding in enterprise corpora," *... 29th Annu. Int. ACM ...*, pp. 43-50, 2006.

[7] J. Zhang, M. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," *WWW '07*, pp. 221-230, 2007.

[8] C. Chiou and J. Hsu, "Capability-aligned matching: Improving quality of games of a purpose," in *Proceeding AAMAS '11 The 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011, no. Aamas, pp. 2-6.

[9] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, vol. 6, no. 1, pp. 319-326.

Identification of Higher Paraphrasing in Slovak Language

Jozef GAJDOŠ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova2, 842 16 Bratislava, Slovakia
xgajdos@is.stuba.sk

Abstract. This paper deals with the identification of higher paraphrasing in text documents written in the Slovak language. By higher paraphrasing we mean that the paraphrased text was created by using more advanced methods, for instance synonyms replacement. We present here our own approach that uses support vector machine (SVM) and contains several specifics in the pre-processing phase regarding Slovak language. Our solution takes into account the Slovak language characteristics and use of the currently available methods and dictionaries. The main result of our work is the created POS tagger for Slovak language and the way how two sentences and their properties are represented in a vector and then compared on similarity. The carried experiments clearly show that our approach is working well.

1 Introduction

Today large volume of data can be found in an electronic version that is usually available online on the Internet. Most of these data are texts written in natural language. Users often have to search for relevant content on the web based on keywords, phrases or by selecting different categories. Being able to query such data and detect similar content within different texts is therefore crucial these days. Such similarity detection on higher level is however an open problem. To the open problems in detecting similarity belongs the identification of paraphrasing at a higher level. By a paraphrasing on a higher level we refer to the combination of changes in word order, inserting or removing words from sentences and using synonyms.

Paraphrasing is basically a way of saying something in different words while the original idea and its importance during paraphrasing do not change. There are many ways how to paraphrase text in Slovak language. Moreover, Slovak language belongs to the group of inflecting languages. Core vocabulary consists of 60 000 words, has a complex syntax, morphology, and is quite ambiguous.

Identification of paraphrasing in natural language can be helpful in identifying plagiarism or detecting topic similar documents. Such identification can be for instance useful when clustering text data or in case we perform knowledge summarization.

[*] Master degree study programme in field: Software Engineering
Supervisor: Tomáš Kučečka, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

This document is structured as follows. Section 2 describes the existing work in the field of plagiarism identification. In sections 3 and 4 we describe and experiment with our own approach for plagiarism detection. Section 5 summarizes this paper.

2 Related work

In papers [1], [2] and [3] authors introduce an approach to identification of paraphrasing through using the machine learning. The granularity of identification was on sentence level. All proposed methods were designed for the English language and tested on Microsoft corpus, which has marked paraphrased sentences.

In paper [1] are described experiments with machine learning algorithms, SVM and k-nearest neighbour. As a distance measure authors used the normalized Manhattan distance and maximum entropy. Input to the machine learning algorithms were pairs of sentences with the following features:

- the count of matching words between sentences to the total number of words,
- overlap of 4-skip-gramsbetween sentences,
- overlap of sequences in sentences found by LCS (longest common subsequence) algorithm,
- semantic similarity between words,
- sentence containing its own noun received flag 1, else the flag was 0.

In several experiments authors investigated the effect of these features(LCS, n-grams and skip-grams, semantic overlap, etc.). The following table (Table 1) shows the overall results. These experiments showed that the most appropriate technique to identify the paraphrasing is the SVM.

Table 1.Precision and recall for used types of machine learning.

Machine learning algorithm	k - nearest neighbor	Maximum entropy	SVM
Precision	*64.68%*	*66.44%*	*70.43%*
Recall	*71.13%*	*70.50%*	*74.12%*

In paper [2] it is also stated that the SVM is the most appropriate approach to the identification of paraphrasing. Input to the SVM represented:

- string similarity
- count of matching words
- Levenshtein distance between words,
- stems of words,
- marking semantically similar words (synonyms).

The authors used linear kernel with the following parameters –constant10^{-3} and complexity 0.5. The results showed that the overlap of phrases and labelling of semantically similar words significant impact on the precision.

In work [3] the proposed solution also used the SVM where the input to the learning algorithm represented a pair of sentences with following features:

- Overlap between sentences determined by the skip-gram method.
- Overlap, which is determined by the LCS algorithm calculated as the ratio of the number of words found by LCS to the number of words in each sentence.

- The semantic similarity of verbs and nouns in sentences. Semantic similarity was determined by the semantic content of the word, the likelihood of its occurrence and information from WordNet.
- Similarity of numeric attributes in sentences.
- Presence of own nouns in a sentence.

In experiments authors used the linear and the Gaussian kernel function. The authors state that the addition of lexical and semantic information increased accuracy of plagiarism identification. The resulting precisions amounted to 70%.

3 Our solution

We designed our solution to the identification of paraphrasing based on an analysis of existing approaches that were described in the previous section. The main difference is that we focus on Slovak language. Therefore, we have to take into account the specifics of this language, which can be seen mainly in its pre-processing stages. We decided to use support vector machines, because the paper [1] shows their relevance in identifying paraphrasing in natural language.

Input to our method is a textual document that is first converted into a plain text. This text is then chunked on sentences. The two documents are compared on sentence level which means that we detect paraphrasing within pairs of sentences. In the following subsections we give a detail description how our method works.Identification of paraphrasing.

In the first step we determine various properties of the sentence as its length and modality. In the second step individual sentences are divided into words and these words are assigned to individual properties. The words are assigned morphological tags, their lemma and common synonym are determined and marked if they are a stop-word.

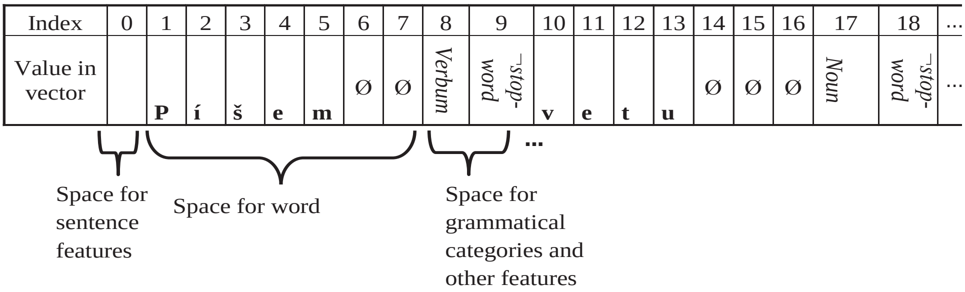
Subsequently, for every pair of sentences we determine their overlap on semantic, morphological and lexical level through:

- LCS algorithm,
- 3-grams where 1-gram is one word,
- 4-skip-grams where 1-gram is one word,
- simple overlap-count of words equals a minimum count of words in sentences.

3.1 Vector representation

The key step of our approach is the way how we represent pair of sentences with their properties in a vector. This vector is then handled by the SVM. In the first part of the vector are the normalized attributes that the two sentences overlap. These attributes are followed by the two sentences. In the first part of each sentence are the properties of a sentence, followed by the words of the sentence. Individual words together with their properties inserted one character after the other, as in the original sentences. For every word there is a fixed number of items reserved in the vector. If a word has fewer letters than these items, the remaining items are filled with zeroes. If a word has more letters than these items, the rest of the letters will be discarded. Length of these items is third quintile of lengths of words in ordinary Slovak text. Similarly, if a sentence contains fewer words than spots reserved in the vector, the remaining spots of the vector are zero. An example of such vector is in Figure 1.

The final step is to evaluate paraphrasing using support vector machine. SVM is a binary classifier, which returns whether it is a paraphrase or not.



Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Value in vector		**P**	**í**	**š**	**e**	**m**	Ø	Ø	*Verbum*	*-stop_ word*	**v**	**e**	**t**	**u**	Ø	Ø	Ø	*Noun*	*-stop_ word*	...

...
Space for sentence features
Space for word
Space for grammatical categories and other features

Figure 1. Example how two sentences and different features are represented by a vector. The text in the vector "Píšem"is one word and"vetu" is the second word.

3.2 POS tagger

POS tagger assigns part of speech to words from sentences. In its process it uses hidden Markov models and the Viterbi algorithm. Viterbi sequence of words in a sentence assigns the most likely sequence of morphological marks.

Viterbi algorithm requires a set of initial states probabilities (the probability that the sentence starts with the word, which is part of speech), transition matrix (expresses the probability of transitions between morphological markers) and generation of output probability matrix of part of speech *t* from the word *w*. Just not likelihood of generating output brands we replaced *tagWordPropability(1)* function, which determines the probability.

$$tagWordPropability(w,t) = \begin{cases} M_1[w,t], & if \exists M_1[w,t] \\ M_2[suffix(w),t], & if\ \exists M_2[suffix(w),t] \\ P(t), & else \end{cases} \tag{1}$$

Where *w* is a word and morphological tag (part of speech) *t* calculates the probability that the word belongs to the part of speech. $M_1$is a matrix containing the probability that a word *w* belongs the part of speech *t*. $M_2$is a matrix containing the probability that any word ending with *suffix(w)* is of speech *t*. Function *suffix(w)* returns the suffix of *w* - the last three letters of the word *w* and *P(t)* returns the probability of occurrence of speech *t*. The length of suffix was determined experimentally.

Matrices M_1, M_2 and function *P(t)* are created with a corpus of *r-mark* (Hand morphologically annotate corpus), which is part of the Slovak national corpus provided by the Language Institute of Ľ. Štúr Slovak Academy of Sciences.

4 Experiments

Our solution was verified on two data sets. The first dataset is a manually annotated corpus (r-mark), which was created with a help of the *Slovak Academy of Sciences*. In this corpus there are to each lexical unit assigned its grammatical categories. Currently the corpus contains 77 755 phrases consisting of 1 199 224 lexical units.

The second dataset was manually created by us and it is used for identification of paraphrases. It contains 1 350 pairs of sentences where half of these pairs contain two sentences that are a plagiary of each other. The rest of the pairs are those in which the first sentence is not a plagiary of the send sentence in the given pair.

Further in this section we describe two experiments. In the first experiment we evaluated the performance of our own POS tagger.

4.1 POS tagger performance

In this experiment we evaluated the performance of our own POS tagger that we use to identify morphological tags in the pre-processing phase. The overall accuracy of this POS tagger reached up to 95.35%. Table 2 shows the precision and recall for all speech and other morphological tags that our POS tagger identifies. The results were achieved on the r-mark dataset.

Table 2. Results that we achieved by using our POS tagger to identify morphologic tags.

Morphologic tag	Precision	Recall
noun	*94,04%*	*97,66%*
adjective	*94,42%*	*92,00%*
pronoun	*95,74%*	*98,41%*
numeral	*97,97%*	*91,16%*
verb	*95,57%*	*97,84%*
adverb	*91,67%*	*85,57%*
clutch	*92,39%*	*92,73%*
particle	*82,95%*	*82,07%*
interjection	*94,54%*	*56,05%*
participle	*98,30%*	*54,04%*
reflexive verb	*98,34%*	*98,70%*
conditional morfen	*99,91%*	*99,88%*
full stop, comma, question mark, exclamation point	*99,81%*	*99,80%*
other	*85,15%*	*29,92%*

4.2 Paraphrasing identification

In this experiment we evaluated the performance of the identification of paraphrases on the second dataset (the one that contains manually created pairs of sentences). In the comparison phase of our solution we used six different kernels with the SVM. Based on the experiments results as the most appropriate turned out to be Gaussian and linear kernel.

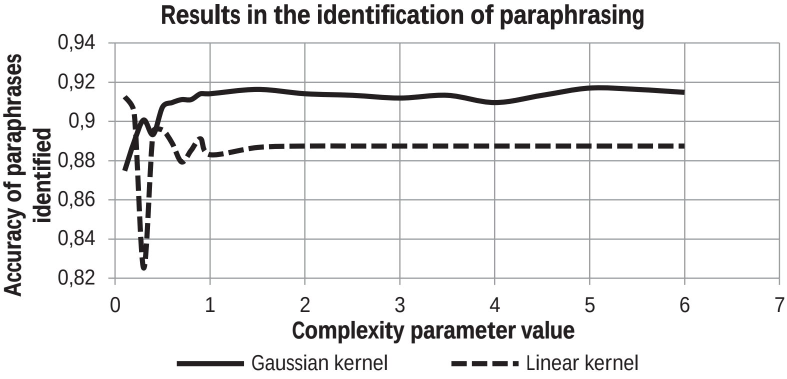


Figure 2.Results of paraphrasing identification based on complexity parameter.

Overall, we obtained the best results (accuracy 90%) when using the Gaussian kernel with *complexity* ranging from 0.2 to 0.7.Other very good results, about 85 to 90%, were achieved by using a linear kernel with parameter *constant* set to value 2.0. The results are depicted in Figure 2.

5 Conclusion and future work

In this paper we have proposed a novel approach for identification of paraphrases in Slovak language. The most significant parts of this approach are our own POS tagger that we use in the pre-processing phase and the way how we represent and evaluate plagiarism between two sentences using SVM.

We managed to achieve 95.35% accuracy with our POS tagger in identifying morphological tags. We consider this as an important enhancement compared to the original approach that scored 87%.Overall, in the area of plagiarism identification we achieved good results when compared with the existing approaches that identify paraphrasing in English language. The best results in our experiments were obtained for the linear and Gaussian kernel and varied from 85% to 91.70%.

In the future we plan to integrate into our solution watching of information overlap that, in addition to overlapping phrases takes into account how much of the informational value of the sentence overlaps. For now, we begin to experiment with the calculation of entropy and tf-idf.

In real situations it may happen that the person who paraphrases divides the original sentence into several sentences. Therefore, the identification of a paraphrased sentence has its limitations. We think that this problem can be addressed by comparing one sentence to two or more sentences by compounding these sentences.

References

[1] Kozerva, Zornitsa. 2006. Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. s.l. :*ACM, 2006.*

[2] Brockett, Chris a Dolan, William. 2005. Support Vector Machines for Paraphrase Identification and Corpus Construction. s.l. :*Natural Language Processing Group Microsoft Research, 2005*.

[3] Chitra, A. 2010. Paraphrase Identification using Machine Learning Techniques. s.l. : ACM, 2010. Zv. *RECENT ADVANCES in NETWORKING, VLSI and SIGNAL PROCESSING.* ISBN: 978-960-474-162-5.

[4] Lintean, Michain. 2009. Paraphrase Identification Using Weighted Dependencies and Word Semantics. Informatica. s.l. :*ACM, 2009*.

[5] Microsoft. WCF Extensibility Guidance. [Online] Microsoft. [Date: 25. 8 2013.] *http://msdn.microsoft.com/en-us/library/gg132853.aspx.*

[6] Nemirovsky, Danil a Dobrynin, Vladimir. 2008. Word importance discrimination using context. s.l. :*TREC, 2008*.

[7] Sangeetha, R. a Kalpana, B. 2010. Optimizing the kernel selection for support vector machines using performance measures. *New York : ACM, 2010*. ISBN: 978-1-4503-0194-7.

[8] Taria, Hirothosi a Haruno, Masahiko. 1999. Feature selection in SVM text categorization. s.l. :*ACM, AAAI, 1999*. ISBN:0-262-51106-1.

[9] Vapnik, V. N. 1995. The Nature of Statistical Learning Theory. *New York : Springer, 1995*.

Weighted Vector User Model for Movie Recommendation

Ondrej KAŠŠÁK *

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
ondrej.kassak@stuba.sk

Abstract. In this paper we propose novel user model specialized on modelling user preferences in movies domain. Based on this model we propose application of multiple personalized recommendation approaches. More precisely we introduce the hybrid recommendation method and the content based method. Both of them use the advantages of proposed innovative principle of model parts weighing to reach the more precise recommendation results. In content based method we show advantages of model weighting, in our hybrid recommendation method we focus mainly on increasing precision on top result positions what is quite a demand in target domain. Our approach is evaluated synthetically on dataset composed of thousands of users, which rated the hundreds of thousands items.

1 Introduction and related work

Watching television is nowadays very popular activity. Experiments[1] show that the average U.S. citizen watches TV daily for more than 5 hours. This means 9 years of life. Average schoolchild spends by watching TV 1200 hours per year, while in the school spends in the same time 900 hours. These numbers say that this kind of activity represents appreciable life part of many people, what is the great motivation to do research in this domain. Our aim is to provide watcher the content which will interest him/her. In the age of smart TVs we face the situation, when there are a lot of movies available and user needs to pick up something he/she will watch. The main problem increasing there is the information overload problem, when user easy find something acceptable to watch for he/she, but it is very difficult to find something he/she would like to watch.

The widely used approach to help user decrease the mentioned problem is a personalized recommendation. It however needs to know something about user preferences or his/her past activities in target domain. For these purposes is common to use some kind of the user modelling.

Senot et al. defines, in general, user model as a set of pairs – category, value. In their concept, the category importance is for modelled user expressed by value, proposed as a real number from the interval <0; 1>. The user model is formed by continuous collecting of user's activity [9]. The work of O'Connor et al. shows approaches which have been used to the user

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Michal Kompan, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] BLS American Time Use Survey, A.C. Nielsen Co., 2012.

modelling successfully in the past. The next sections, which show most commonly used techniques, are based on the classification created by these authors [8].

First group of used approaches are Bayesian networks based on training data with the decision tree in each node [2]. Second approach is represented by user clustering, where users are organized into groups with the greatest degree of mutual similarity [10]. The third approach bases on graphs. There are users represented by vertices connected by weighted edges representing the degree of mutual similarity [1]. The fourth widely used approach models user preferences as a set of vectors, which is the most widespread way [6]. Our solution is based on vector user model too.

Vector user modelling paradigm captures the metadata which describe seen movies (in the movie domain) from multiple points of view. Yu et al. proposed user model considering genres, actors and keywords [11]. Mukherjee et al. proposed model based on genres, actors, actresses and directors vectors [7]. Ferman et al. used in their system model considering only genres vector [5]. Vector model composed of actors, directors, genres and keywords, where each vector element has assigned its own weight, used to recommendation Debnah et al. [4]. Mentioned systems using vector user model proved the popularity of this kind of model paradigm.

Existing models however, do not sufficiently express which parts of model deem individual users as crucial when selecting movies. For this reason we propose a solution using vector approach capturing user's preferences form multiple points of view, which we extend by the concept of weighting of individual vectors. Every person is unique and puts the different importance to individual information from user model. Somebody chooses movie primarily by director, other person rather by its content. Thus it is appropriate to find out what influence levels have individual parts of model.

By understanding users' similarity not only by selected movies, but also by selecting method, we should improve the value of information concluded in user models. For this reason, the proposed user model stores, in addition to preferences of vector elements (e.g. concrete genre or actor), also the importance weight of whole vector (e.g. genres vector or actors vector). Based on this information we can identify for example that some user chooses movies mostly by directors, where he/she has got favourite "Hitchcock" and "Spielberg".

2 User's preferences modelling

In order to model user preferences we propose to use specific types of metadata widely available in target domain, which helps us to express user's preferences. Four of them describes movie characteristics (genres, keywords, directors and actors), fifth describes movies themselves (movie identifiers, e.g. unique names).

Our main contribution is the addition of importance weights to vectors in proposed user model. Weights are used to express level of user's interest in information type contained in the individual vectors and their importance in movies selection process. They for example express that user primarily chooses movies by directors and by genres, but he/she doesn't care about keywords.

Weights are calculated from two parts – the initial static weights and individual weights. Initial vector weights are equal for every new user. They represent average vector importance of all users. Individual vector weights capture the variation between modelled user activity and the average activity of all users. They are possible to use in process of search for similar users or for appropriate movies.

2.1 Model components

Proposed user model for domain of movies consist of five basic components (descriptive metadata vectors and user's activity vector) – genres, keywords, directors, actors, items IDs, describing different views on user's preferences in target movies domain. Every vector consists of several elements, which are represented as triples – name, value, weight. Name expresses which vector element we work with (e.g. 'action' (genre), 'summer' (keyword) or 'cumberbatch' (actor)). The value reflects an average of modelled user’s ratings of movies containing actual element in their metadata. The weight means the number of element ratings realized by modelled user.

User’s personal preferences are stored to the model sequentially as he/she made activities (rates movies) in the target domain. After user rates a movie, this rating is added to model to all its metadata elements. That means, in the model is updated every movie genre (concrete genre e.g. ‘comedy’, ‘historical’), keyword, actor, director and the movie identifier - Figure 1.

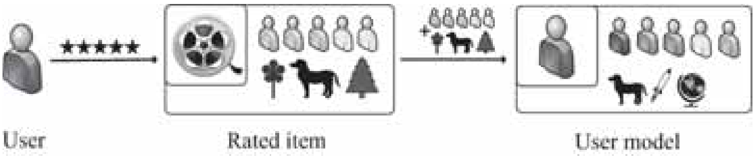


Figure 1. Principle of user modelling. User rates movie, which is described by some metadata. Rating is added to user model for each metadata element (concrete genre, actor etc.).

After user rates the movie, in the user model changes each of its metadata element values (genres, actors, etc.). Adjustment involves the calculation of element value by adding new element rating and the increasing of element weight by 1.

In case, the user revaluated the movie he/she has rated in the past, element attributes change little differently. There will be added only the difference between new and old rating to the element value and there will be no increasing of element weight in this case.

2.2 Preference modelling

From the basic vector user model, it is possible to determine the user's interests. In this paper, however, we add to each vector a factor indicating how much the user decides by the information contained in that vector. It is known that someone decides primarily according to the director, someone determines rather by genres. Our aim is therefore to find out how important individual vectors for the specific user are.

As the first step of weighting process we established initial static weights. Their values are the same for each new user and help us get through the cold start problem. They are calculated as average coefficients from all users.

In the second step of weighting process we adjust initial weights by personal user preference weights. This information says how preferences of modelled user differ from average. Similar principle of user model weighting was previously used in the field of multimedia content by Ferman et al. [5]. They calculated the sum of deviations as division, but we prefer the subtraction operation, because we think that it captures the observed deviations more realistically.

3 Recommendation

Proposed user model has been designed to have the widest possible usage. Actually it is possible to base on it for example the collaborative or the content based recommendation approaches. Both of these two need to their work the information about user’s preferences in target domain, which they can get form vector user model. In advance, usage of weighting principle allows to enhance the knowledge acquired form user model and thus to make a more quality recommendation. When we know what kind of information (in meaning of directors, actors, genres) has for user bigger importance when he/she chooses movies, we can include it to the recommendation process and offer him/her the more accurate results.

Mentioned pure recommendation approaches are well known and were described by multiple other authors. So in this paper we experiment with hybrid recommendation method combining multiple pure methods (all based on proposed user model) into only one process. Hybrid recommendation approaches were well described by Burke [3], who classifies the seven basic hybrid types. Our method can by classify as mixed hybrid method, thus it combines results from collaborative and content based approaches into one common list. The main recommendation process is divided into two steps.

In first, we use collaborative recommendation. There we find, to the target user, the *N* most similar users taking into account the vector weights. So if we recommend to user who likes the most the directors, we look for users who like the most the directors and form them we find out these, who prefer the same ones as target user. Next we get movies which similar users liked and target user didn't see before. Collaborative approach is in our hybrid method the only one who chooses movies which are possible to recommend.

The second step of main recommendation process is the content based one. There we consider items from first step and partially reorder them according to level how are their metadata similar with top metadata elements form individual vectors of user model. When determining similarity, we consider weights of vectors. So we will more prefer items that are similar in metadata elements according to higher weighted vectors.

4 Evaluation

We performed several experiments in order to investigate the performance of recommender approaches using proposed user's preferences modeling. We experimented with collaborative, content based and hybrid methods. In experiments, we used 3 000 users, who performed at least 20 ratings each. Together they rated more than 300 000 movies. Simulation was carried out in a way that we have for each user randomly assigned his/her ratings to the 80% training and 20% test set. From the training sets, we created user models. Recommended items were compared with the movies form test sets, which were in this case replacing user's real choice.

For evaluation of the compared methods, we chose a standard precision metric, which represents the proportion number of items selected by user from the recommended set and the total number of recommended items. More specifically, we investigated the precision of 1, 3, 5, 10 and 15 recommended items.

In first experiment we aimed to verify user model in usage with single content based recommendation. There, we for each user get his/her *N* most preferred elements from each vector in user model. Then we choose from all movies, excluding these from the training set, the movies containing in its metadata some of selected metadata elements (concrete genre, actor or director).

In this experiment we used two variations of content based recommendation method. The first one, which didn't consider vector weights, orders the described set of movies only based on number of searched metadata elements that individual movies contain. After ordering process, we recommend to user top *N* items form this list.

As the extension of this method, we applied on metadata elements the weights, based on user model vector they belong into. By vector weighting principle described in Chapter 2, we set vector weights to 0.12 for genres, 0.06 for keywords, 0.25 for directors and 0.35 for actors. We consider these vector weights for individual metadata elements in process of ordering suitable items, which belongs to content based recommendation.

From experiment result is obvious, that content based recommendation using proposed user model with different vector weights achieves the higher precision in comparison to the same recommendation method based on unweighted user model (Figure 2).

In the second experiment we combined results of collaborative recommendation using proposed vector user model with the content based recommendation approach, described in first experiment, into one hybrid recommendation method. Aim of experiment was to find out how precise can be recommendation method built on proposed user model. As first, we recommend movies collaboratively. In next phase of hybrid process, we reordered results from collaborative method based on their suitability for content based approach (Figure 3).

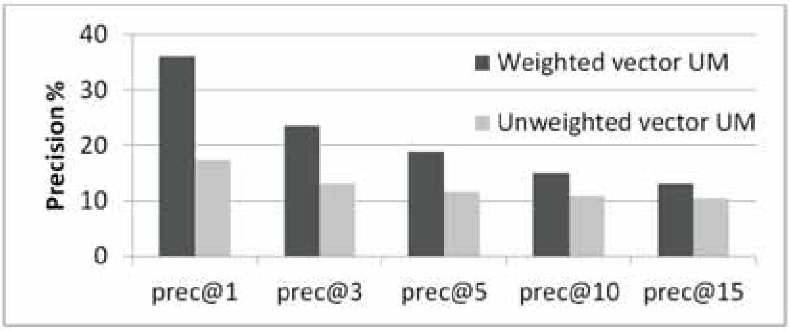


Figure 2. Results of the experiment comparing the precision of content based recommendations using weighted or unweighted user model. Results show precision for N recommended items.

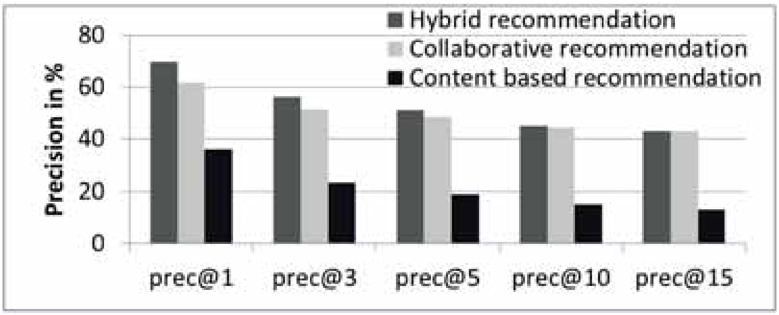


Figure 3. Results of the experiment combining collaborative and content based approaches based on proposed user model into one hybrid method. Results show precision for N recommended items.

We can conclude that recommendation based on proposed user model achieves higher precision than methods based on unweighted user models. The largest improvement can be seen when recommending less number of items, what represents in the domain of movie recommendation the ideal situation. In this domain, it takes relatively long time to play an item and the user usually watches only one movie per session. It is therefore unnecessary to recommend him/her too many items and is much better to offer him/her less number of more appropriate items.

5 Conclusions

In this paper we present the weighted vector user model specialized on movies domain and the possibilities of its usage in multiple methods of personalized recommendations. Our proposal bases on widely used vector approach, which we extend by new idea of personal vector weighting. Proposed model is composed of five vectors – item genres, keywords, actors, directors, and item unique identifiers. Vector weights are calculated from the static preference values and also from individual variations of user preferences to the average of a large number of users. They express level of user's interest in information included in the individual user model vectors in the process of selecting movies. Model is designed to be usable in wide scale of recommendation methods.

We evaluate it indirectly by the content based and the hybrid personalized recommendation methods. We watched mainly the precision of recommendation based on our model in comparison to recommendations based on unweighted vector user models. Recommendation using proposed user model reaches the significantly higher results and we were able to build an effective hybrid recommendation method based on this model. Main improvement occurred when recommended less number of items, which is in target domain ideal situation, due to the long duration of items.

Significant improvement of precision on top positions was possible to observe mainly in hybrid recommendation, where was caused by appropriate combination of multiple pure recommendation methods based on proposed user model.

The model is actually designed for domain of movies. Its principle is however generally usable for any domain, in which are items descriptive by multiple metadata elements. Our future goal is to verify it in another target domain. As example we can propose the domain of cultural events such as theatre and opera plays or for example musical concerts recommendation. It is followed by our next goal, which is to apply our user model and the single user recommendation to groups of users which is in described domains quite desired, because multiple users like to perform some actions in same time and together.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Aggarwal, C., Wolf, J., Wu, K., Yu, P.S.: Horting hatches an egg: A new graph- theoretic approach to collaborative filtering, In Proc. of the 5th ACM SIGKDD Int. Con. on Knowledge Discovery and Data Mining, San Diego, CA, (1999), pp. 201-212.

[2] Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering, In Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence, San Francisco, CA, (1998), pp. 43-52.

[3] Burke, R.: Hybrid web recommender systems. The adaptive web, Springer Berlin Heidelberg., (2007), pp. 377–408.

[4] Debnath, S., Ganguly, N., Mitra, P.: Feature weighting in content based recommendation system using social network analysis. In Proc. of the 17th Int. Conf. on World Wide Web (WWW '08). ACM, New York, NY, USA, (2008).

[5] Ferman, A. M., Beek, P. L., Errico, H. & Sezan, I.: Multimedia content recommendation engine with automatic inference of user preferences. ICIP (3), (2003), pp. 49-52.

[6] Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm, UCB ERL Technical Report M00/41, (2000).

[7] Mukherjee, R., Sajja, N., Sen, S.: A Movie Recommendation System - An Application of Voting Theory in User Modeling. In proc. of UMAP, (2003), pp. 5-33.

[8] O'Connor, M., Cosley, D., Konstan, J. A., Riedl, J.: PolyLens: a recommender system for groups of users. In Proc. of the 7th Conf. on European Computer Supported Cooperative Work, Kluwer Academic Publishers, Norwell, MA, USA, (2001), pp. 199-218.

[9] Senot, C., Kostadinov, D., Bouzid, M.: Analysis of strategies for building group profiles. User Modeling, Adaptation, and Personalization, Springer Heidelberg, (2010), pp. 40–51.

[10] Ungar, L.H., Foster, D.P.: Clustering Methods for Collaborative Filtering, In AAAI Workshop on Recommendation Systems, Menlo Park, CA, (1998).

[11] Yu, Z., Zhou, X., Hao, Y., Gu, J.: TV Program Recommendation for Multiple Viewers Based on user Profile Merging. User Modeling and User-Adapted Interaction '16, Springer Netherlands, (2006), pp. 63-82.

Information Retrieval and Navigation in Heterogeneous RDF Graph

Matúš MICHALKO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
matus.michalko@gmail.com

Abstract. Data obtained in RDF graphs contains a lot of valuable knowledge. Unfortunately, according to the nature of RDF format, which was designed mainly for machine processing, effective access and obtaining of this knowledge might be too complex and complicated task even for technically educated users. On the other hand, elementary principles of information representation in semantic repositories are quite simple and understandable even for people without previous education. In this paper we describe Tripleskop, web tool aimed at comprehensible raw RDF data visualization and visual based SPARQL query construction. We focus at supporting of users actions during identified data exploration activities and information retrieval process. Proposed tool is completely domain independent and can be used for exploration and visualisation of any heterogeneous RDF repository.

1 Introduction

Amount of linked data in recent years increased significantly. This kind of formalized knowledge allowed authors of web content describing the semantics of published information in more fine-grained way. Main idea of information representation in web 3.0 resides in its serialization to semantic graph. The most atomic elements in this graph are resources, that represent real world entities and literals, which contains atomic measurable information such as numbers, or labels. These atomic elements are interconnected to other ones by resources named predicates. Predicate assigns oriented relationship between connected resources, or resource and given literal value. Single predicate relation between resource, predicate and its value is called *triplet*. Visual example of such an information presentation can be found in the Figure 1, where we can identify 3 simple triplets.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Miroslav Líška, Consultant: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 68–73.

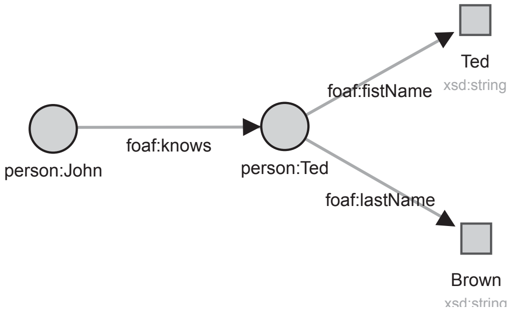


Figure 1. Example of visualization of simple linked data.

In this figure, we can observe the simplicity of information representation in web 3.0 environment. Data obtained in semantic repositories of rich information systems can consists of millions of such triples. Information and knowledge described in RDF graph is often too dense, relationships between some certain resources can be too numerous and direct interaction by human can lead to confusion [4]. Moreover, if there is lack of schema describing stored data (e.g. classes in the ontology, labels, domain and ranges of predicates etc.), searching for concrete piece of information or knowledge in such large and heterogeneous graph can be compared to searching for the needle in a haystack.

As mentioned earlier, data stored in RDF was primary designed for machine processing. However, if we are able, to visualize this stored content in a way, that allows efficient explorative approach and information retrieval, we can bring the power of semantic knowledge closer to people without necessary knowledge from domain, where the search takes place, or even to people without necessary detailed technical education from area of semantic web. In this paper we introduce Tripleskop, web based workbench focused at raw large scaled RDF graphs visualization, exploration and visual way of SPARQL query construction.

2 Related work

Quite respectable progress in a field of visualization of networks and graph data has been made. Wide variety of tools for real time graphs visualization that consists up to ten thousands of nodes and up to one million edges is currently available. However, RDF graphs are specific subset of graph data with some specific characteristics. Common techniques used in big data visualization are not well suited for large knowledge RDF graphs. Some authors even came to conclusion, that many RDF visualization tools threat linked data as a big fat graph that somehow needs to be displayed directly in its graph form although this single form of data visualization might not be sufficient [1]. From the other point of view, visualization of RDF directly in a graph layout has many advantages (e.g. faster understanding of relationships between given resources). Various graph layouts has been presented to allow well-arranged linked data visualization, such as common force-directed layout, best suited for cyclic graphs rendering, various tree layouts for hierarchies visualization, or even innovative layouts such as reduced square layout [3]. Most of these tools focus at identified difficulties of raw RDF visualization such as information overload, advanced filtering (zoom) and search context loss prevention [9]. Weakness of these tools is often they immature and less user friendly user interface and suitability just for presented datasets and domains. Many interesting features have been presented to made RDF visualization as efficient as possible, however, most of examined tools are still in state of academic prototypes. As negative we also consider insufficient experimental evaluation of presented tools in various domains.

Many tools focused on information retrieval from RDF are based on explorative approach. Faceted browsing is nowadays common example of this king of search and can be found even in some tools built upon semantic databases, e.g. Rhizomer [5] or Factic [8]. Disadvantage of this kind of tools is that they quite often rely on some schema (ontology) existence, or they are focused at search only in given domain [7]. Advantage of this approach resides in common user interface reuse, thus user of this tool does not need to know that search takes place in large semantic RDF graph. On the other hand, mentioned approach might not offer as precise and fine-grained search query construction as tools aimed directly at query construction. Languages for querying of graph repositories such as SPARQL are based on specifying of graph patterns, which can be compared to smaller subparts of stored data with some resources and literals substituted with variable nodes or edges. Due to the graph nature of these query languages, visual way of query construction seems to be suitable even for technical users without deeper knowledge of RDF. Example of such query, which returns all persons that know persons with last name "Brown" can be seen in the Figure 2.

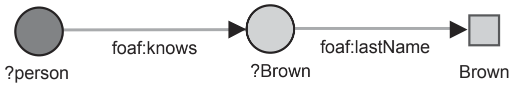


Figure 2. Simple visual example of SPARQL query.

Tools allowing visual query construction have many advantages, e.g. constructed query is always syntactically valid and user does not need detailed former technical knowledge about principles and syntax of given underlying query language [6]. Process of query construction is quite straightforward, thus more easily supported and assisted by used tool. More tools that focus on visual SPARQL query construction (e.g. Gruff [1], iSPARQL[1], Nitelight [6] or SEWASIE [2]) currently exists. While first two focus at direct SPARQL construction that is based on conjunction of conditional triples, later two are based on different diagrammatic approach. Nitelight and iSPARQL support more advanced SPARQL constructs and filtering upon various graphs, with possibility to specify optional conditions. While Nitelight, iSPARQL and Gruff are closely bound to the structure of SPARQL, SEWASIE puts emphasis on different query refinement strategy. It is based on ontology that describes concepts of given domain. According to this hierarchy user substitutes some more general constrains with more specific ones until he reaches satisfying result set.

3 Visualization and navigation in RDF graph

Information retrieval with Tripleskop tool can be divided into two subsequent phases:

1. phase of initial data exploration, examination of used classes hierarchy, namespaces and common relations, understanding of some smaller parts of stored RDF graph
2. phase of more precise visual query construction which benefits from knowledge obtained in first phase

In this section we describe methods of raw RDF visualisation, which is first step in the information retrieval process in presented tool.

We agree with multi paradigm visualization approach presented in [1][1]. Our tool currently offers three different integrated views, each suitable for different purpose. The most important component is the force directed graph layout. We have chosen traditional swimming layout without central rooted resource. We came to conclusion, that this layout best suits all needs of dense networks visualization with cycles, which are very common in RDF graphs. However, following difficulties have to be considered, which make raw visualisation unclear and confusing:

- presence of nodes with large degree (branching factor possible higher than few hundreds)

[1] http://dbpedia.org/isparql/

- common cycles and properties pointing to the same resources, or literals
- lack of schema, presence of long URIs, absence of catalog of used namespaces

Two different layout approaches has been tested. First solution was based on arbor.js canvas implementation based on repulsive forces, second one (d3.js force layout) utilized geometrical constraints approach. While first one required less hardware resources, it experienced common problem when too many nodes were present. At some point, repulsive forces might get into state, when they stop converging into final state. Second approach is based rather on geometrical constraints than repulsive forces approach, thus allowing faster motion convergence to fixed state[2]. Disadvantage of this approach resides in fixed links length, what usually allows fewer resources to be shown at a time when compared to first repulsive approach. Solution to this problem resides in custom implementation of nodes repulsion based on dynamically computed links length. Main idea of this principle is based on proportional relationship between mutual node repulsion and their degree. Value of this repulsion is expressed in formula 1, where N is set of all nodes, $\{A, B\} \subseteq N$, $deg(X)$ is degree of node X. Constants k, q have been empirically obtained. Value of $rep(A, B)$ is equivalent to final length of link between interconnected nodes A and B.

$$rep(A,B) = \begin{cases} k, & min(deg(A), deg(B)) = 1 \\ k + (deg(A) + deg(B))\frac{|N|}{q}, & min(deg(A), deg(B)) \neq 1 \end{cases} \tag{1}$$

To promote visualization clarity, prevent information overload and increase amount of displayed data, several techniques has been adapted, and improved:

- Clustering: Resources or literals bound with the same property to the source or destination node are displayed as clustered nodes. This allows us to display instances of classes, or multiple labels of given resource in more user friendly way
- Semantic zoom and filtering: When user wishes to view data from higher perspective with less details displayed, he can either set up filter on various resources and properties types, which are not points of his interest, or allow collapsing of some common properties directly into node which are they associated with (e.g. literal values such as rdfs:label, or resource classification such as rdf:type and rdfs:subClassOf relations).

Introduced visualization approach allows expanding (or collapsing) of node neighbourhood, thus changing the focal point of introduced view. As we declared earlier, we adapted the idea of multi paradigm visualization approach that consists of several integrated views:

- Introduced force directed layout with zoom and clustering features
- Tree visualization of hierarchical relations (e.g. class hierarchy from contained ontologies)
- Traditional tabular key-value resource explorer
- Simple full text search upon common rdfs:label properties allows rough, but fast resource lookup

Tree visualization of whole class hierarchy or full text search can be the entry point to main force directed layout visualization. If user prefers tabular view instead of graph visualization, he can at any point display node context in tabular input, or vice versa: pin selected node from tabular output or class hierarchy visualization directly to main graph canvas and explore its relations with resources that are already present in visualization. Big advantage is the nature of RDF data, which are networks of small world. That thanks to nodes with higher degree we can reach almost any 2 resources by small number of hops. If user can fast find single resource similar, or semantically closer to searched resources, he would be able to navigate to search target quite easily.

[2] https://github.com/mbostock/d3/wiki/Force-Layout

4 Visual query construction

After user gains necessary knowledge about data character, explore namespaces and common resources and properties URIs, he can proceed to visual query construction. Visual query construction uses the exactly same view as force directed visualization layout. Result of visual query construction is valid SPARQL query, which is de facto graph pattern based on the structure of stored data. Consequence of this fact is that any visualization can be quite easily transformed into search query by replacing some of its parts by variable substitutions and removal of too specific parts (e.g. literals). For example, if user wants to explore all relations between two selected resources, he can easily convert interconnected edge into variable. If user wishes to formulate more specific query, he can build completely new query from scratch.

Process of query construction is based on drawing of graphical representation of SPARQL query. During this process user puts resources with known URIs to visualization. To retrieve all relationships between given resources, he can draw variable edge between those nodes and mark this yet unknown edge to be presented in query output. If user also wants to find out values associated with some known property, he can draw this property relationship from already known resource node to yet unknown variable node. Constructed query is small graph with some known parts (resources, filters upon literal values), which form query constraint and some variable nodes and edges. Those variables, which are subject of given search session, can be marked to be present in query output. Other ones will be used as substitutions for parts of graph that can take any available value from underlying RDF dataset. To support user during query construction process, available URIs of predicates and URIs of all classes extracted from obtained ontologies are autosuggested in used input fields. According to frequent presence of geo-spatial entities, geo-filtering based on interactive map has been implemented to promote user experience.

5 Evaluation

To evaluate efficiency of presented visualization and information retrieval methods, preliminary experiment with 9 respondents has been realized. All respondents were technically educated students. None of them had knowledge of technologies of web 3.0, nor was an expert in selected domain, where experimental search took its place). Respondents were asked to accomplish two different search scenarios in different domains of real estates and medical products, to verify tool domain independency. The main goal of first search scenario was to get familiar with graph data representation and all necessary features of evaluated tool. This search was of navigational character. It took place in mentioned healthcare domain. Result should be the full understanding of vitamins ontology and finding of some examples of concrete medical products available on the market according to prescribed taxonomical classification. Second search scenario was mainly focused on visual query construction in a domain of real estates. Goal of this search was to find all suitable real estates in described area according to given hypothetical preferences and limitations.

During search we tracked time spent in various visualization components and all user actions. This preliminary experiment has shown the amount of average time needed for understanding of RDF data representation and proper usage of presented tool. During experiment we have noticed, that almost all respondents in the beginning tended to use full text search and key-value property explorer. After closer explanation of graphical visualization, they almost completely switched to graph visualization layout. Possibility of visual query construction was used rarely, mainly for more precise filtering purposes.

Measured durations of experimental searches can be found in the Table 1.

Table 1. Summarized results of preliminary experiment.

	Navigational search	Informational search	Basic principles understanding
Average time	12.5 minutes	7.5 minutes	4.5 minutes
Deviation	4.5 minutes	3 minutes	4 minutes

6 Conclusion and future work

In this paper, we have shown that properly visualized RDF graph can be understood even by users without previous knowledge in selected domain and technical education in a field of semantic databases. All improvements of RDF graph visualization effectively maintain whole view understandable, even when higher number of resources is present at the time. Perspective view, clustering and integration with traditional views prevent information overload during navigation in large RDF graphs and avoid context loss at the same time. As a contribution of this tool we also consider web-based platform and architecture that allows large graphs exploration.

Visual way of SPARQL query construction has potential, to be adapted even by users without previous detailed technical knowledge in quite acceptable time. Main problem during presented approach still resides in 100% accuracy needed during resources and relationships naming. Our future work will focus on shortening of time needed to construct proper visual queries. Prepared approach will be independent of any ontological data; all predicates and types will be extracted dynamically from whole RDF repository. Search time could be also reduced by dead-end queries prevention and semi-automated visual query correction.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Aasman, J., Cheetham, K.: RDF Browser for Data Discovery and Visual Query Building. In: *Workshop on Visual Interfaces to the Social and Semantic Web*, Palo Alto, USA, 2011.

[2] Catarci, T., et al.: An ontology based visual tool for query formulation support. In: *On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops*. Sicily, Italy, (2003), pp. 32–33.

[3] Dokulil, J, Katreniakova, J.: Visualization of Large Schemaless RDF Data. In: *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Papeete, (2007), pp. 243–248.

[4] Dokulil, J., Katreniaková, J.: RDF Visualization – Thinking Big. In: *20th International Workshop on Database and Expert Systems aplication*, Prague, (2009), pp. 459 – 463.

[5] García, R., et. al.: Publishing and Interacting with Linked Data. In: *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, New York, (2011).

[6] Russell A, et. al.: Nitelight: A Graphical Tool for Semantic Query Construction. In: *Semantic Web User Interaction Workshop (SWUI 2008) at CHI*, Florence, (2008).

[7] Rutledge, L., Ossenbruggen, J., Hardman, L.: Making RDF Presentable In: *WWW '05 Proceedings of the 14th international conference on World Wide Web*. New York, (2005), pp. 199–206.

[8] Tvarožek, M., Bieliková, M.: Factic: Personalized Exploratory Search in the Semantic Web, In: *10th international conference on Web engineering*, Berlin, (2010), pp. 527–530.

[9] Zhang, K., Wang, H., Tran, D.T., Yong, Y.: ZoomRDF: Semantic Fisheye Zooming on RDF Data. In: *Proceedings of the 19th international conference on World wide web*, North Carolina, USA, (2010), pp. 1329–1332.

Using Complex Event Processing to Detect Plagiarism

Peter ŠINSKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xsinskyp@fiit.stuba.sk`

Abstract. The attention is pointed into plagiarism revealing domain for a long time. Plagiators are more and more sophisticated in plagiarism creation and there is need to focus on growing available document corpus. This is why document processing speed is important. Among important and powerful software system we include those based on complex event processing systems (CEP). Available CEP systems have performance of event processing up to hundreds of thousand events in second. There is a potential to create powerful system for plagiarism revealing based on this principle. Assumption for this system is located in usage of the n-gram method, where we see a connection between n-grams and complex events. We proved that in CEP domain an identical gram occurrence can represents event and occurrence of more identical grams or n-grams represents complex event. Performed quazi-experiment proves correctness of our presumptions and brings a bit more of discoveries.

1 Introduction

Plagiarism revealing is important discipline, which is given attention for a long time [8]. Especially today when broadband internet connection is available, plagiarists have easy access to wide range of documents online. Plagiarists are more and more sophisticated in plagiarism creation. Besides, available document corpus is constantly growing. For example, schools and science organizations are constantly publishing new papers to libraries like ACM or IEEE. Every year there are added ten thousands articles, so there is need to focus on document processing performance too.

There are several systems for plagiarism revealing [7], which uses different methods for plagiarism revealing [6]. These systems does not consider complex event processing (CEP) concept. We are not claiming that these systems are not sufficient or satisfactory enough. CEP is one of three general styles of event processing [9]. It allows monitoring several multiple event flows, analyze them in terms of performance indicators and react in real time based on detected occasions. Available CEP systems claim their performance up to hundreds of thousands events in seconds. This is why there is a potential to create powerful plagiarism revealing system based on

[*] Master degree study programme in field: Information Systems
Supervisor: Dr. Ján Lang, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 74–79.

CEP. Base idea is usage of the n-gram method, where we see connections between n-grams and complex events. We confirmed that compliant gram occurrence means event and more of gram or n-gram occurrences means complex event. Realized quasi-experiment proves meaning of our assumptions and brings more discoveries.

2 Related works and existing solutions

After reviewing available literature, we found only one thesis which directly connects plagiarism detection and CEP system. In article [2] authors introduce system for text copy detection (TCD system) based on complex event handling architecture. This system is organized in packages. Every package will provide several text processing phases like cleaning text, feature choosing and plagiat detection. Whole system is based on OSGi platform. Advantage of this system is, that every package contains different algorithm for text processing. Thanks to this they will be able to recognize texts in different ways, based on starting different packages. For text comparison authors chose TF-IDF method and method based on hash function. In this point our theses differ, since we decided to use n-gram method. TCD system was not implemented yet, so it stays only in conceptual form.

Copypaste [4] application was created by students for team project at our faculty of informatics and information technologies. Goal was creating system for plagiat detection, which except slovak text documents is able to find match in source files of programs. The resulting application covers all aspects of system for palgiat detection, therefore it solves text pre-processing goals. Application uses several text comparing methods including n-grams. Advantage of this method is parallelism usage. Application does not consider CEP concept.

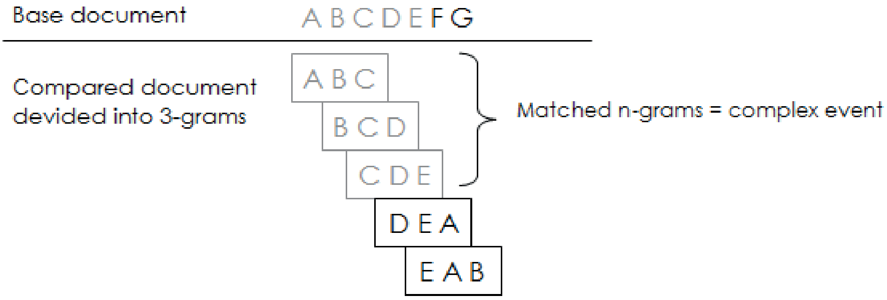
Second solution created at our faculty is application PlaDes [5]. Application focuses on english and slovak text document comparison Application solves text pre-processing, text comparison and result visualization. Advantage of this application is supported parallel processing. For text comparison is used n-gram method specifically 3-gram. In future we are going to compare our processing results with this application.

3 Connection between n-grams and events

Complex event processing is one of three general styles for event processing. It allows monitoring multiple event flows, to analyze them in the meaning of performance indicators based on pre-defined rules and reacts in real time based on detected threats. In this paper we introduce system model for plagiarism revealing based on complex event processing. Available CEP systems have performance of event processing up to hundreds of thousand events in second. Event is base unit of the event processing system. Essentially there are two distinct meanings of event [1]. The first meaning of event is an activity that happens for example landing aircraft or sensor output. The second meaning of event is event object that represents that activity in a computer system for example RFID sensor reading message or e-mail confirmation of the plane ticket reservation. These events can be structured into single event that we calls complex event.

One of many available methods for text document comparing is the n-gram method [10]. N-gram represents a set of n characters or words. If we have a sentence and we want to split it to n-grams, we do the following technique: Take the first n words at the beginning and go on up to the end of sentence word by word. Now we have all n-grams. Now we can search for these n-grams in suspicious documents. This method is very effective as the authors claim in their paper [4]. However, the most effective is usage of 2 or 3-grams [3].

Based on mentioned facts we can point a connection between events and n-grams. We can assume n-gram as a complex event. This assumption is based on idea, that event will represents match occurrence of one gram and complex event will represents occurrence of n-gram. This is illustrated on the picture (see Figure 1).


Base document
A B C D E F G
Compared document devided into 3-grams
A B C
B C D
C D E
D E A
E A B
Matched n-grams = complex event

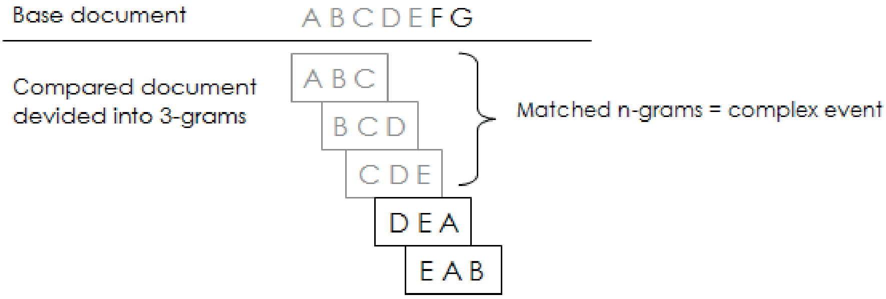

Base document
A B C D E F G
Compared document devided into 3-grams
A B C
B C D
C D E
D E A
E A B
Matched n-grams = complex event

Figure 1. Comparison text using n-grams.

4 Base model of system for plagiarism detection based on CEP

At the beginning we have decided create a basic system model for plagiarism detection based on CEP, where we can test basic assumption mentioned in the previous part. At first was necessary choosing the appropriate CEP system. For this purpose we have chosen CEP system Esper[1] due to many reasons:

- allows many options event representation,
- provides EPL language for expressing rich event patterns,
- exceeds over 500 000 event/s on a dual CPU 2GHz Intel based hardware on a VWAP benchmark with 1000 EPL demands registered in the system.

System model idea lies in searching 3-grams (letters) in a event stream, which represents sequence of letters. Esper allows data searching in a event stream based on EPL demand. The final draft of model is very simply. As it is presented on the picture (see Figure 2) model consists of two components: event generator and event processor.

Event generator is a component, which generates random letters in range of “A” to “BZ”. Event is expressed by means of class Gram, which represents given generated letter. Events generated like that are sending to input processor of events which represents CEP system. CEP system receives stream of events to data window, where on the basis of EPL demand is searching 3-grams. We have used one of three possibilities EPL demand registrations by means of clause match recognize in this experiment. Example of this EPL demand looks as follows:

```
"SELECT a_docName from Gram "
       "match_recognize ( " +
```

[1] http://esper.codehaus.org

```
"measures A.Word as a_word,
           B.Word as b_word,
           C.Word as c_word,
           A.DocName as a_docName  " +
"pattern (A B C) " +
"define " +
"A as A.Word = 'AA', " +
"B as B.Word = 'F', " +
"C as C.Word = 'R')";
```

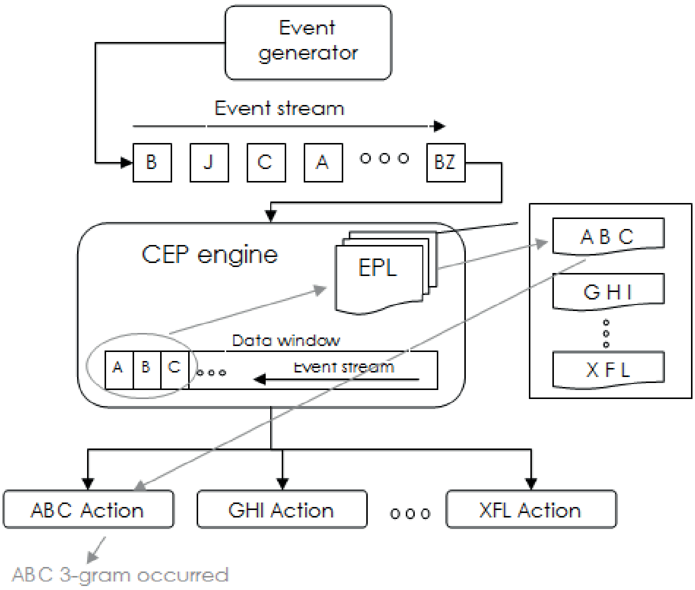


Figure 2. Base model of system for plagiarism detection based on CEP.

EPL demand is word by word: “Search 3 consecutive events, which attribute *Word* represents values ‘AA’, ‘F’, ‘R’”. CEP machine monitors events in this way which is coming to date window and react in case of conformance. This model monitors one stream of events only, because we have experimented only with one event generator.

5 Discussion of results

After implement was designed model capable search 3-grams in a stream of events through predetermined EPL demands. The Model has been tested, that we have gradually increased the number of EPL demands and subsequently has been stream of events turned on to CEP machine, where we have tried searching 3-grams. Generator of events has generated 100-thousand streams of events during each testing. During testing we have focused on processing speed and if EPL demand searches in stream of events 3-grams.

On one hand experiment shows, that creating EPL demands for searching is time consuming. It means that the more 3-grams have been created for searching, the more time has been needed to create EPL demands. We are considering results as a little bit frustrated in fact that the number of text document and 3-grams are creating by dividing text. We still have two ways of representing events provided by Esper, which can be faster in the registration of events and the comparison. On other hand, we assume that if we are able search 3-grams in a stream of random generated events, we will be able to search 3-grams in a stream of events generated from compared text document based on EPL demands generated from base text document. See Figure 3 for experiment results.

6 Conclusions and feature work

In this thesis we have reported assumption, which introduces relationship between complex event processing and n-gram methods, which is used for plagiarism detection. We have confirmed our assumptions and reached the resolution, that it is possible to use complex event processing mechanisms in plagiarism detection system, based on executed experiment. Achieved results compel us to consider other processing methods.

In our experiment we were searching 3-grams using preset EPL queries, where every query expressed a particular 3-gram. Essentially it is dynamically generated EPL query. Here we like to use more EPL query record methods supported by CEP system Esper. Except the dynamically generated EPL queries search we like to try comparison using one EPL query, which will compare two events representing 3-grams.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0233-10.

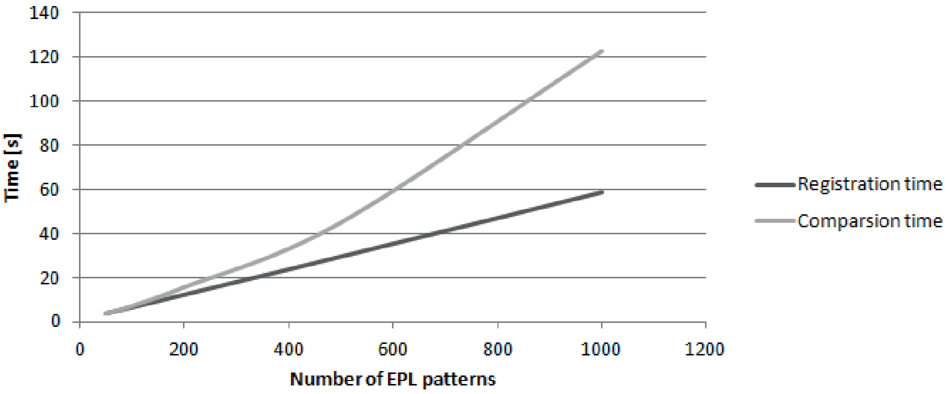


Figure 3. Time difficulty of registration EPL patterns and event comparison.

References

[1] Luckham, D., Schulte, R., Adkins, J., Bizzarro, P., Jacobsen, H., Mavashen, A., … Tucker, D. (2011). Event processing glossary-version 2.0. *Event Processing Technical Society*.

[2] Bao, J., Qi, Y., Hou, D., & He, H. (2011). A text copy detection system based on complex event processing architecture. *... a Service-Based Internet. ServiceWave 2010 ...*, 203–207.

[3] Barrón-Cedeño, A., & Rosso, P. (2009). On automatic plagiarism detection based on n-grams comparison. Advances in Information Retrieval, 696–700.

[4] Freml, M., Chalupa, D., & Mego, M. (2009). *Podpora kontroly plagiarizmu*. *Tímový projekt*. STU. Retrieved from http://labss2.fiit.stuba.sk/TeamProject/2009/team09is-si/doc/TP1ria_final.pdf

[5] Chudá, D., Návrat, P.: Support for checking plagiarism in e-learning. Procedia - Social and Behavioral Sciences, Innovation and Creativity in Education, 2010, vol. 2, no. 2, pp. 3140-3144, ISSN 1877-0428.

[6] Alzahrani, S. (2012). Understanding plagiarism linguistic patterns, textual features, and detection methods. *Systems, Man, and …*, *42*(2), 133–149.

[7] Bin-Habtoor, a. S., & Zaher, M. a. (2012). A Survey on Plagiarism Detection Systems. *International Journal of Computer Theory and Engineering*, *4*(2), 185–188. doi:10.7763/IJCTE.2012.V4.447

[8] Maurer, H., Kappe, F., & Zaka, B. (2006). Plagiarism-A Survey. *J. UCS*, *12*(8), 1050–1084.

[9] Luckham, D. (2008). A Brief Overview of the Concepts of CEP1. Available at: http://complexevents.com/wp-content/uploads/2008/07/overview-of-concepts-of-cep.pdf

[10] Kučečka, T. (2011). Plagiarism Detection in Obfuscated Documents Using an N-gram Technique, *3*(2), 67–71.

Evaluating Context-aware Recommendation Systems

Juraj VIŠŇOVSKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
visnovsky.j@gmail.com

Abstract. We propose a novel approach for evaluating context-aware recommendation systems using supposed situations. In this paper, we focus on improving a number of recommendations covered by user study evaluation approach and thus reduce its costs. In this concern, supposed situations are used to simulate various context-aware situations and to simulate the behaviour of a bigger set of users using only a standard number of user study participants. In the experiments we use a dataset from a simple event recommendation system. To evaluate the proposed approach we compare outputs of user study using supposed situations with standard user study.

1 Introduction

In the age of information overload we witness an increase the popularity of personalized systems. Among many solutions coping with content adaptations to users' needs, recommendation systems seem to stand above all. The main purpose of recommendation system is to deliver relevant information to the user and thus simplify his navigation in data overflow.

User's actions and decision, while using software system, are influenced by many factors. These factors, also known as contexts, describe the situation and the environment of the user. Personalized systems may collect a wide range of different types of context. Such may be a user's current mood, health condition, activity, location, etc. Including contexts in the user model, which represents user's preferences and interests in a given domain, make it possible to generate recommendations for specific situations. This may lead to recommendation quality enhancement.

In general, when evaluating a context-aware recommendation system, we use methods dedicated to evaluate common recommendation systems. Although there might be some differences as user's preference data are collected under various contexts. When evaluating a context-aware recommendation system, it is often difficult to set up environment to match examined contexts.

When evaluating or choosing one of several recommendation algorithms, we may employ one of three possible approaches: offline evaluation, online evaluation and user study [9]. Offline evaluation is easy and cheap to conduct, but its outputs are less reliable than outputs from online

[*] Master degree study programme in field: Software Engineering
Supervisor: Dušan Zeleník, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 80–85.

evaluation and user study [10]. Although, both user study and online evaluation are more accurate, they have one major drawback, which is their costly conduct [10]. In this paper, we focus on improving user study evaluation approach for context-aware recommendation systems.

2 Related work

Evaluation of recommendation systems is well covered by Shani and Gunawardana [9]. They reviewed three types of experiments (offline and online experiments, and user studies). The authors described various properties of recommendation systems that may be measured in the process of evaluation and they emphasize the importance of forming a hypothesis, controlling variables and the power of generalization when drawing conclusions. Sean in his paper [8] explained why focusing on improving recommendation system's accuracy in some cases might be detrimental.

Including context in recommendation systems may lead to improvement of recommendation accuracy [6] and recall [1]. On top of that, Zeleník in his work [11] proposed a method to improve accuracy of context-aware recommendation systems using context inference as he solved the problem of context sparsity. Asoh et al. had a similar idea when solving problem of lack of training data in context-aware system [5]. They used so-called supposed situation to acquire sufficient amount of training data and then to construct statistical preference models.

3 User study evaluation

User study seems to be superior evaluation approach as it provides several relevant assets in comparison to other approaches. User study requires a set of test subjects, performing a number of various tasks. Given tasks require an interaction of participants with examined recommendation system. While users perform these tasks their actions are being observed and recorded. User study allows us to not only measure standard recommendation aspects, but we may also observe many other aspects of examined system (such as difficulty of performing various tasks). Collecting qualitative data is crucial for interpreting the quantitative data [9]. As we have mentioned, the main disadvantage of user study is its costly conduct. Collecting a satisfying set of participants, performing several tasks is difficult and expensive as well. Therefore, we propose a method for more efficient user study execution by exploiting the potential of its participants.

4 User study evaluation using supposed situations

To reduce the drawbacks of the user study evaluation approach (setting up environment according to contexts and gathering satisfying set of users) we propose using supposed situations. Our goal is to maximize the amount of evaluated recommendations.

Proposed evaluation approach (**Error! Reference source not found.**) requires input data of context-aware recommendations and user study participants. Firstly, we analyze recommendations data and identify its frequent patterns. At the same time user study participants are being modelled. Participant models reflect test subjects' ability to evaluate recommendations using supposed situations. Then recommendations are assigned to user study participants to be evaluated. The output of this approach is a set of evaluated recommendations.

Supposed situation is a situation where the user study participant pretends to be in a specific situation given by inquiry. This is a cheaper alternative to setting up a real situation matching given contexts and putting participant into it.

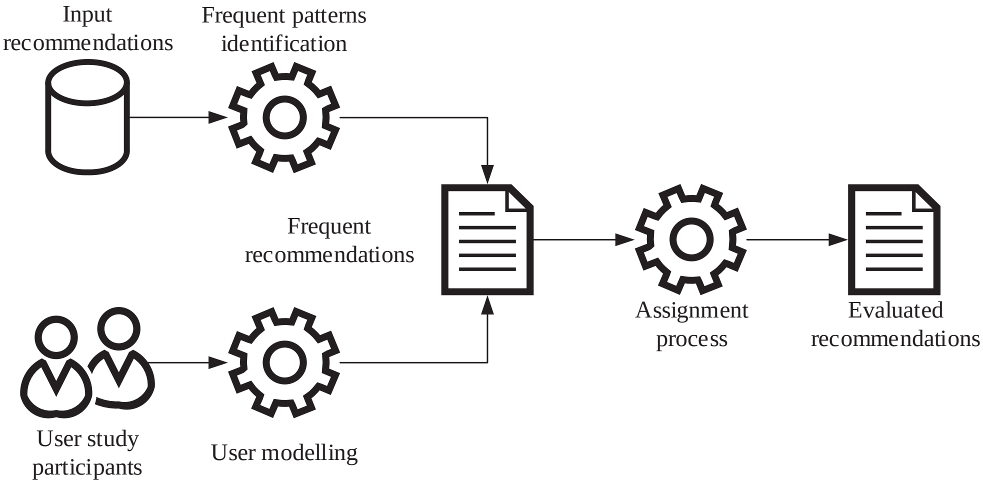


Figure 1. Process describing application of user study experiment using supposed situations.

4.1 User study participants modelling

Participants modelling consists of determining their ability to evaluate supposed situations. In our work, we adopt explicit user modelling approach [2], as we allow users to provide us all relevant information about their ability to evaluate supposed situations via inquiries. Acquired user model represents participant's credibility (weight) when evaluating given supposed situation. We identified and analyzed two approaches to measure participants' reliability.

A general approach is based on measuring social perspective taking skills, which reflects the participant's tendency to adopt psychological perspectives of another person. To measure this skill we use Interpersonal reactivity index inquiry [3] and inquiry based on Gehlbach's approach [4].

Another, more specific approach includes two possible applications. The first application is based on experience of participant. Basically, in this case participant is asked whether he has been in conditions described by presented supposed situation. This approach is based on the assumption that if participant found himself in given conditions in the past, he is able to evaluate the supposed situation. The second application of the specific approach consists of mutual measuring of participants' reliabilities. In this case, we let a subset of participants to evaluate recommendations originally dedicated to them, so that we get a small set of evaluated recommendations. Then we let other participants to evaluate supposed situations based on this small set of recommendations and determine their reliability.

4.2 Recommendations to user study participants assignment

Once we have determined the reliability of user study participants' evaluation, we may proceed to a process of assigning recommendations to participants to be evaluated. Every participant of user study has a limit of evaluated recommendations as it is not our intent to overload any of them.

The profit of participant *p* evaluating recommendation *r* for user *u* is given by:

$$profit(u,r) = \begin{cases} f(r), & u = p \\ reliability(p) \cdot f(r), & u \neq p \end{cases} \tag{1}$$

where *f* is a function returning frequency of recommendation passed as an argument and reliability is the ability of participant to evaluate supposed situations measured in the participant modelling process.

After estimating reliability of evaluating input recommendations by all user study participants we get a P×R matrix, where P is participants' count and R is a total number of input recommendations. Our goal is to solve problem of assigning a set of *n* profits to *m* participants, such that the selected profit sum is maximized and capacity of any participant is not exceeded. This problem is called a multiple knapsack problem, which is formally defined as [7]:

$$\text{maximize} \sum_{i=1}^{m}\sum_{j=1}^{n} p_j x_{ij} \ , \tag{2}$$

$$\text{subject to} \sum_{j=1}^{n} x_{ij} \le c_i, i = 1,\ldots, m, \tag{3}$$

$$\sum_{i=1}^{m} x_{ij} \le 1, x_{ij} \in \{0,1\}, i = 1,\ldots, m, \ j = 1,\ldots, n, \tag{4}$$

where *p* is a profit of participant *i* for evaluating given recommendation, x_{ij}=1 if recommendation *j* is assigned to participant *i*, and is zero otherwise.

Over the years, many approximation and exact algorithms were proposed to solve this optimization problem. In our work we use the Mulknap algorithm which provides the best solution times in comparison with other algorithms, even with high *n*:*m* ratio [7]. Mulknap is a branch and bound algorithm, which consists of an enumeration of all possible solutions and recursive reducing solution space using upper and lower bounds.

5 Evaluation

We conducted a pilot experiment to evaluate reliability of user study participants. In this experiment we let six participants fill out a form defining their short- and long-term contexts. Then we addressed them Interpersonal reactivity index inquiry. Finally, we let the participants to rate twenty recommended items. The rating of an item is an integer from -100 to +100.

In the following step, we let a subset of the original set of participants to evaluate a random set of supposed situations, based on explicit feedback from participants acquired in the previous phase. Experiment participants evaluated a total of 87 supposed situations. The accuracy of evaluated supposed situation is defined as:

$$accuracy = \frac{|rating_e - rating_{ss}|}{rating range} \ , \tag{5}$$

where *rating*$_e$ is the original rating of evaluated recommendation, *rating*$_{ss}$ is the rating provided by participant evaluating current supposed situation and *rating range* is the number of possible ratings, in our case it is 200.

We observed that good results obtained with Interpersonal reactivity index inquiry did not affect the ability of participants to correctly evaluate supposed situations (Figure 2), as the value of Spearman's correlation coefficient between these two variables is -0,39.

Then we examined effects of the rate of experience with presented contexts on evaluation accuracy (Figure 2). We observed that there is no correlation between these two variables, as the value of Spearman's correlation coefficient is -0,08.

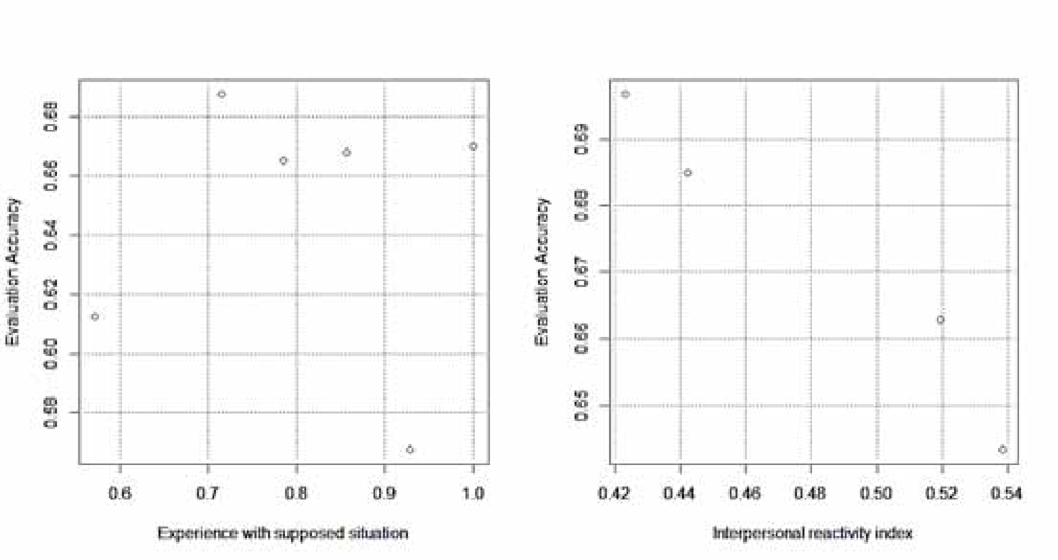


Figure 2. Correlation between supposed situation evaluation accuracy and contexts experience (left Figure). Correlation between supposed situation evaluation and interpersonal reactivity index (right Figure).

One of the main cause of the unsatisfactory results measured in the pilot experiment is that we have allowed users to use too wide range of possible ratings. As we can see in Figure 2 many participants were unable to reproduce their own rating of recommended item. This problem could be solved by shortening the rating range.

In our next experiment we will introduce two more approaches as described in 4.1. We believe both of these new approaches will be superior to approaches examined in the pilot experiment. Gehlbach's inquiry outputs should be more reliable than outputs of Interpersonal reactivity index inquiry as it is harder for the participants to sanitize its results. We also assume that higher number of participants will be beneficial as it will bring diversity.

Then, we will carry second experiment to evaluate proposed user study evaluation approach. As our goal is to improve user study coverage, we will examine the attributes of common user study approach with proposed user study experiment using supposed situations. Our attention will be paid mostly on the number of recommendations covered by each approach, but we will measure accuracy and F-measure as well. We expect our approach's accuracy to be slightly inferior to common user study experiment, but we believe in increase of F-measure. To assure credibility of the experiment's results it is necessary to collect at least 20 participants to take a part in the main experiment.

6 Conclusions and future work

In this paper, we described a novel approach for evaluating context-aware recommendation systems using supposed situations. We focused on improving user study evaluation approach. The main contribution of our work is in minimizing user study experiment conducting costs. We believe our proposal helps to conduct user study experiments more efficiently by exploiting the potential of its participants. In the proposed approach, participants may be used not only to evaluate recommendations dedicated to themselves, but they may evaluate recommendations assigned to other users as well.

In the future work we plan to examine all described approaches of the user study participants' reliability to evaluate supposed situations. We intend to analyze each approach and identify their feasibility in our case. Then we intend to conduct the main experiment to compare our novel approach with common user studz evaluation.

All mentioned experiments will be conducted on one context-aware recommendations dataset we possess with. Consequently we are not able to prove proposed approach domain independence. In our work, we aim to demonstrate novel possibilities of conducting experiments and reducing their costs by using the proposed approach. In our future work we intend to examine domain dependency of the novel evaluation approach.

Acknowledgement: This work was partially supported by the Science and Technology Assistance Agency under the contract No. VG1/0675/11.

References

[1] Adomavicius, G., Sankaranarayanan, R.: Incorporating contextual information in recommender systems using a multidimensional approach. In: *ACM Transactions on Information Systems*, ACM New York, USA, 2005, vol. 23, no. 1, pp. 103–145.

[2] Barla, M.: Towards social-based user modelling and personalization. In: *Information Sciences and Technologies Bulletin of the ACM Slovakia*, 2011, vol. 3, no. 1, pp. 52-60.

[3] Davis, M.: A multidimensional approach to individual differences in empathy. In: *JSAS Catalog of Selected Documents in Psychology*, American Psychological Association, 1980, vol. 10, no. 4, pp .85.

[4] Gehlbach, H.: A New Perspective on Perspective Taking: A Multidimensional Approach to Conceptualizing an Aptitude. In: *Educational Psychology Review*, Kluwer Academic Publishers, 2004, vol. 16, no. 3, pp. 207–234.

[5] Ono, C. et al.: Context-aware preference model based on a study of difference between real and supposed situation data. In: *UMAP '09 Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, Springer-Verlag Berlin, Heidelberg, 2009, pp. 102–113.

[6] Palmisano, C., Tuzhilin, A., Gorgoglione, M.: Using Context to Improve Predictive Modeling of Customers in Personalization Applications. In: *IEEE Transactions on Knowledge and Data Engineering*, IEEE Educational Activities Department, USA, Piscataway, 2008, vol. 20, no. 11, pp. 1535–1549.

[7] Pisinger, D.: *Algorithms for knapsack problems*. Dissertation thesis. 1995, 200 pp., Computer science department at the Faculty of science, University of Copenhagen, Denmark.

[8] Sean, J., Mcnee, M., Konstan, J.: Accurate is not always good: How accuracy metrics have hurt recommender systems. In: *Extended abstracts of the ACM conference on Human Factors in Computing Systems*, 2006.

[9] Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: *Recommender systems handbook*, 2011, pp. 257–297.

[10] Shani, G., Gunawardana, A.: A survey of accuracy evaluation metrics of recommendation tasks. In: *The Journal of Machine Learning Research*, 2009, vol. 10, pp. 2935–2962.

[11] Zeleník, D., Bieliková, M.: Context Inference Using Correlation in Human Behaviour. In: *Semantic and Social Media Adaptation and Personalization (SMAP)*, 2012, pp. 3–8.

Identifying Author Profiles Using Citation Based Techniques

Zoltán HARSÁNYI*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`harsanyi@fiit.stuba.sk`

Abstract. Building expert profiles and ranking authors in a scientific community is a common task in digital libraries. By ranking authors and building their profiles we can also classify and predict conference quality, especially when the traditional conference ranking methods and measures can not be applied effectively. We propose a citation based method to build expert profiles of authors based on their scientific publications using multi-faceted approach. The approach ranks authors using their publications, number of citations received, extent and proportion of citations within a particular area.

1 Introduction

Searching the human expertise has recently attracted much attention in Information Retrieval (IR) community. Considering experts as objects, expert finding is one of the challenging types of object level search, which concerns itself with ranking people who are knowledgeable in a given topic. Different approaches for expert finding have been proposed to identify experts in simple environments such as organizations and universities. While these approaches are quite effective in these simple domains, they are not appropriate for complicated environments such as bibliographic networks [3].

Research on quantitatively evaluating researchers contributions is an important task because of its practical importance for making decisions in science, such as matters of individual promotion and allocation of grants. Moreover, in academic domains, identifying experts for a research field has also a several potential practical applications: finding program chair members for a conference, selecting a panel of researchers for a track, determining important experts for consultation by researchers embarking on a new research field, assigning papers to reviewers automatically in a Peer-Review Process, from the point of view of students applying to graduate schools finding researchers as supervisors. Automatized expert finding can also be applied to rank emerging conferences by ranking their PC members or authors. In this work we are trying to solve the "first step" of our approach in achieving automatized conference ranking.

[*] Doctoral study programme in field: Program Systems
Supervisor: Assoc. Professor Viera Rozinajová, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 86–92.

Expert finding has received increased interest in recent years since the advent of the expert search task in the TREC Enterprise track[1]. The task of expert finding is to come up with a ranked list of experts with relevant expertise in a given topic. The current developments in expert search are concentrated in the Enterprise corpora. They have provided a common platform for researchers to empirically assess methods and techniques devised for expert finding, as one of the most valuable knowledge in an enterprise resides in the minds of its employees. Enterprises must combine digital information with the knowledge and experience of employees. Expert finding in the enterprise corpora addresses the task of finding the right person with the appropriate skills and knowledge: "Who are the experts on topic X?" Given a query (describing the area in which expertise is being sought), participating systems have to return a ranked list of person names in response [1, 2].

Expert finding and expertise modeling are related in that expert finding techniques can make use of the profiles obtained from expertise modeling. However, several expert finding algorithms avoid explicit modeling of experts (known as candidate-centric approaches) and instead adopt a document-centric approach where ranking is based on a subset of documents obtained using the query. For both expert ranking and expertise modeling, the evidence of expertise depends on the context [5].

The quality and quantity of publications has a significant contribution on assessing researcher performance. As a result, bibliometric indicators such as citation counts, publication numbers and different types and versions of impact factor are widely used. These methods are based on a natural rule that high quality papers are more likely to attract recognition and be cited frequently. In fact, fetching accurate and complete citation networks is a challenging task. For example, survey and review papers are generally more frequently cited than other papers, but this may not be a measure of the quality of research [9]. To overcome the issues mentioned above, we use an automated citation mining technique which incorporates multiple facets in providing a more representative assessment of expertise. We see these facets as providing multiple sources of evidence for a more reflective perspective of experts. The system mines multiple facets for an electronic journal and then calculates expertise weights. The measures provided are however limited by the coverage of the database of publications and expert profiles used.

2 Related work

The participants of the Enterprise track of TREC studied expert finding in context of enterprise data on the W3C collection. Balog, et al. proposed probabilistic models for expertise profiling and expert finding in context of sparse data environments such as webpages pertaining to research institutes and universities where the documents are more structured and relatively noise-free.

Balog et al. also formalized and presented three models for expert finding based on the large-scale DBLP bibliography and Google Scholar for data supplementation. The first, a novel weighted language model, models an expert candidate based on the relevance and importance of associated documents by introducing a document prior probability, and achieves much better results than the basic language model. The second, a topic-based model, represents each candidate as a weighted sum of multiple topics, whilst the third, a hybrid model, combines the language model and the topic-based model [1].

Researches dealing with expert profile creation in enterprise environment are mostly based on analyzing textual documents of authors and using different models (mostly the document centric method, proposed by Balog et al., presented in [3]) for ranking these documents [7, 8, 10]. There are two principal approaches to expert modeling using document centric method: query-dependent and query-independent. In both cases the expert-finding system has to discover documents related to a person and estimate the probability of that person being an expert from the text. Zarandi et al. however presented a technique for generating evolving expert profiles of individuals composed of

[1] `http://trec.nist.gov/`

their skills and competencies using heterogeneous data from divergent sources of information. They used self-declarations, completed learning activities, and previous work experience to generate the initial profile [11]. Dozier et al. described how an online directory of expert witnesses was created from jury verdict and settlement documents using text mining techniques. They created an expert witness directory based on approximately jury verdict and settlement documents, publicly available professional license information, an expertise taxonomy, and automatic text mining techniques. They automatically linked the created expert profiles to medline articles and jury verdict and settlement documents. They used information extraction from text via regular expression parsing, record linkage through Bayesian based matching, and automatic rule-based classification [4].

Other researches [5, 7] presents graph-based models for expertise retrieval with the objective of enabling search using either a topic or a name.

Gollapalli et al. addressed the "similar researcher search" problem for the academic domain. In response to a 'researcher name' query, the goal of their researcher recommender system is to output the list of researchers that have similar expertise as that of the queried researcher. They proposed models for computing similarity between researchers based on expertise profiles extracted from their publications and academic homepages [6].

3 Dataset

We used different datasets in the design and evaluation of our methods. We obtained our researchers data from the DBLP[2] digital library, for processing the large dataset we used methods presented in. After processing we populated researchers data with selected attributes and relationships into our relational database for easy access. We selected and stored the following data about researchers:

- publications,
- co-authors,
- conference or journal the title was published in,
- year and length,
- indexes to other libraries if any are given,
- DBLP index.

The DBLP XML records include a large amount publications and authors, as shown in Table 1. We have not used the latest dataset available from multiple reasons: for our experiments the amount of previously processed data is more than sufficient and mainly because the processing time of the dataset is huge. The DBLP records provide a way to trace the work of researchers and to retrieve bibliographic details when composing lists of references for new papers.

To obtain citation relations among publications, we needed to refer to the CiteSeer[3] database which is a popular search engine and digital library with a collection of 1.2 million scientific documents. In the experiment, we treat data from DBLP as our main information as it has a clear and effective mechanism to process name disambiguation. CiteSeer mainly provides information about citations. We crawled each author and co-author obtained from DBLP in CiteSeer database and paired the found publications from both data source. If a publication obtained from DPLB was not found in CiteSeer, it was discarded, as no citation data is available for it, hence it is useless for our research. If the publication was found, we stored the following details:

[2] http://www.informatik.uni-trier.de/~ley/db/

[3] http://citeseerx.ist.psu.edu/index

Table 1. Statistics of the academic network used in our experiments.

Number of authors	> 30000
Number of papers	> 15000
Number of authorships	> 60000
Number of co-authorships	> 80000
Number of citations	> 30000
Example conferences	KDD, ICDM, VLDB, EDBT, SDM, SSDBM, DASFAA, SIGMOD, ICDE, PODS

- citations,
- BibTeX entry,
- abstract.

The database presents a coherent view of all data with relationships (category, paper, authors, and citations). We obtained and prepared our dataset to analyze the relationships between them and try to create expert profiles for found authors.

4 Expert profile creation

In exploring a comprehensive characterization of expertise, we utilized a multifaceted approach of mining the expertise. The multiple facets are represented by the following measurements: number of publications, number of citations received, extent and proportion of citations within a particular area, expert profile records, and experience. Combining all these factors provides a better indication of expertise with regards to a particular topic. Figure 1 shows the view of expert profile construction as described above. The expert profile computation itself considers the number of publications of an individual, number of citations a person receives, and the person's duration of publication in the respective area.

One of the most challenging issue was how to assign a publication to an ACM classification category. We were considering two basic approaches. The first one is based on keyword extraction and from publication and is mapping the closest ACM category and the second one is mining the conference sites and searching for a topic the publication was presented at. The second approach is more precise in publication to category assignment, as the section or workshop topics are mostly named according to ACM categories. However it is a challenging task to even find a proper data source for conference section mining, moreover no all conferences publishes their sections for public. Mainly for these two issues we have chosen the first approach, assigning a publication to a category according to keywords.

The keywords for a publication are acquired from BibTex records, so we do not have to find and download raw publications and manually extract keywords from them. However the BibTex records does not contain keywords neither from DBLP, nor from CiteSeer. For that reason we had to search for more detailed BibTex records, which we found at ACM DL. Not all publications are found at ACM DL, hence not all publications can be assigned to ACM category. For now we treat the uncategorised publications the same way as the categorised one, but assigned them to a category "other". We grouped the experts into one of two categories:

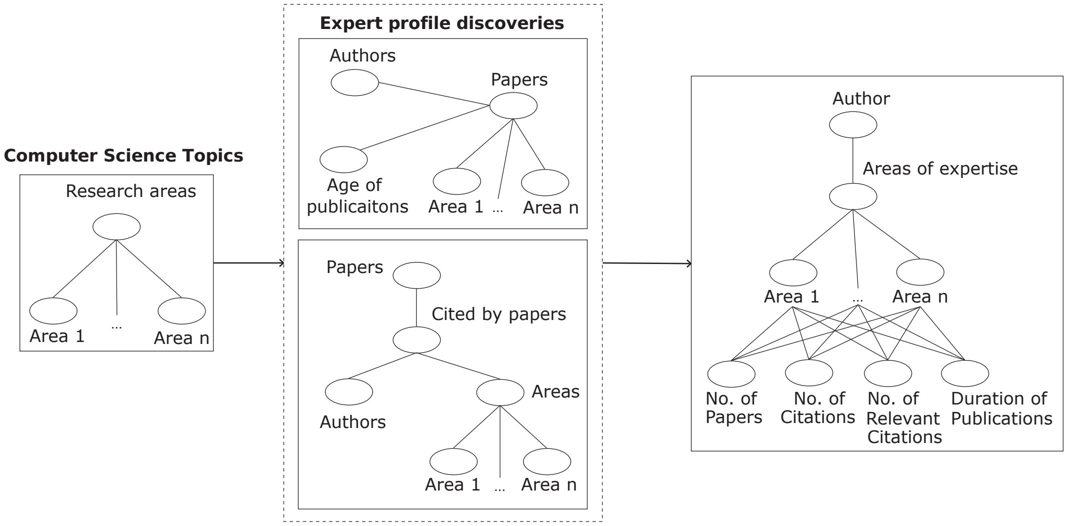


Figure 1. Expert profile.

1. reviewers (persons currently manually assigned as reviewers for a particular ACM[4] topic category),

2. high-profile authors (persons flagged automatically as experts in a particular topic.

High profile authors are computed using our method. Reviewers can be treated as experts automatically and they could be identified from conference sites as well. For now, we treat every author as being a “normal” researcher not considering the possibility to be an expert, on the other hand to deal with this possibility is an important part of our future research.

High-profile authors are calculated based on weights assigned to them. Three weights called publication weight and citation weight are calculated as follows:

Publication Weight = No. of publications / duration (No. of years).

Citation Weight = No. of citations received by an author / total citations.

Total Weight = publication weight + citation weight.

To assign a total weight to an author, we iterated through our optimized dataset and counted a publication weight and a citation weight for each of them based on their publications and citation details. We clustered the publications from authors according to ACM categories by extracting the keywords from the publication’s BibTex entry and assigning it to the closest possible category. So the weight computation was executed for a particular CS area.

Authors are ranked according to their total weight. An author is considered as being an expert if his total weight reaches a certain value. One author can be marked as an expert in multiple information science topics. By mining the research areas and linking high-profile authors to them, we can count the number of experts at a specific research area. We can also find all areas the particular researcher is marked as an expert.

[4] `http://www.acm.org/about/class/class/2012`

Table 2. No. of experts for selected topics using small dataset.

Topic	**No. of experts**
Information Extraction	1
Intelligent Agents	0
Machine Learning	6
Natural Language Processing	0
Semantic Web	5
Support Vector Machine	2

4.1 Evaluation

For evaluating our method we divided our large DBLP dataset into small and large groups. The small group contains approximately 50 authors and their publications along with their citation records, while the large group contains all records shown in Table 1. The main reason of the division was to evaluate the implementation of our method from the point of view of functionality and correctness of ranking. On the small group we ranked authors manually to compare the results with the implemented prototype. While on the small dataset our prototype performed well, unfortunately switching to the large group resulted in inaccurate results, which means that the computed ranks does not corresponds with the expected ones. The computed ranks for authors from small group, who had correct ranks while computing the results using small dataset, differs from the results achieved using large dataset. This means that we need to work out and have to optimalize our prototype to generate correct results to be able to compare with other ranking approaches to evaluate, how our method performs comparing to the existing solutions.

Table 2 shows found experts among authors from our small dataset for selected topics, while the we set the boundary of "being" a potential expert to have at least 3 citations in the last 2 years of at least 3 publications. These boundaries defined only for experimental usage and do not represent constants used by our method.

5 Conclusion and future work

Most current ranking methods address expert identification in enterprise environment. In scientific area the most of the used methods are extensions of famous ranking methods on such as PageRank or Hyperlink-Induced Topic Search. Our method uses multi-faceted approach for providing multiple sources of evidence for a more reflective perspective of experts.

In this paper, we examined a number of methods to create expert profiles and identify experts among authors in different areas, including scientific community based on DBLP dataset. We focused on identifying potential experts in topical areas of a scientific discipline. It is used in the context of a computer science disciplines to identify and assign high-profile authors to areas of computer science, but is easily generalized to other scientific communities. The main contributions of this paper are:

- A method for automatically identifying potential experts from author profiles.
- Generating ranked list of potential experts at each level in the hierarchy.

Our current expert-finding approaches in the DBLP dataset only consider the publications of the experts. To further improve the performance of our methods, we plan to take into account other types of information in future work, such as social information. We would also like to extend the experiments to a large datasets, as well as consider academic networks in other domains that have different publishing practices than computing. Another possible improvement would be applying other techniques for expert identification to our model, such as Statistical language model or Topic-based model proposed in [3].

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Balog, K., Azzopardi, L., de Rijke, M.: Formal Models for Expert Finding in Enterprise Corpora. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06, New York, NY, USA, ACM, 2006, pp. 43–50.

[2] Balog, K., Rijke, M.D.: Determining expert profiles (with an application to expert finding). In: *Proceedings of the 20th international joint conference on Artifical intelligence*. Volume 7., 2007, pp. 2657–2662.

[3] Deng, H., King, I., Lyu, M.R.: Formal Models for Expert Finding on DBLP Bibliography Data. *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 163–172.

[4] Dozier, C., Jackson, P., Guo, X., Chaudhary, M., Arumainayagam, Y.: Creation of an Expert Witness Database Through Text Mining. In: *Proceedings of the 9th International Conference on Artificial Intelligence and Law*. ICAIL '03, New York, NY, USA, ACM, 2003, pp. 177–184.

[5] Gollapalli, S.D., Mitra, P., Giles, C.L.: Ranking Authors in Digital Libraries. In: *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*. JCDL '11, New York, NY, USA, ACM, 2011, pp. 251–254.

[6] Gollapalli, S.D., Mitra, P., Giles, C.L.: Similar Researcher Search in Academic Environments. In: *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL '12, New York, NY, USA, ACM, 2012, pp. 167–170.

[7] Gollapalli, S.D., Mitra, P., Giles, C.L.: Ranking Experts Using Author-document-topic Graphs. In: *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL '13, New York, NY, USA, ACM, 2013, pp. 87–96.

[8] Hashemi, S.H., Neshati, M., Beigy, H.: Expertise retrieval in bibliographic network: a topic dominance learning approach. In: *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management*. CIKM '13, New York, NY, USA, ACM, 2013, pp. 1117–1126.

[9] Meng, Q., Kennedy, P.J.: Discovering Influential Authors in Heterogeneous Academic Networks by a Co-ranking Method. In: *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*. CIKM '13, New York, NY, USA, ACM, 2013, pp. 1029–1036.

[10] Reichling, T., Wulf, V.: Expert Recommender Systems in Practice: Evaluating Semi-automatic Profile Generation. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09, New York, NY, USA, ACM, 2009, pp. 59–68.

[11] Zarandi, M.F., Fox, M.S.: Constructing Expert Profiles over Time for Skills Management and Expert Finding. In: *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. i-KNOW '11, New York, NY, USA, ACM, 2011.

Exploring Multidimensional Continuous Feature Space to Extract Relevant Words

Márius ŠAJGALÍK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xsajgalik@stuba.sk

Abstract[1]

With growing amounts of text data the descriptive metadata become more crucial in efficient processing of it. One kind of such metadata are keywords, which we can encounter e.g. in webpages and can be leveraged to various purposes, e.g. in web search or content-based recommendation.

In our work we focus on vector representation of words to simulate the understanding of word semantics. Each word is thus represented as a vector in N-dimensional space, which includes the advantage of utilising various vector operations like easy similarity measuring between words, or vector addition to compose meaning of longer phrases. We can also calculate what words are the most similar by finding the closest vectors, or vector that encodes a relationship between words (e.g. vector transforming singular into plural). With word vectors, we can encode many syntactic and semantic relations. This is superior over all those manually crafted taxonomies, ontologies and various thesauri, which are often rather imprecise and erroneous. In such hand-crafted data, there is no means of measuring similarity directly between words. Most relations are just qualitative (i.e. described by their type, but we cannot determine the relation quantitatively) and thus all existing similarity methods are limited to achieving only rather imprecise results.

We research the computation of keywords in vector space. This perspective on the keyword extraction problem also brings another new interesting challenges and there are lots of unsolved open problems. So far, we have developed a method for extracting relevant words in clusters of words with similar meaning. Based on our latest results, we conclude that word vectors are sufficient to extract relevant words relatively successfully, even without word frequencies that are used by most methods like TF-IDF, but in the task of keyword extraction we should leverage word statistics to extract just several keywords (to choose the most common terms out of the synonymic clusters). It means that each cluster contains a keyword, but is cluttered with other similar words that often correspond to less common synonyms or their misspelled alternatives, so that computed data is still noisy and needs to be cleaned. We hypothesise that this can be achieved by using frequency statistics, which is our next task to complete.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

[*] Doctoral degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 93-98.

Web Science and Engineering

Student Motivation in Interactive Online Learning

Tomáš BRZA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xbrzat@fiit.stuba.sk

Abstract. In today's age of modern technology, alternative ways of educating are becoming popular. Usage of interactive online learning systems is slowly starting to complement, sometimes even replace, studying in traditional school. This method of learning not only improves students' understanding of underlying concepts but also provides more natural learning process where students can decide their own pace and move to next topics when they themselves feel they are ready. However, students often lose motivation to study on their own using this method. In our project we look at different aspects and types of motivation and existing evidence in order to create elements of interactive learning system that would motivate students into using it more often. We divided motivational elements into three categories which are to be compared among each other to determine actual changes in students behaviour that they caused. In order to understand these changes in behaviour each of them will be evaluated on its own and not just merged together into motivation of a student as a whole.

1 Introduction and Related Work

It is important for students to be motivated to learn and gain new knowledge in fields they are studying. For this reason many teachers try to make their classes as motivating as possible. Modern age enables teachers to use different ways of teaching, one of them being interactive online learning system. This way has great advantages over longer used alternatives that were introduced when Internet didn't exist. One of the advantages of online learning is that student can move at his own pace, repeat any lecture he did not fully understand. This is very difficult in commonly used classes of 20 students who are learning new thing every lesson even if some of them did not grasp the content yet.

Online learning systems also bring certain difficulties, one of them being less human contact with the teacher. This can cause loss of motivation for student as he is not physically present in class and does not feel the teacher looking over his shoulder making sure he is trying his best.

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Jozef Tvarožek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 101–106.

Motivation refers to “the reasons underlying behaviour” [4] and is divided into two categories: *Intrinsic* and *extrinsic* motivation. As Deci et al. [1] observe, “intrinsic motivation energizes and sustains activities through the spontaneous satisfactions inherent in effective volitional action. It is manifest in behaviours such as play, exploration, and challenge seeking that people often do for external rewards”. Extrinsic motivation, on the other hand, is influenced by rewards, whether they are physical, for example money, or not, for example promotion or higher prestige and standing in community.

Educators traditionally consider intrinsic motivation to be of higher importance than extrinsic motivation because it results in better learning results[1]. It is also noted that extrinsic motivation does not have equal effect on all students and it tends to decay over time [8].

Because of its nature, intrinsic motivation is harder to achieve but it is not impossible. There are strategies for increasing intrinsic motivation. One such strategies is to give student more autonomy [5][10]. Many researches argue that providing students with more control over their own learning helps developing a long-term and stable interest in them.

Although some researches [6] claim that extensive amount of extrinsic motivation can work against intrinsic motivation there are also opinions that extrinsic motivation should be used to compliment intrinsic motivation for difficult tasks.

Using cooperative learning methods also fortifies motivation [5][8]. Peer encouragement may improve task engagement. Although it is recommended to group students equally by their ability to perform tasks it is also recommended to create groups with students that focus on different aspects of the task so they can complement each other and complete task that would otherwise require someone with vast knowledge [8].

Enhancing extrinsic motivation in computer software has been used at the birth of video games. Motivational elements used in them were later implemented outside of video games. This method is called *Gamification*. In 2011, Deterding et al. [2] defined gamification as the use of game design elements, characteristic for games, in non-game contexts.

There already are examples of online interactive learning systems, one of the most successful examples is Khan Academy. It is based on short videos that describe solutions to problems, followed by exercises that are focused on the problems shown in videos. Developers of Khan Academy also tried to increase motivation of their users in form of gamification, by introducing badges, points and also by various statistics showing how well is user doing and how far he has gone in his studying.

Another example is programming web site TopCoder which focuses primarily on teaching people how to write optimised algorithms in shortest possible time. It also hosts speed coding competitions. This online learning system also implemented elements from gamification. There are graphical representations of how well users are doing in completing tasks and there are badges to provide milestones for completing tasks.

Motivation is important outside of learning systems as well. An example is StackOverflow which is forum dedicated to helping struggling programmers and IT developers and providing them with suggestions on how to solve issues in their codes. This forum lets users up vote or down vote helpful and not helpful answers so that users are motivate to help as much as they can.

Video games which include gathering points and gaining achievements for performing stunts, are trying to motivate the person playing them to play them more.

From existing learning systems we can see, that gamification is widespread and considered very useful but it is mostly based on extrinsic motivation which can cause issues, if it is implemented as standalone feature.

In this project, we look into different ways of improving motivation that are already used by other systems in order to determine their actual effect on students. We look into different ideas and we compare them to see which are more powerful contributors to increasing students’ motivation and which could potentially cause problems. Since many systems reproduce these elements we want to see their real effects and their priority in implementation into new learning systems that require some basic motivational elements. We are not just looking for evaluating the motivation as whole but as small parts and different behaviours which are all used in calculating motivation in different works.

At the end of this paper we will disclose information about our current progress at this stage of our project and a results we gathered to see if our methods of implementing motivational elements were noticed and used by students.

2 Methods for motivation in online learning systems

In our project we divided motivational aspects into 3 categories:

- Rankings
- Community
- Feedback

We wish to compare these categories and find out which of them have bigger effect on student's motivation and to evaluate their priority for implementation by any other interactive online learning systems who would wish such information.

2.1 Rankings

Rankings involve points for completing tasks, for helping others and badges for completing challenges inside the system. These rankings can be displayed in tables or in graphs.

We propose creating subject-wise ranking of total exercises completed by all students attending subject. This ranking will be displayed in tables sorted by the amount of exercises completed and it will also be shown in profile of each student in graph that will also include average amount of exercises completed and value of the top students. This way student can check his position in table but also see how he was doing in past and how he is doing now using graphs in his profile. These graphs will also try to anticipate how will student do in future if he continues working at pace he is working at current time and therefore student will be able to see the potential outcome of his work and if he is satisfied with it or he needs to work harder than he already does.

These graphs will be displayed publicly alongside with badges for each student in their profile so anyone can see them. This will be required for later use of these elements.

Badges will be awarded to students by proving themselves in completing challenges set by the teachers. For example completing certain amount of task will award student a badge. Since many subjects are divided into smaller lessons, completing each task in one lesson will award student badge for that particular lesson, this badge will represent the fact that students mastered given lesson. Since we are working primary on programming exercises, we will also be implementing execution time ranking and badges. Every nontrivial task will have execution time ranking that will show execution time of top students. Execution time in this instance is meant as time it took the algorithm to complete its calculation. This way students are also awarded if their programs are better optimised and faster which will in turn let them improve on tasks they already completed.

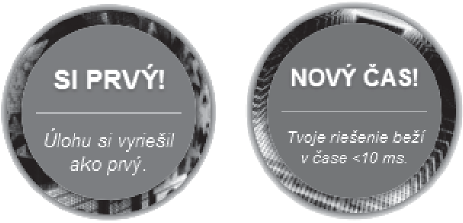

SI PRVÝ!
Úlohu si vyriešil ako prvý.
NOVÝ ČAS!
Tvoje riešenie beží v čase <10 ms.

Figure 1. Examples of badges: FirstToSolveTask, ExecutionTime.

2.2 Community

We think that important part of motivating students is to enable them to be part of larger community. When students sit in class they are a part of community composed of all their classmates. In online learning system this might be harder as students are usually sitting along in their homes and using computer and therefore lack the contact with others. By creating UI elements that let students talk to each other and write on forum, this contact with community is returned to online learning system and students can work together towards goal whether it is just helping each other with simple tips and advices, or by trying to solve tasks together.

We will introduce forum for our students which will enable them to talk about tasks and share ideas and suggestions. This will also create community between students. This way of helping each other can be very good for intrinsic motivation which will be discusses later, but it can also create certain amount of competition between students.

In one of the researches [3] it is noted that even in helpful environments such as discussing forums, this competition exists and drives people into participating in what makes them part of the community. They do not help each other just so that others can benefit by ones information but also to gain higher standing in the community and to earn prestige from others by doing this.

To further increase ways to friendly compete with other students we decided to make their badges and results public, as stated before. This means that students can compare each other’s results and decide to push their limits to prove to others that they are better.

2.3 Feedback

As stated before, intrinsic motivation is very important even compared to extrinsic motivation. Although it is harder to create it, there are ways of making sure it is fortified properly. One of the ways we will implement is by using feedback for struggling students. With each attempt to solve a task that fails because the solution proves to be inadequate, student is given reason for evaluation that returned information that task has not been completed successfully. With this reason provided student knows where he needs to improve and what he needs to focus on. Without such information, student would not know what is wrong and even despite his willingness to successfully complete the task, he might not be able to and that could cause frustration and anger and eventually loss of motivation.

With this feedback in mind, student can truly attempt to complete even harder exercises which will provide more intrinsic motivation based on the success and also more extrinsic motivation that is accompanied by badges and points students will be awarded for completing these harder tasks.

3 First implementation and results for future improvements

We managed to implement some of the functionality mentioned into interactive online learning system used by about 300 students. At this early stage we decided to implement feedback for students when they submit incorrect algorithm, badges and we also created messaging wall present in system that lets students communicate with each other.

As we didn’t have enough data to start analysing as of yet, we still wanted to make sure that these elements were implemented properly and that students noticed them and used them. If these elements were useless or not good enough, we would have to redo.

We asked all students to provide their opinion about these two functionalities, using questionnaire. We asked them how useful these two examples were, in order to determine if it has any impact on their work inside the system. Both of these questions had a second question of similar fashion with open answer which let students give us their ideas on improving or changing these two existing features should they think there is any place for improvement. 220 students responded to this questionnaire.

Table 1. Results of questionnaire for messaging wall.

	What is your opinion on messaging wall?
I liked and used it.	*3.8%*
I liked it and wish more people use it.	*19%*
I did not use it.	*71.9%*
I did not notice it.	*5.2%*

As it can be seen from the table 1, messaging wall was not really a success. Students also filled the open question writing about their dissatisfaction with formatting and structure of this way of communication, asking for a discussing forum with better layout instead. This was a real eye-opener for us as it lets us focus more on forum that would be structured in better way. On the other hand some of the students liked it and showed their concern for unwanted possibility of removing the only way to talk to others because it was not used. So this shows some form of community and chatter is wanted.

The feedback proved to be more satisfactory for students as majority of them used it at some point in their struggles with tasks. This can be seen in table 2.

Table 2. Results of questionnaire for feedback to students.

	How did feedback provided by our system helped you?
It always helped.	*1.4%*
It helped most of the time.	*28.2%*
It helped me sometimes.	*55%*
It did not help me at all.	*2.7%*
I did not notice it.	*12.7%*

From these results we can deduce that feedback is welcomed feature. 1.4% of students that relied on it and might have not been able to complete their tasks without it. 28.2% said it was helpful most of the time which means those times were not catalyst for anger and frustration that would be caused by lack of this feedback. 55% of responders answered with option that indicated minor effect on their work in system. 15.4% of responders either did not use this feature at all or did not even notice it.

These results show that this feature could positively improve and fortify intrinsic motivation. To be sure about exact results of this feature on motivation, we will have to gather more data than this questionnaire provided.

Results from this questionnaire will help us in implementing new and shaping existing features to better suit students and to continue our work in analysing motivation of these features.

Few of the badges that were already implemented seemed to be used quite a lot. Students proved to be curious about badges obtained by their peers but also about their own badges. We can already see that about half of the students is interested in the badges while the other half is not. At this moment it is still too soon to be able to tell more about these two categories of students.

4 Conclusion and Future Work

We completed analysis of the problem of motivating students to use interactive online learning systems more and we manage to make first pilot tests of the three features that were implemented. With these results we can confirm that providing feedback for students, who struggle with completing harder exercises, is a wanted feature that could improve student's motivation. As for our attempt to create community, we see that this feature requires extended work and we have to redesign it to better suit students' needs.

In following months we would like to finish all planned features that were previously mentioned in this paper and we would like to compare each of these elements, to see, which one has bigger effect on students' motivation. This comparison will no more be done only by questionnaire, but also by logging details of each student interests for these elements, gathered within the system. For example by entering pages with these elements, comparing gained badges with other students, using forum to gain hints etc. Knowing which motivational elements are considered more important by higher amount of students will make it possible to compare these elements between themselves.

We have data from before implementation of these elements which shows students' behaviours within the system. We wish to compare these with the data gathered after all planned elements are implemented to compare the difference in students' behaviours. These behaviours include students' persistence in solving difficult tasks, time spent inside the system, amount of attempts on successful and unsuccessful tasks, etc. We won't merge all of these together and calculate some pure value of motivation on its own, we wish to evaluate each of these by itself to see what truly changed and if the change is what we were looking for.

Dividing students into categories based on by what elements they used the most, and looking at these behaviours, we should be able to tell exact changes in behaviour caused by each of motivational elements implemented.

We hope to prove that feedback will increase persistence in trying to solve tasks while having badges rewarded for completing higher amount of tasks will increase amount of tasks attempted and possibly completed.

These results will be based on data from the system, possibly enhanced by questionnaire if the data will not be clear enough, or more questions will be raised that will require it.

References

[1] Deci, E. L., Koestner, R., Ryan, R. M.: A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. In: Psychological Bulletin, (1999), pp. 627-668.

[2] Deterding, S., Dixon, D., Khaled, R., & Nacke, L.: From game design elements to gamefulness: Defining ''Gamification''. In: Proceedings of MindTrek, (2011).

[3] Gottfried, A. E.: Academic intrinsic motivation in young elementary school children. In: Journal of Educational Psychology, (1990), pp. 525-538.

[4] Guay, F., Chanal, J., Ratelle, C. F., Marsh, H. W., Larose, S., Boivin, M.: Intrinsic, identified, and controlled types of motivation for school subjects in young elementary school children. In: British Journal of Educational Psychology, (2010), pp. 712.

[5] Guthrie, J. T., Wigfield, A., VonSecker, C.: Effects of integrated instruction on motivation and strategy use in reading. In: Journal of Educational Psychology, (2000), pp. 331-341.

[6] Hidi, S., Harackiewicz, J. M.: Motivating the academically unmotivated: A critical issue for the 21st century. In: Review of Educational Research, (2000), pp. 151-179.

[7] Pintrich, P. R.: A motivational science perspective on the role of student motivation in learning and teaching contexts. In: Journal of Educational Psychology, (2003), pp. 667-686.

[8] Stipek, D. J.: Motivation and instruction. In: Handbook of educational psychology, (1996), pp. 85-113.

[9] Stipek, D., Feiler, R., Daniels, D., & Milburn, S.: Effects of different instructional approaches on young children's achievement and motivation. In: Child Development, (1995), pp. 209-223.

[10] Turner, J. C.: The influence of classroom contexts on young children's motivation for literacy. In: Reading Research Quarterly, (1995), pp. 410-441.

Knowledge Sharing by Means of Graph-based Diagrams on Web

Terézia KAZIČKOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
terezia.kazickova@gmail.sk

Abstract. We live in information time when get an access to extensive wisdom is not a problem anymore. The question is how to find relevant information and share knowledge with the right audience. Nowadays, the Community Question Answering systems (CQA) play a fundamental role in knowledge sharing. We believe that their potential is not fully employed yet. As CQA systems offer possibility to write questions and answers only as unstructured text, expressing ideas might be often complicated and lead to misunderstandings. Using graphical representation along the text could increase the understanding, especially in fields of engineering. Our goal is to create a web application as an extension to current CQA systems enriched with a possibility to create graphical representations to the questions and answers. We performed a qualitative UX study to determine how web-based diagramming library mxGraph can be used for this purpose.

1 Introduction

Probably the most common way of finding information is by means of web search engines. Although search engines are very useful tools in process of looking for information, in cases of more complex or subjective problems, also human intelligence and discussion are required in addition to computer intelligence and algorithms. As the result, the systems such as Community Question Answering systems (CQA) gained their popularity. CQA systems offer their users possibility to find answers by asking a group of users that form a community willing to share their knowledge. Popular CQA systems include Stack Overflow, Yahoo! Answers or Quora, to name a few.

However, CQA systems play a fundamental role in knowledge sharing, we believe their potential is not fully employed yet. Therefore, we would like to provide more efficient alternative for knowledge sharing within CQA systems. CQA systems offer possibility to write questions and answers only as an unstructured text. We are not aware of any CQA system that would offer a possibility for users to create their own diagrams directly in the CQA system. However, there are some CQA systems that enable to add picture (diagram) created in some other external tool. We believe this form of including graphical representations in CQA systems is not sufficient, since

[*] Bachelor degree study programme in field: Informatics
Supervisor: Ivan Srba, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

users cannot edit their diagrams afterwards. Especially, these diagrams cannot be used by other users in their responses to the original questions.

Following this motivation, our goal is to create a web application named *cqaGraph* which could enrich functionalities of current CQA systems with possibility to create graphical representations as a complement to the questions and answers for better understanding. Main focus being on the domain of software engineering, our priority is to implement support for creation of UML diagrams. We defined two hypotheses, which sum up our goals:

- *Hypothesis 1*: Created application cqaGraph offers required functionality for creating the most common UML diagrams.
- *Hypothesis 2*: Created application cqaGraph offers robust, intuitive and reliable user interface.

This paper presents an architecture and a specification of cqaGraph consisting of functional as well as non-functional requirements. We decided to base the implementation of our application on web-based graphical and diagramming library named *mxGraph*. The possibilities of this library were evaluated by means of qualitative experiment using UX Lab.

2 Related work

Graphical representation brings many benefits into the process of finding and sharing information. It is language neutral and as such shows clear message regardless of native language of the audience. To other benefits of graphical representation belongs possibility to represent possible solutions and create models of certain situation what helps by estimating the risks. According to this reasons, graphical representations are widely used in fields like medicine, where visualizations and simulation help by preparation for difficult operations [3], or in cosmology by planning space expeditions [5].

Graphical representation in CQA systems means the possibility for users to create and edit pictures (diagrams) within CQA systems and add them to their answers and questions as a supplementary visual explanations to their answers and questions in form of text. Although it is generally believed that graphical representation can be a very useful tool for knowledge sharing and also in discussions, most of the CQA system do not offer possibility to express ideas graphically. Partial exception is CQA system Stack Overflow, which enables to insert already existing pictures to questions, however, there is no possibility to add pictures to answers on the original question, neither is it possible to edit this picture later.

In spite of that, there have been several suggestions, how to enrich classical CQA systems with multimedia materials, e.g. pictures or videos. These CQA systems are known as Multimedia Question Answering (MQA) systems. One interesting alternative of MQA system introduced a concept where existing pictures and videos are automatically searched out in the Internet an additionally assigned to the questions and answers [4]. Other interesting alternative of MQA system suggests a framework which enables their users to support questions or answers with a photo, which displays or explains the subject of the question [6].

Graphical representation in software engineering. Graphical representation plays key role also in software engineering during all phases of software development process. Usually, there is more than just one team participating in process of software development, whereas these teams are often in different countries. These teams need to communicate with each other and, moreover, do it effectively. These above mentioned aspects as well as complexity of problems in everyday practice lead us to necessity of using graphical representations. In software engineering, the UML diagrams fulfill these purposes.

Although every type of UML diagram has its purpose in certain phases of software development, some UML diagrams are, in general, more frequently used then other. Brian Dobing and Jeffreys Parsons [2] were searching for an answer at the question, which UML diagram is the most widely used. All respondents of their research were professional analysts with at least 15 years of practical work experience which participated in an average of 27 projects. As Figure 1 shows, the most frequently used is class diagram, then use case diagram and as third follows sequence diagram. Then to less frequently used diagrams belong activity diagram, state diagram and as the least used is collaboration diagram. Based on these findings, we decided to implement functionality of creating the four most common used diagrams: class diagram, use case diagram, activity diagram and state diagram.

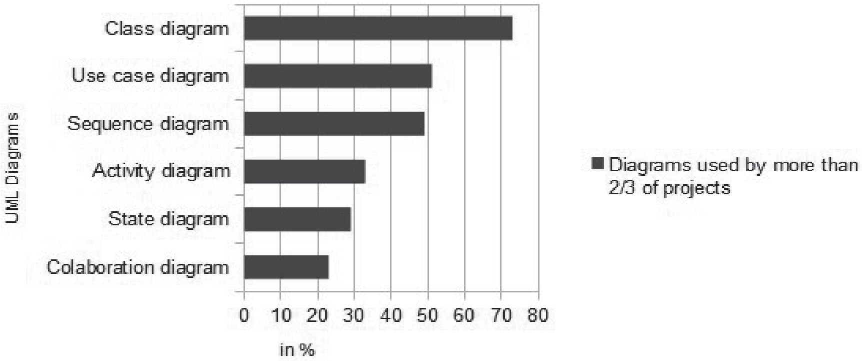


Figure 1. Frequency of UML diagrams' employment in software projects [2].

3 Support of graphical representation of knowledge in CQA systems

We are not aware of any such CQA system used in software engineering, which would enable to create graphical representation of shared knowledge. Therefore, we would like to provide a solution which will solve this problem. More specifically, our goal is to create a web application cqaGraph as an extension to regular CQA systems, which enables to create UML diagrams. We defined following functional requirements of cqaGraph:

- *Support of the most common UML diagrams*: class diagram, use case diagram, activity diagram, state diagram;
- *Diagram persistency and its static representation*, i.e. export into format JPG, PNG, PDF.

Further, we defined non-functional requirements as following:

- *Intuitive navigation*: user knows immediately how to use cqaGraph without reading manual;
- *Easy to use*: creating diagrams easily by drag-and-drop without the necessity of writing code;
- *Robust*: the application is resistant against incorrect input of users;
- *Reliable*: smooth using of the application without any problems;
- *Web-based*: cqaGraph is supposed to be employed in a web environment.

We intend to implement this functionality by means of web-based technology, e.g. HTML 5 and JavaScript. Based on the defined functional and non-functional requirements we were looking for such graphical library, which would support creating of diagrams. According to extensive study of seven libraries, we identified the mxGraph[1] as the best fit for this purpose. This library represents the important element in the architecture of cqaGraph (see Figure 2). In addition, mxGraph provides several example applications which are not, however, specialized on UML. Therefore, we

[1] http://www.jgraph.com/mxgraph.html

decided to perform an UX study on one of these example applications to identify how mxGraph library can be employed to create cqaGraph which satisfies the defined requirements.

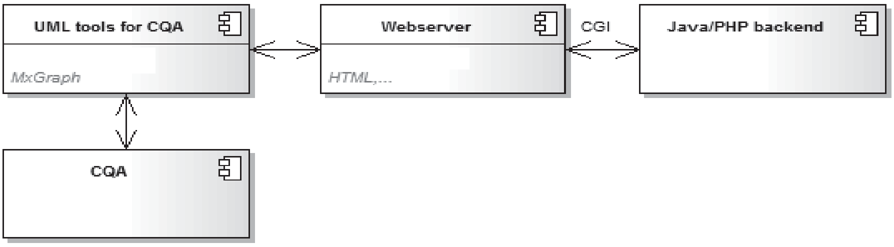


Figure 2. Architecture of cqaGraph.

4 UX Study

UX lab is a laboratory for observation of human-computer interaction. We decided to perform the qualitative UX study in order to specify users' requirements and expectations as well as determine the possibilities and restrictions of diagramming library mxGraph. UX lab is formed by a computer with web camera, eye-tracking sensor and software, which simultaneously records voice, visual stream, mouse moves as well as intensity and trace of user's gaze across the screen and, most importantly, this software enables also further analysis of gained data. Results of such an experiment offer very important insight into the way how users work in the environment of the application, what features are beneficial for them and which are making their work more difficult.

We employed this kind of UX lab in a qualitative experiment, in which three respondents participated. All of them have very good experience with UML and are used to work with several different UML modelers. For purposes of the experiment, a sample application based on mxGraph was used which can be found at *www.draw.io*. Figure 3 shows user interface of the used sample application with a created sample class diagram.

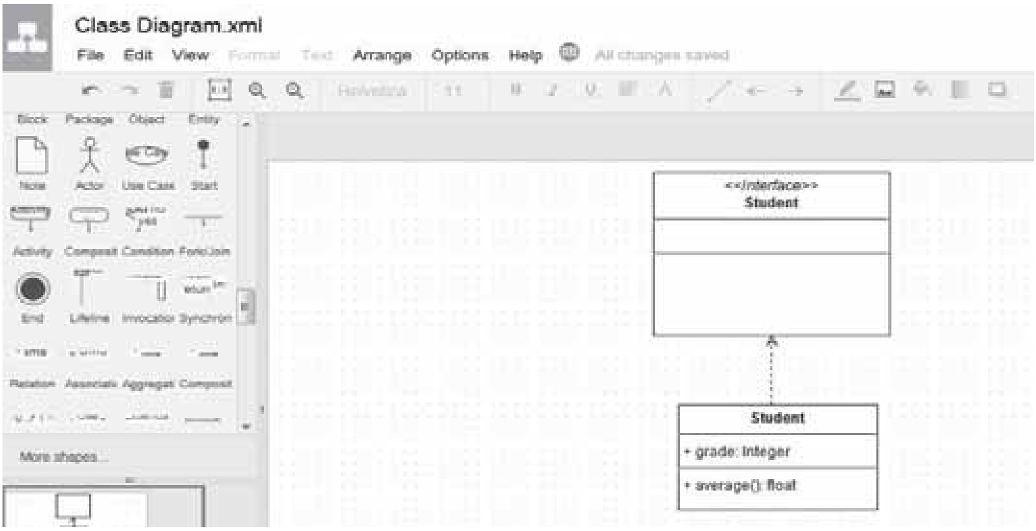


Figure 3. Sample application (www.draw.io) used during UX study.

In spite of that, this sample application enables to create various kinds of diagrams, it is not primarily intended for creating UML diagrams. However, it is sufficient for purpose of our experiment because it presents the necessary possibilities of mxGraph library.

Our experiment consisted of two parts. In the first part respondents were given a set of simple instructions which led them in process of creating a simple UML diagrams: class diagram and use case diagram. The second part consisted of six questions related to non-functional features of application. During the whole experiment, respondents were asked to express their opinion and comment on the currently performed action in the application. They could also make suggestions what to change. Data collected during experiment can be displayed in different ways: heat map, gaze plot, moves of the cursor across the screen.

If we are able to interpret the meaning of gained data in the right way, UX lab technology can bring us valuable information. We have to choose from different types of data we gain and afterwards set up the accurate methodology for evaluation of the selected data. Although there are metrics for statistics calculation and many algorithms (e.g. regarding time of gaze fixation), using eye-tracking still demands certain degree of subjective assessment of the situation [1].

For our purposes, very important information are represented by mouse moves across the screen, as it shows not only how long it took user to find out certain functionality or object, but also it shows how many times user had to click at the object to perform desirable action. This indicates, whether the user interface is intuitive enough as well as if it keeps up with commonly used standards. If it does not, using the application causes user problems and makes his work slower what lowers his motivation to use the application. Other very valuable information is user's comments and opinions about the application. We divided the feedback from respondents into three categories. In the following overview, we provide the most important findings in each of these categories:

1. *Feedback to organization and navigation of application*. All respondents suggested that the application should enable to change properties of the elements by changing its' attributes by means of the context menu.
2. *Feedback to overall transparency and layout*. All respondents would prefer if the elements of UML diagrams were organized into categories according to UML diagram type so it would be easier to find what they need.
3. *Feedback to unexpected responses of application*. The application does not keep up with standards for application controls what makes working with the application confusing. This was also noticeable at the mouse movements recording. Respondents were given a task to place element class onto the plane and rename it as *Student.* By analysis of gained data of mouse movements, we chose the time interval since placing the element class to the plane until the respondent successfully renamed the element. Average count of left-mouse clicks was 7.33 clicks for renaming the element (see Figure 4). Such high number of clicks for such a simple task indicates that users had difficulties finding the right way for editing the element and therefore had to select the element more times.

5 Conclusions

Graphical representation plays fundamental role in many disciplines. Our main goal is to support graphical representation of knowledge, which supplements questions or answers in the question answering process in CQA system. We proposed the architecture and the specification of the application cqaGraph as an extension to custom CQA system. The proposed application is supposed to be employed in the field of software engineering and thus it focuses on support of UML diagrams.

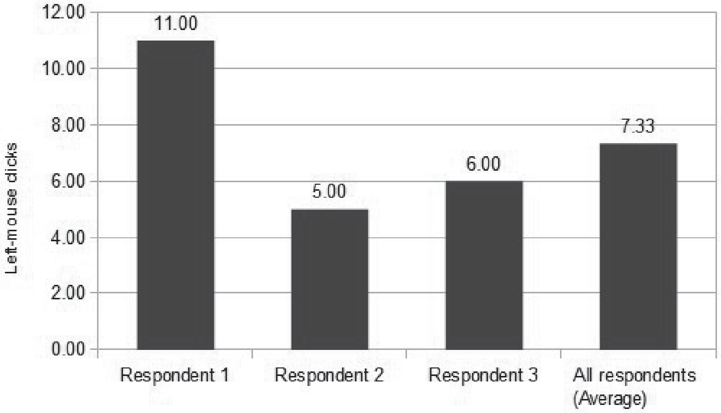


Figure 4. Count of left-mouse clicks of each respondent by renaming element Class and average count of necessary mouse clicks.

The implementation of such application is very complicated and cannot be performed without a supplementary graphical library. Therefore we conducted the qualitative UX study to determine reactions and expectations of users as well as to determine the possibilities and restrictions of diagramming library mxGraph.

The results of this UX study represent important input to development of cqaGraph, such as how the application should be structured (e.g. organization of UML elements according to different kinds of supported UML diagrams) or what the expected behavior for various users' actions is (e.g. the provided content of context menu). We believe that cqaGraph has a potential to become a helpful tool in knowledge sharing in software engineering.

Acknowledgement. Author would like to thank all respondents who participated in UX study for their helpful comments which helped us to evaluate mxGraph library. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Al Maqbali, H., Scholer, F., Thom, J. A., Wu, M.: Using eye tracking for evaluating web search interfaces. In: *Proceedings of the 18th Australasian Document Computing Symposium on - ADCS '13,* ACM Press, (2013), pp. 2-9.

[2] Dobing, B., Parsons, J.: How UML is used. In: *Communications of the ACM*, (2006), vol. 49, no. 5, pp. 109-113.

[3] Johnson, C. R., Weinstein, D. M.: Biomedical computing and visualization. In: *Proceedings of the 29th Australasian Computer Science Conference,* (2006), vol. 48, pp. 3-10.

[4] Nie, L., Wang, M., Zha, Z., Li, G., Chua, T.: Multimedia Answering : Enriching Text QA with Media Information. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval,* (2011), pp. 695-704.

[5] Zhang, W., Pang, L., Ngo, C.: Snap-and-ask: answering multimodal question by naming visual instance. In: *Proceedings of the 20th ACM international conference on Multimedia - MM '12,* ACM Press, (2012), pp. 609-618.

[6] Wright, J., Burleigh, S., Maruya, M., Maxwell, S., Pischel, R.: Visualization experiences and issues in deep space exploration. In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, (2003), pp. 619-621.

Recommendation Based on Parallel Browsing

Viktória LOVASOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
viktoria.lovasova@gmail.com

Abstract. Recommendation systems are an important part of learning systems. Parallel browsing may represent a type of implicit feedback for use in recommendation – including actions such as switching between websites, spending time on a site, opening links into new tabs or reusing existing ones. We propose a recommendation method based on students' parallel browsing in the adaptive learning system ALEF. We evaluate the method through an experiment, in which students will learn in the system and get mixed recommendations from a sequence recommender and from a parallel recommender. There should be more clicks on the recommendations generated from the parallel recommender.

1 Introduction

When people need an advice, their typical habit is that they look for it on the internet. Although in the past, the searching was harder for them, because they had to decide, which option is the best. Today recommendation systems help them recommend items, which could be interesting for them. They are part of almost every web site, for example electronic shops, movie or music systems and restaurants.

It is very common nowadays, that students learn via adaptive web based learning systems. Every user learns by a different tempo though, that is why it is appropriate to estimate a user's level of knowledge of the learning object and recommend him the most useful item.

Thanks to modern web browsers the user has the option to view different web pages at once in multiple windows or tabs, so he can then focus on one page and switch between the others. When the user concentrates on one page and has another one opened in a second window or a tab it may mean, that he would like to come back to it in the future, because he might be interested in it. Similarly the user can open a new page from the current one as a link, which could be useful for him. On the other side, if he opens a new page and in a short time closes it, that can mean that he did not find the information, he was looking for. If we could capture user's behaviour with the use of parallel web browsing while he is learning, it could help us make better recommendations.

Therefore we propose and implement a recommendation system in ALEF based on users' parallel web browsing with the goal to improve recommendations. ALEF is an adaptive learning

[*] Bachelor degree study programme in field: Informatics
Supervisor: Martin Labaj, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 113–118.

education system being developed at the Faculty of Informatics and Information Technologies, which is used as a tool to educate students in several courses.

2 Recommender systems

Recommender systems provide the user with a list of recommended items they might prefer, or predict how much they might prefer each item. These systems help users to decide on appropriate items, and ease the task of finding preferred items in the collection [8].

The most used recommender systems are collaborative filtering and content based systems. The difference between them is that collaborative filtering systems utilize the given ratings of training users to make recommendation for test users while content-based filtering systems rely on contents of items for recommendation [10]. In our research, we chose to use collaborative filtering, it is probably the most widely used recommender method [4], it is also completely independent of any machine-readable representation of the objects being recommended.

Collaborative filtering algorithms can be divided into 2 main categories: memory-based and model based algorithms [3]. Memory-based algorithms use the user-item database to make a prediction. These systems find users known as neighbours that have a history of agreeing with the target user. When the neighbourhood of users is made algorithms are used to combine the preferences of neighbours to produce a prediction. Model-based algorithms first develop a model of user ratings – they use various algorithms, for example: Bayesian network, clustering, and rule-based approaches. Then they compute the expected value of user prediction, given his ratings on other items.

2.1 Explicit and implicit feedback

All recommender systems need users' interests in order to make good recommendations. A common approach to building a user preference model is by capturing feedback from the user, either explicitly or implicitly. Explicit feedback is more accurate than implicit [1], the user can show more exactly how much he likes or dislikes the items by Likert scales, or questionnaires. On the other hand implicit feedback is abundant. The system observes user's actions – e.g. if the user listens to a track 5 times, he probably likes it [6].

Parallel browsing could represent a type of implicit feedback, users express their opinion without knowing it, and so they act more naturally. Even if the explicit feedback is more accurate users do not always rate the items.

2.2 Parallel browsing as an input for recommendation

In [2] they proposed a recommendation model called TABAKO, which is based on all-k^{th}-order Markov model and extracts linear sessions from active user's session. The new recommender model showed to lead to a more accurate system than all-k^{th}-order Markov model.

In [7] they proposed a model for parallel browsing behavior based on events observed by client-side scripting in the ALEF educational system. Within the frame of study 143 out of 254 users were using parallel browsing. They could identify new relations that are not represented in the domain model that can be used for better recommendation.

A study of tabbed browsing was done in [5] where they learned the main reasons for people using tabs: reminders, opening links in background, multitasking and others. Parallel browsing became a common activity, which users do when they are on the internet so we can utilize it to make recommendations better in technology enhanced learning.

3 Recommendation based on parallel browsing in learning systems

Users can browse the internet parallel by having different sites open concurrently in more tabs or windows and switch between them. In [7] they made a research about parallel browsing in ALEF and demonstrated, that students actively browse ALEF in parallel tabs. We propose a recommendation model based on parallel browsing in ALEF, which is described more closely in the next section.

3.1 ALEF

ALEF is an adaptive learning framework [9] developed on the Faculty of Informatics and Information Technologies in Bratislava and used by students in several courses: Principles of Software Engineering, Functional Programming, C Language Programming.

The system makes use of 2 basic models: a domain model and a user model. The *domain model* is the subject of education, it consists of objects content and metadata, which are interconnected with various relations. Learning object inherits from content and in our study we recommend learning objects. There are 3 types of learning objects in ALEF:

- Explanation – describes a subject domain.
- Question – represents an interactive part of the system, provides an immediate feedback to the student’s knowledge.
- Exercise – allow students to practice gained knowledge.

3.2 Tracking the parallel browsing

We track user’s behaviour via client-side script, which is included in ALEF websites. We cannot know what sites other than from ALEF are opened by the user, because the script scope is only within the system being tracked. Because we recommend only learning objects from ALEF, this is not a severe limitation.

We consider these actions important for our research:

- Page load
- Page unload
- Time spent on the page
- Switching between tabs

If the user opens a target page in a new tab from the current source page and is active there for some time, that means he found useful information there. We take the time into account, because when the user is focused for too short and closes the tab semi-immediately, he didn’t find the right item for him. When he spends there e.g. 3 or 5 minutes (depending on the type of object and the object itself), there is a connection between the source and the destination learning object and we could recommend the destination learning object. If he stays on the new site too long, that means he could left the room or simply is not focused.

3.3 Recommendation

According to such browsing behaviour, one learning object could rate another one using the user’s tab switches between them. We proposed the following formula where $R_{i,j}$ is the rating for learning object, v is a value which is added to $R_{i,j}$ according to the time spent on the tab in seconds.

$$R_{i,j} = R_{i,j} + v \tag{1}$$

Let LO_1, LO_2 and LO_3 be learning objects in ALEF and suppose that the user switches from one tab with LO_1 to other tab with LO_2 and there spends 2 minutes, then he switches back to LO_1 (Figure 1). From these actions, we can assume that there is a connection between LO_1 and LO_2, because the user spent some time on LO_2. When another user visits LO_1, we can recommend LO_2.

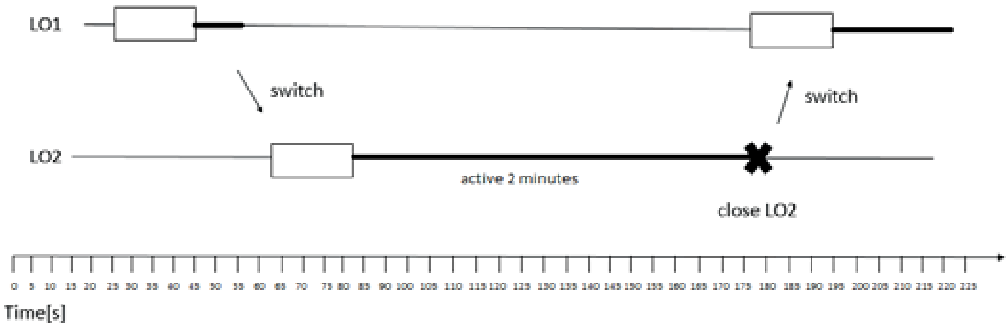


Figure 1. Switching from LO_1 to LO_2.

In Figure 2, the user switches from LO_1 to LO_3 where he spends only 10 seconds and then he closes the tab. We suspect that LO_1 and LO_2 do not have any connection. Although to recommend items, we use item based collaborative filtering to recommend appropriate learning objects according to users' rating tables. One student may close the tab because he already knows what he sees and other student closes it because he did not find the relevant information.

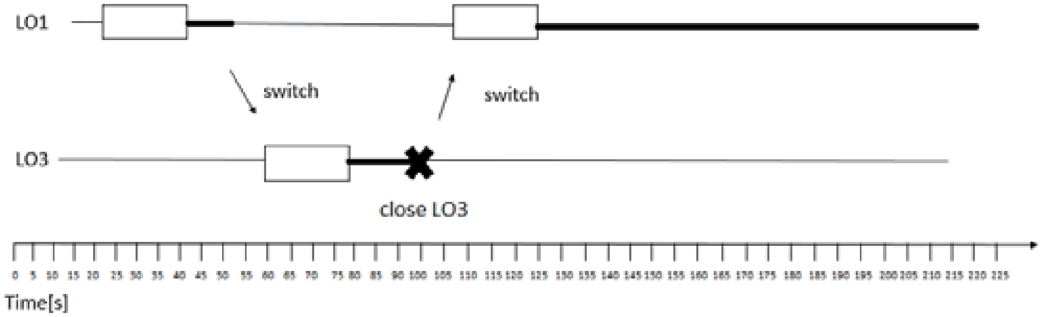


Figure 2. Switching from LO_1 to LO_3.

Table 1 illustrates how actions from previous Figures 1 and 2 affect the ratings on learning objects. According to the time spent on learning object the previous learning object rated the destination learning object. If the time was too short as in Figure 2, we do not take this into account and the rating remains unchanged, if the time was reasonable as in Figure 1, LO_1 raised the rating for LO_3 by $v = 120$, where v denotes the time in seconds spent on the site, $R_{i,j}$ is the rating for LO_j from LO_i:

Table 1. User´s rating table.

UserID: 9129	**LO_1**	**LO_2** (LO_j)	**LO_3**
LO_1 (LO_i)	-	$R_{1,2}$ $(R + v)$	R
LO_2	R	-	R
LO_3	R	-	-

With the use of this method, if a user u_x is visiting a learning object – e.g. explanation-type learning object LO$_5$, the recommender system looks up the most similar m users (neighbors) based on co-rated items and computes their similarity $sim(u_u, u_x)$ according to their browsing patterns (tab switch tables), where u_u is a user which is similar to u_x for whom we want to recommend other LOs. From these neighbors, using their switch/rating rows for LO$_5$, the rating for every LO$_j$ for u_x is computed:

$$R_{i,j}(u_x) = \frac{\sum_{u_u=0}^{n} sim\,(u_u, u_x) * R_{5,j}(u_u)}{n} \tag{2}$$

After this computation, we choose the top-n (initially 3) ratings $R_{i,j}(u_x)$ and recommend corresponding learning objects.

To prove that our method improves the recommendations in an adaptive web-based learning system we compare it with a sequence recommender. It is a content-based recommender, which finds the most similar items and items following the progress of the course not yet seen by user.

4 Evaluation

4.1 Relations between objects to be recommended

Since the switch/rating tables express relations between learning objects inferred from user behaviour, we first evaluate the underlying data. As the ALEF system is being used in two courses (Functional and Logic Programming and Principles of Software Engineering), the tabbing data is collected.

We can evaluate the content-based similarity between the object rated by user switch pairs by computing their cosine similarity on the concepts assigned to them in the domain model. If the user switch pairs correlate with the content-based similarity, this can mean that this parallel browsing based algorithm can be used instead of content analysis in domains, where such analysis is costly – e.g. video or image recommendation.

If the similarity is not achieved, it can mean that the browsing behaviour express different relations than those stemming from the content. In that case, we will prepare a dataset by selecting pairs rated high according to browsing behaviour (our method) together with those rated according to content similarity, and, as a control, we mix in pairs rated low and random pairs. A domain expert for the specific domain (course) will then evaluate the switch pairs to identify relations.

4.2 Recommendation experiment

We plan a closed experiment with 20 students from our faculty. The participants will receive instructions written down on a paper – based on the results from previous evaluation (what relations switch pairs express), we will choose an appropriate task. Students will have one hour to study a specific section of the Functional and Logical Programming course within ALEF system. All of them will have a widget on the right side, where there will be recommendations for them. Half of the recommendations will be generated from the sequence recommender and the other half will be generated from the recommender based on parallel browsing. They will be displayed intermixed together.

After one hour, there should be more clicks on the recommendations generated from the parallel recommender than from the sequence recommender. The second option is that during the experiment, the recommended objects will be logged in the database. The recommendations generated from parallel based recommender and from sequence recommender can be evaluated against each other.

We aim for realizing the recommendation method in the live system for use in the aforementioned courses. In a similar manner to the closed experiment above, we can then perform an uncontrolled experiment where students use the system over longer time periods to prepare for seminars and exams.

5 Conclusions

We proposed a method for recommendation based on parallel browsing, which should improve recommendations in ALEF. The value of ϑ in the experiment is set only preliminary, it will change according to experiments.

Our recommender can be used in any learning system where learning objects are present. One of its advantages is that the system does not recommend items based on their content or metadata (concepts) so it can be not only used in courses not annotated with concepts (not having domain models), but also in different areas–in an e-shop, when the user switches between tabs, he may be comparing related products, in Wikipedia, he might be opening relevant links, etc.

A specific use can be found in domains where the content analysis is too expensive. Our method can be used since it does not need to know any metadata about the content, only user behaviour over the items. Typical examples are movies, pictures, videos.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Amatriain, X., Pujol, J.M., Oliver, N.: I like it ... I like it not : Evaluating User Ratings Noise in Recommender Systems. In: *User modeling, Adaptation, and Personalization*, Springer, (2009), pp. 247-258.

[2] Bonnin, G., Brun, A., Boyer, A.: Towards tabbing aware recommendations. In: *Proc. First Int. Conf. Intell. Interact. Technol. Multimed. - IITM ’10*, ACM Press, (2010), pp. 316-323.

[3] Breese, J.S., Heckerman, D., Kedie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: *Proc. of 14th conf. on Uncertainty in Artificial Intelligence - UAI'98*, Morgan Kaufmann Publishers, (1998), pp. 43-52.

[4] Burke, R.: Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, (2002), vol. 12, no. 4, pp. 331-370.

[5] Dubroy, P., Balakrishnan, R.: A study of tabbed browsing among mozilla firefox users. *Proc. 28th Int. Conf. Hum. factors Comput. Syst. - CHI’10*, ACM Press, (2010), pp. 673-682.

[6] Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of implicit and explicit feedback from an online music recommendation service. In: *Proc. 1st Int. Work. Inf. Heterog. Fusion Recomm. Syst. - HetRec ’10*, ACM Press, (2010), pp. 47-51.

[7] Labaj, M., Bieliková, M.: Modeling parallel web browsing behavior for web-based educational systems. In: *Emerg. eLearning Technol. Appl. (ICETA), 2012 IEEE 10th Int. Conf.*, IEEE, (2012), pp. 229-234.

[8] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommnender System Handbook, Springer US, (2010).

[9] Šimko, M., Barla, M., Bieliková, M.: ALEF: A framework for adaptive web-based learning 2.0. In: *Key Competencies in the Knowledge Society*, *IFIP Advances in Information and Communication Technology*, Springer, (2010), pp. 367-378.

[10] Zhai, C., Si, L. & Jin, R.: A study of mixture models for collaborative filtering. *Information Retrieval*, (2006), vol. 9, no. 3, pp. 357-382.

Presentation of Snippets in Web Summarizers

Filip MAZÁN*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
filip.mazan@gmail.com

Abstract. The purpose of this paper is to analyze existing solutions of displaying summarized content and find the most appropriate way to do so. Various methods are compared to each other and verified by our own built testing platform summarizing news articles parsed from various external sources. The experiment will take place on a relatively large sample of users. The result is therefore to compare the suitability of each method for displaying summarized content and to create a general overview of the tested display options narrowed to the area of news articles aggregators, which could be used to improve the services of existing news aggregators.

1 Introduction

Everyday internet users must regularly navigate in large amounts of information they find of the web, which is often a difficult task. Users usually spend a lot of time with data filtering, which could be better used for already filtered-out information they are interested in. That is why more and more content summarization projects were created – whether it is summaries of news articles, cheap flight tickets or ski resort reviews [1]. One of the issues content summarization project must deal with is a question how to display the summarized content to users.

For the purpose of this paper, we must first declare a term *snippet*. Snippet is a graphic element of the web page usually of a relatively small area, which contains the most important information about the referred entity. In the case of news articles, the snippet could contain a title, short description, representing image, publishing date or author name. Our goal is therefore to find the best possible layout of snippet components, which is well-arranged, does not distract the user, provides useful information which is user interested in and invokes a faster action – whether to explore the content more or reject it and skip to the next snippet.

In the area of news aggregators it is a positive step to offer the users only the content which they are interested in and hide the content which does not fall in their areas of interest. This way the time needed for searching for interesting content decreases and user has more time for the useful rest. This part of the summarization process can be called as a content recommendation part. As this part covers a wide variety of methods how to recommend content, this paper will not explore them all in depth.

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Jozef Tvarožek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 119–124.

The objective of this paper is to analyze existing solutions of displaying summarized content and find the most appropriate way to do so by means of creating a web-based testing platform aimed at summarizing news articles and showing them to the users. The platform consists of more interconnected parts from which the most important ones that show the content to users in form of snippets and capture the behavior of users for later analysis.

2 Related work

The textual part of presentation of the snippets has been studied many times, but there is a limited source of studies which deal with layout and graphic aspect of snippets shown to users. The paper published by Hideo J. and Joemon J. indicates that a simple experiment was ran in the area of search engines. Four layouts of search result snippets were designed which consisted of different components including thumbnail, URL, result size, description and even top ranking sentences pulled out of the linked result. The experiment was held with 24 test subjects and concluded that less experienced users preferred snippets with more information available whilst more experienced users preferred simple snippet layouts. The test also concluded that additional components were beneficial to some aspects of searching, but there was not found any evidence that suggested either of textual and visual representations were consistently useful across searches [6].

The study performed by Dziadosz S. and Chandrasekar R. compared usage of three types of snippets shown in a search engine by an experiment with 35 subjects. Although this work studied only comparison of text-only, thumbnail-only and combined approach, authors successfully concluded that using combined snippets helps with correct decisions and reduced error rates at a little cost of mental processing time compared to text-only and thumbnail-only snippets [7].

The paper published by Young G. H. et al. specifically describes Really Simple Syndication (RSS) format, structure and its role and usage in the area of news aggregators which helped us with designing our testing platform [8].

Our testing platform makes use of basic recommendation of news articles. As it was mentioned in introduction, there is a variety of methods how to do so. There are three basic categories of recommendation – content-based recommendation, collaborative filtering and hybrid methods [5]. Content-based recommendation is built on the descriptions of the referred items and their properties together with the profile of user’s interests. Collaborative filtering is based on collecting the data about behavior of users and their preferences. Our testing platform employs mostly content-based recommendation with an element of collaborative filtering, which was inspired by Google News recommendation system [2].

3 Snippet analysis

Before the experiment itself, it was needed to look for as most snippet types on the web as possible. For this purpose we collected data from 33 different web pages including news aggregators, search engines and online newspapers. The snippets shown on those web pages were analyzed - we observed the shape of the snippet, its components and the layout of the components. We identified the following basic types of the components:

- Title is a heading usually summarizing the content in several words.
- Description contains a summarization of the referred item in a larger scale than the title usually in few sentences.
- Image is directly or indirectly related image to the item. It is usually the first user’s contact with the snippet as the image recognition in humans is faster than text recognition.
- Category is a means of defining item similarity or belonging to group of items which have some properties in common.

- Publish date helps the user to find out the “freshness” of referred item and hence it’s relevancy and desirability.
- Popularity is often measured and shown as a number of users who have visited the referred item.
- Number of comments (if they are supported by system) similarly to popularity does not relay any information about the content, but shows the general interest of the public.
- Author’s name may be an important component as some users prefer items posted by an author they like. Similarly, some users may not like posts by particular author and will avoid them.
- Similar items show one or more related items to the primary one in one snippet.
- Source of the item is an important factor too, some users may prefer content from particular sources.
- Rating captures explicit feedback from users, usually in form of 5 star rating or rating on scale of 1 to 10.
- Social network connectors are similar to popularity – this component also tells nothing about the content, but contains information about the general fame of the item.

Another important aspect of the snippet is its shape. However, the sample of the snippets we analyzed contains a wide spectrum of diverse shapes – one line snippets, horizontal rectangles, squares, vertical rectangles, towers and their combinations.

The samples were divided into three categories based on the content they refer to – news aggregators, daily papers and search engines. Since the category this experiment is mainly about is news aggregators, the Table 1 describes the relative frequency of snippet components used. Although the sample is relatively small – we could only find 6 usable news aggregators, it gives us better understanding on which components they mainly use to render snippets. The average number of snippets per one webpage is 80.67 with relatively large standard deviation of 86.27.

Table 1. Relative frequency of snippet components in news aggregators.

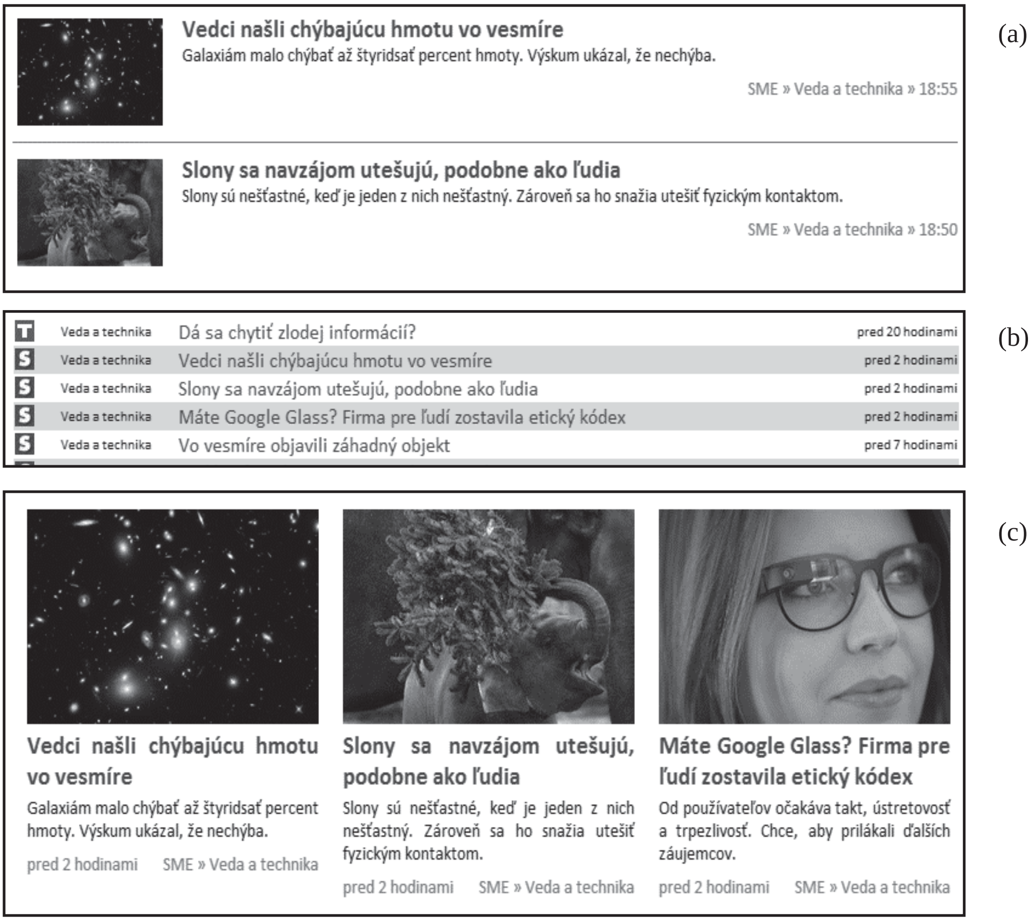
Snippet component	Relative frequency
Title	100.00 %
Description	100.00 %
Source	83.33 %
Image	66.67 %
Social network connectors	66.67 %
Category	50.00 %
Similar news	33.33 %
Author	33.33 %

4 Experiment design

The testing platform consists of several connected parts. The first part handles gathering fresh news articles from various sources and storing them in the database. Next part controls the recommendation aspect, so the users can see fitting articles they are interested in on the front page. The crucial part is responsible for rendering the snippets to users and recording their behavior.

The system provides several variants of snippets by utilizing A/B/..Z testing method, where each user is assigned with one of those variants during the whole testing period [3]. Snippets shown in Figure 1 were chosen according to the results of the observations of other news aggregators and they differ in the following aspects:

- number of snippets per web page,
- snippet shape,
- snippet components.



Vedci našli chýbajúcu hmotu vo vesmíre
Galaxiám malo chýbať až štyridsať percent hmoty. Výskum ukázal, že nechýba.
SME » Veda a technika » 18:55

Slony sa navzájom utešujú, podobne ako ľudia
Slony sú nešťastné, keď je jeden z nich nešťastný. Zároveň sa ho snažia utešiť fyzickým kontaktom.
SME » Veda a technika » 18:50

T	Veda a technika	Dá sa chytiť zlodej informácií?	pred 20 hodinami
S	Veda a technika	Vedci našli chýbajúcu hmotu vo vesmíre	pred 2 hodinami
S	Veda a technika	Slony sa navzájom utešujú, podobne ako ľudia	pred 2 hodinami
S	Veda a technika	Máte Google Glass? Firma pre ľudí zostavila etický kódex	pred 2 hodinami
S	Veda a technika	Vo vesmíre objavili záhadný objekt	pred 7 hodinami

Vedci našli chýbajúcu hmotu vo vesmíre
Galaxiám malo chýbať až štyridsať percent hmoty. Výskum ukázal, že nechýba.
pred 2 hodinami SME » Veda a technika

Slony sa navzájom utešujú, podobne ako ľudia
Slony sú nešťastné, keď je jeden z nich nešťastný. Zároveň sa ho snažia utešiť fyzickým kontaktom.
pred 2 hodinami SME » Veda a technika

Máte Google Glass? Firma pre ľudí zostavila etický kódex
Od používateľov očakáva takt, ústretovosť a trpezlivosť. Chce, aby prilákali ďalších záujemcov.
pred 2 hodinami SME » Veda a technika

Figure 1. Snippet types. (a) Screen wide snippets, 20 per page. (b) Screen wide snippets, 50 per page. (c) Three snippets per row, 30 snippets per page.

During the test, the implicit behavior of users is recorded, including but not limited to timestamps of logins, timestamps and the locations of clicks of users, overall activity of users and their retention [4]. After the testing period the explicit feedback will be collected by a survey sent to all the participants.

The system contains an article gathering algorithm, that periodically scans predefined RSS feeds of news providers and for each feed iterates through the articles. If the current article has not been saved in the database yet, it is added and continues iterating until it finds an article which already was in the database. After that, it continues parsing the next RSS feed. The gathered data about the article consists of the title, description, publish date, article URL, image URL, source identification and the category.

To deliver the most interesting content to the users, the system uses content-based recommendation with a small portion of collaborative filtering. Every article is assigned a score value defining a probability of the user liking that particular article. The requirement for this is logging the user's activity and creating user's profile. The profile consists of two vectors. The first one describes user's interest in article categories and is calculated with the Equation 1 below, where $D(u,t)$ is the vector of interest in categories of user u during the time period t, N_i number of clicks on articles belonging to category c_i and N_{total} is a total number of clicks on articles of every category in the defined time period.

$$D(u,t) = \left(\frac{N_1}{N_{total}}, \frac{N_2}{N_{total}}, \dots, \frac{N_n}{N_{total}}, \right) \tag{1}$$

Similarly, the vector of source preferences is calculated with the Equation 2, where $S(u,t)$ is the vector of interest in article sources of user u during the time period t, N_i number of clicks on articles originating from source s_i and N_{total} is a total number of clicks on all the articles in the defined time period.

$$S(u,t) = \left(\frac{N_1}{N_{total}}, \frac{N_2}{N_{total}}, \dots, \frac{N_n}{N_{total}}, \right) \tag{2}$$

The collaborative part of the recommendation relies on the calculation of a popularity of an article in a specified time frame. The final Equation 3 defines how the final scoring is calculated is shown below, where $ctime$ defines current date, $ptime$ publish date of article a. Index $popularity$ denotes a general popularity (relative frequency of article a) in a specified time frame and provides a small score boost.

Note that only articles which have been read by user for a longer period of time are accounted into statistics and recommendation. This makes sure that users are not be falsely assigned with categories of articles which they have closed immediately.

$$\begin{aligned} Score(u,a) = 0.6\,.\,D(u,t,c_a) + 0.15\,.\; S(u,t,s_a) + 0.25\,.\max\left(0, 1 - \frac{(ctime - ptime)}{2\ days}\right) \\ + popularity \end{aligned} \tag{3}$$

5 Preliminary evaluation and future work

Once the first prototype of the testing platform was set up, we performed a small experiment to test out the system stability, ability to measure correct data, overall appearance and usefulness. The preliminary test was held with 6 subjects who are everyday internet users aged 22-36 with university education which were assigned different snippet types. Subjects were asked to use the news aggregator platform as ordinary users to read the news articles they were interested in. The experiment was running for two days after which the subjects answered a survey about their experience with the system. The survey consisted of questions about how usable was the aggregator, if the snippets were well-arranged, whether the users were given sufficient information about the linked articles, if the users were recommended articles they were interested in and for statistical reasons the survey included personal questions asking about their gender, age, education and experience with web. This way we have been given a valuable feedback which will be later used to improve the system and get it ready for a large scale test.

The survey answers indicate relatively good usability of the system, its lucidity and also interest in using the news aggregator in the future. However, as the answers suggest, the recommendation system will need to be tweaked to provide users with better results. Also, there were no bigger differences showing imbalance of the chosen snippet types, all of them were rated as usable and easy to read.

The final goal of this project is to use the testing platform on a large scale audience and deduce the usability of the snippet types on a wide spectrum of users. Although the alpha test did not show any differences, we predict the test with more diverse subjects might yield more interesting results in a correlation with various personal characteristics. We expect that more experienced users will prefer the more condensed layout with less information, whilst recreational users will prefer the layout with large images and more information.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Anderson, C.: *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, New York City, (2006).

[2] Liu, J., Dolan, P., Pedersen, R. E.: Personalized news recommendation based on click behavior. In: *Proceedings of the 15th international conference on Intelligent user interfaces*, 2010, pp. 31-40.

[3] Kohavi, R. et al.: Controlled experiments on the web: survey and practical guide. In: *Data Mining and Knowledge Discovery*, 2009, pp. 140-181.

[4] Elly, D., Belkin, N. J.: Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 408-409.

[5] Li, L., et al.: SCENE: a scalable two-stage personalized news recommendation system. In: *SIGIR*, 2011, pp. 125-134.

[6] Joho, H.; Jose, J. M.: Effectiveness of additional representations for the search result presentation on the web. In: *Information processing & management*, 2008, 44.1, pp. 226-241.

[7] Dziadosz, S., Chandrasekar, R.: Do thumbnail previews help users make better relevance decisions about web search results?. In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002. pp. 365-366.

[8] Han, Y. G., et al. A new aggregation policy for RSS services. In: *Proceedings of the 2008 international workshop on Context enabled source and service selection, integration and adaptation*: organized with the 17th International World Wide Web Conference (WWW 2008). ACM, 2008. pp. 2.

Collaborative Enrichment of Learning Content

Martin SVRČEK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova2, 842 16 Bratislava, Slovakia
mato.svrcek@gmail.com

Abstract. Collaborative learning is a situation in which two or more people learn or attempt to learn something together. In the context of collaborative learning, web became a medium in which students ask for information, evaluate one another's ideas and monitor one another's work, regardless of their physical locations. Our aim is to enrich the learning content using a new type of annotations - definition (within the educational system ALEF). On the one hand, definitions can help students find the most important keywords and their explanation. On the other hand, we can enlarge conceptual metadata about the content, which can be used to improve services in the system (e.g. search, recommendations ...) by making Web page data in such a way that is understood by computers. In context of definitions we face problems such as synonyms or different explanation of one definition. Therefore, we also want to evaluate these definitions. Rating of definitions will enable us to show students the most accurate information. There are a lot of factors that can influence the rating of the definition (e.g. student reputation, number of similar explanations …). By solving these problems we can both help students and improve the information processing and presenting service in the educational system.

1 Introduction

Nowadays, we live in a world, where we are surrounded by plenty of information. For many people, the school is an essential source of information. In school facilities, we are surrounded by people with the same goals and hobbies. This fact allows us to collaborate with each other while learning. Situation, in which two or more people learn or attempt to learn something together, is called collaborative learning. We can differentiate between implicit and explicit collaboration [1].

Currently there are number of approaches to improve collaboration. Here we have to mention the Web. In the context of collaborative learning, the Web has become a medium in which students ask for information, evaluate one another's ideas and monitor one another's work, regardless of their physical locations.

Typical examples of the use of collaboration on the web are educational systems. There are several educational systems, such as ALEF [1], which allow the use of collaborative learning.

* Bachelor degree study programme in field: Informatics
Supervisor: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Within the educational system ALEF, our goal is to enrich learning content and support students in learning. For this purpose we want to use annotations.

When learning, students often encounter many new terms. And they want to know what these terms mean. This is why we decided to enrich learning content using a new type of annotations – *definitions*.

Definitions constitute an explanation of terms. This is the main focus of definitions because we want to support students in learning. Student can easily find unknown terms and understand them. In the context of the student learning support, our goals are:

1. Facilitation of student's acquisition of relevant and correct definitions of key terms being part of the course learned.
2. Convenient way of learning by reducing the number of actions necessary to search for term explanations.
3. Support navigation in the course through definitions of unknown terms in a document (utilizing the dedicated widget).

All these features will be discussed later in more detail.

In addition to student learning support, we can enlarge and enrich conceptual metadata about the educational content – e.g., relevant domain terms can be compared lexically by using terms' definitions. As a result, we can improve metadata-based services in the system such as recommendation. For example, by comparing explanations of definitions we might be able to detect synonyms as a part of automated domain modeling. A more detailed description of this aspect is out of scope of this paper.

The rest of the paper is structured as follows. In chapter 2 we mention several systems designed to support collaboration. In chapter 3 we will focus directly on the definitions. We will discuss two types of definitions, advantages of definitions and we will further describe the creation of definitions and access to the definitions in educational system ALEF. In chapter 4 we mention how we evaluate assumptions and suitability of our implementation. In chapter 5 we summarize the purpose and objective of our method and we will describe the future work.

2 Related Work

Collaboration has a lot of different interpretations arising from a number of areas of its application. In any case, collaborative achievement of objectives is very interesting and demonstrably beneficial approach [2]. Important is, that each team member has different views and opinions on the problem.

As mentioned, there are several systems supporting collaboration. ALEF [1] is a system with many features that support collaboration. First is tagging. Tagging allows you to enrich the content and improve the services provided by the system based on those tags. An important feature of tags is that tags in a certain way describe the document. They represent keywords and they are very similar to relevant domain terms (RDT) [3]. Therefore they are suitable for extracting RDTs. In addition to tags, external sources are present in the ALEF, allowing to add interesting sources of information. External sources as well as tags allow to enrich the content and improve the services.

There are also a number of other systems. SemKey is a system with support for tagging [4]. Tags offer more information and determine relationships between concepts and resources. These tags are used to enrich the content but also to enhance the search.

Yahoo! Answers is question and answering system with a rich base of different content. The authors of the [5] focused on the use of feedback to determine the quality of the content. Interestingly, punctuation and grammar affect the quality of the text. Hybrid recommender system is a system that uses tagging and concept maps as a basis for the recommendation [6]. Users can organize content by adding tags. Users also construct representation of concepts, concept maps. OATS is another collaborative tagging system [7]. Users can select text and assign to it the mark –

tag. OATS is a tool for efficient navigation and organization of learning content. CoWeb is a collaborative learning environment [8]. CoWeb is based on the Wiki and allows editing and creating pages with the learning content. Students can discuss their problems by adding comments to the learning content. COALE is collaborative and adaptive learning environment [9]. The system is aimed at dynamic learning organization of teaching through personalized recommendations. The users solves the task, discuss each other and the system (based on its activities) shows recommendations.

We have described several generic or educational collaborative systems. Although there still is a space for supporting collaboration during learning-specific tasks. We see the need of students to obtain information about the concepts during learning in educational system easier. Our goal is to provide students the definitions of these terms in one place without the need for search. And we want to take advantage of implicit collaboration (e.g., rating, voting) for filtering and selecting the most useful definitions for students.

3 Definitions: collaborative enrichment of educational content

We propose a new type of annotations within the educational system ALEF. This new type of annotations we refer to as *definitions*. The definition is an explanation of some term or concept. Students will be able to add definitions for certain terms. They themselves will form a list of important definitions for them. The proposed tool for adding annotations allows students to add definitions in two ways:

1. *ALEF definition* (AD) - annotation, where the source of information or explanation is learning content available in educational system ALEF.
2. *Own definition* (OD) - annotation, where the source of information or explanation is external source somewhere on the Web.

If a student interested in the contents of the document in ALEF, he can look at the tags (an existing feature in ALEF) as a form of keywords. In this case, however, it may happen that a student does not understand some terms. Such a scenario can be seen especially at the beginning of a course, when students just start to learn. In this case, the definitions are a great advantage. Definitions provide an explanation of the concept and, therefore, the student can better understand the intent of the document.

Another problem is the diversity of interpretations of terms. In the classic case, if the student does not understand some term, he is looking for a solution on the Web. On the Web, we can find a term that has several meanings in different areas. Therefore, it is very convenient to obtain information directly from the learning system, as the information there (provided by teachers) are usually relevant. For definitions added from external sources, it is important to evaluate the correctness of definitions (this can be done by student rating).This will allow students to determine the accuracy or inaccuracy of the definition.

Another advantage of definitions is the concentration of information in one place. If a student has all the necessary information in one place, he may not need to look for them on the Web. For this purpose we use the text highlighting and widget.

Now we will describe the definition annotations in terms of accessing and creating definitions.

3.1 Accessing a definition

In the learning process we meet many new terms that we need to understand. Because of this we need clearly visible explanations – definitions. In ALEF access to the annotations is provided by means of four basic elements [10]:

1. *In-text interaction and presentation* – provides the opportunity to work with annotations directly in the text (annotations are highlighted directly in the text).

2. *Sidebar* – provides the ability to view annotations directly next to the learning content.
3. *Annotation browsers* – Allows you to view all annotations of the same type in one place.
4. *Annotation filter* – allows students to choose the type of annotations that they want to have visible. For example we want to see only reported errors while fixing those [10].

We want to use these four basic elements for displaying and viewing definitions. Viewing a definition, however, is not the end of “the use case”. The most popular definitions will be mapped to relevant domain terms (RDT) in documents/learning objects they are assigned to, and become a part of a domain model (so called intentional descriptions of RDTs).

For example, if a student finds an unknown word while reading the text, he can simply look into the annotation browser and find here an explanation of the word. After understanding the term, he can continue reading the text. This use case shows how the definitions significantly improve educational system. Without definitions student would have to look for information on the web, which is significantly more complex scenario. The student also can filter out only the definition from educational content.

3.2 Definition creation

As mentioned, students will add definition in ALEF by assigning it to the learning content. Students will collaborate in working with definitions. On the one hand, they will add definition. And on the other hand, they will use them and rate them. It is very important to provide a simple method for adding definitions to the learning content.

As mentioned, we allow students to add definitions in two ways:

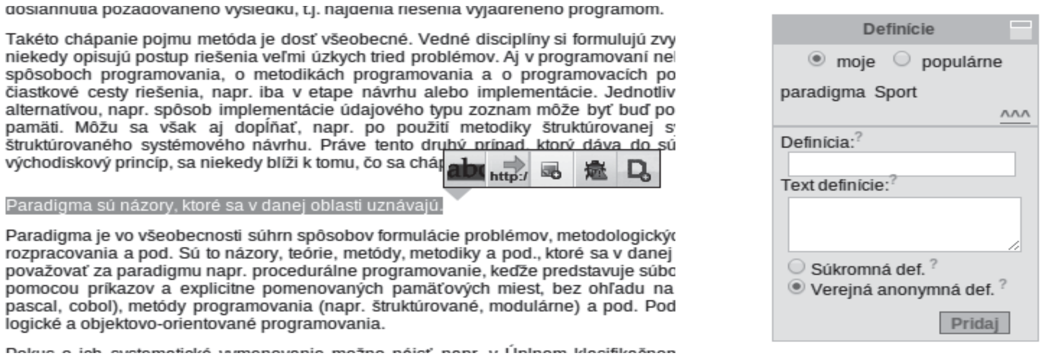
1. ALEF definitions (AD)
2. Own definitions (OD)

ALEF definition

The purpose of this use case is to encourage students to find term definitions within the educational system in ALEF. Marking text and assigning annotations to it arises shareable definition that can help all users. ALEF definitions are displayed directly in educational content. Therefore, the student must first select text in a document (see Figure 1).

Own definition

This feature enables the addition of definition from an external source, which will enrich the learning content. Own definitions are displayed in annotation browser or widget (see Figure 1). When students want to add their own definition, they must select the respective function in the widget.



dosiahnutia požadovaného výsledku, t.j. nájdenia riešenia vyjadreného programom.

Takéto chápanie pojmu metóda je dosť všeobecné. Vedné disciplíny si formulujú zvy niekedy opisujú postup riešenia veľmi úzkych tried problémov. Aj v programovaní nel spôsoboch programovania, o metodikách programovania a o programovacích po čiastkové cesty riešenia, napr. iba v etape návrhu alebo implementácie. Jednotliv alternatívou, napr. spôsob implementácie údajového typu zoznam môže byť buď po pamäti. Môžu sa však aj dopĺňať, napr. po použití metodiky štruktúrovanej s štruktúrovaného systémového návrhu. Práve tento druhý prípad, ktorý dáva do sú východiskový princíp, sa niekedy blíži k tomu, čo sa cháp

Paradigma sú názory, ktoré sa v danej oblasti uznávajú.

Paradigma je vo všeobecnosti súhrn spôsobov formulácie problémov, metodologický rozpracovania a pod. Sú to názory, teórie, metódy, metodiky a pod., ktoré sa v danej považovať za paradigmu napr. procedurálne programovanie, keďže predstavuje súbc pomocou príkazov a explicitne pomenovaných pamäťových miest, bez ohľadu na pascal, cobol), metódy programovania (napr. štruktúrované, modulárne) a pod. Pod logické a objektovo-orientované programovania.

Pokus o ich systematické vymenovanie možno nájsť napr. v Úplnom klasifikačnom

Definície

● moje ○ populárne

paradigma Sport

^^^

Definícia:?

Text definície:?

○ Súkromná def. ?
● Verejná anonymná def. ?

Pridaj

Figure 1. ALEF definition (on the left side) and Own definition (on the right side).

4 Evaluation

In order to evaluate our approach, we will perform several experiments in real world setting of educational system ALEF. Through evaluation, we want to verify assumptions and suitability of our implementation. Our aim is to collect definition from students during learning, analyze them and asses their quality to confirm our assumptions. If we find that the definitions are used and help students in learning, so we can say that we were successful.

The evaluation is planned to be performed on the course *Functional and Logic programming*. We will conduct an uncontrolled long-term experiment motivating the students to use our type of annotation. After a defined period of time, we will analyze and evaluate this data. Evaluation of the experiment will follow the abovementioned goals of our method.

First, we will determine whether the definitions represent the key concepts in the document and we will also verify whether students are able to identify the correct explanation to definition. Either by adding definition explanations themselves or by rating definition explanations provided by other students. The results of students' actions we will validate by comparing them with opinions of experts.

We also want to find out how many students use the definitions, based on 1) the number of added definitions and 2) the portion of the definitions added as ALEF definitions and the portion of the definitions added as OWN definition.

We also would like to get the students' views on the definition. We will collect these views through questionnaires. The fundamental question in the questionnaire will be how much the students liked the definition. Evaluating of the obtained responses will depend on this question. We will examine how many definitions were added by particular students and how particular students contribute to content enrichment using definition annotations.

In addition, we will also evaluate the contribution of student-added definitions to the domain model of the course in ALEF by employing domain expert judges and simple text-based similarity measures.

5 Conclusions and future work

At the beginning of our work we wanted to help students. We wanted to improve the work with the system ALEF, in the context of collaboration. We analyzed note-taking and adding annotations in the system ALEF but also in other systems. We also examined various forms of collaboration.

After that we realized that we need a new form of collaboration, a stronger tool:

1. Tool, which in turn will help to enrich the learning content.
2. Tool that will potentially improve the advanced functionality of the system.

To achieve such behavior we choose a new type of annotations – definitions. We are not aware of similar approach in the state-of-the-art approaches and we consider the potential of definitions for both learning and metadata enrichment to be worth researching.

We implemented definition annotations in the adaptive educational system ALEF and use them to facilitate learning in this system. Our goal is to support learning by making navigation in the educational course easier (since it is often difficult for students to find relevant information in educational system quickly). In addition to learning support, the definitions have a potential to obtain useful metadata for improving services in the system. For example, definitions' explanations can be compared to detect synonyms.

The important challenge for us is to evaluate the relevance of definitions. Without evaluation, we will not know which definitions are relevant and which are not. We plan to conduct an experiment assessing our assumptions about usefulness and usability of definition annotations.

We see great potential that our definitions bring into the educational system ALEF. They can greatly help students in order to understand the subject-matter. However, in the future, the definition can be used in many other areas that we mentioned above. We believe that we can get a lot of interesting findings and contribute to the improvement of collaboration and learning in general.

Acknowledgement: This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. 009STU-4/2014.

References

[1] Mária Bieliková, Marián Šimko, Michal Barla, Jozef Tvarožek, Martin Labaj, Róbert Móro, Ivan Srba, Jakub Ševcech. “ALEF: from Application to Platform for Adaptive Collaborative Learning.”

[2] Anuradha A. Gokhale. Collaborative Learning Enhances Critical Thinking. 1995.

[3] Jozef Harinek and Marián Šimko. "Improving term extraction by utilizing user annotations." *Proceedings of the 2013 ACM symposium on Document engineering*. ACM, 2013.

[4] Marchetti, Andrea, et al. "Semkey: A semantic collaborative tagging system." *Workshop on Tagging and Metadata for Social Information Organization at WWW*. Vol. 7. 2007.

[5] Agichtein, Eugene, et al. "Finding high-quality content in social media." *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 2008.

[6] Kardan, Ahmad A., Solmaz Abbaspour, and Fatemeh Hendijanifard. "A hybrid recommender system for e-learning environments based on concept maps and collaborative tagging." *Proceedings of the 4th International Conference on Virtual Learning ICVL*. 2009.

[7] Bateman, Scott, et al. "Oats: The open annotation and tagging system." *Proceedings of the Third Annual International Scientific Conference of the Learning Object Repository Research Network, Montreal*. 2006.

[8] Rick, Jochen, et al. "Collaborative learning at low cost: CoWeb use in English composition." *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*. International Society of the Learning Sciences, 2002.

[9] Furugori, Nobuko, et al. "COALE: collaborative and adaptive learning environment." *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*. International Society of the Learning Sciences, 2002.

[10] Šimko, Marián, et al. "Supporting collaborative web-based education via annotations." *World Conference on Educational Multimedia, Hypermedia and Telecommunications*. Vol. 2011. No. 1. 2011.

Activity Context-aware Personalized Search

Ľubomír VNENK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
lubomir.vnenk@zoho.com

Abstract. The Web has so much variable information, therefore searching a specific one is complicated. To find something valuable, specifying good query is crucial. However, average query consists of only about two words, which cannot specify intent of a searcher well. We suppose that these words follow in most cases the searcher's activity. We propose an approach for search query extension by activity context. The activity context contains words reflecting the searcher's activity context expressed by keywords gathered from one of the very recent activity provided by the searcher. We present a method for finding out searcher's activity context by logging content and his interaction between applications considering both standalone applications and applications running inside a browser. We have designed a method to use the logs to find a connection between the query and specific application. Then we extend the query by terms gathered by analyzing the selected application's content. We evaluate our approach in a series of experiments based on data gathered by monitoring small group of searchers by means of developed logger prototype.

1 Introduction

Searchers get used to finding searched information immediately. Search session is short and query consists of 2-3 keywords in average [5]. However, if they can't find required information at first page, they are not sure, what should they change to reach their goal. They get frustrated and start walking in cycles. If search session is successful and searcher has found what he was looking for, it usually doesn't take longer than 5 minutes, but if search session is not successful, it takes more than 10 minutes to give up searching.

There are many aspects that affect search success. The most common reason is ambiguous query. Just 2 keywords may have many meanings and search engine is not able to determine, which meaning is the right one, so its result list contains each meaning, or worse, the wrong meaning. We think that searcher search is the result of ambiguity rose in one of applications' content he has been working with a while ago. Therefore if we could get that content, we would be able to specify query and give it the same meaning the application's content have.

In this paper we propose a method for automatic activity context-aware query expansion. User's context is defined by his application's actual context. Application's context is processed

[*] Bachelor degree study programme in field: Informatics
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

application's contents to describe user's actual intentions and interesting parts inside the application. User's context is therefore mix of his current intentions and interests arising from his applications. The method consists of three steps: automatic logging of searcher's context, finding connection between query and an application and choosing proper keywords to extend query. We developed logging software to capture searcher's search activity context. We propose 3 strategies to find connection between searcher's query and an application. Each one has different weight and uses different methods to measure connection.

Suitable domain for the validation of methods for automatic activity context-aware query expansion is google domain itself with cooperation of the Firefox browser extension Annota [2]. Thanks to Annota we can modify search result page to verify our methods by comparing quality of our result list and original result list by implicit and explicit feedback.

This paper is structured as follows. We look at related works at section 2. In section 3 we propose methods for activity context-aware search and in section 4 we propose an experiment to evaluate our methods. We conclude in section 5.

2 Related work

There are many attempts on personalizing web search results with aim of achieving more accurate search results for individual searcher. Modern search engines try to personalize search results. There are also many other projects, that are using a popular search engine and personalize its results by their methods, to get even more precise results. There are many context-aware strategies, but we aim on activity based strategies. The most of them are evaluating click-through data to infer searcher's preferences. Click-through data are data that show where searcher clicked.

Based on the research of search engine searchers' browsing behaviors, Joachims et al. [4] developed a framework that utilized click-through data to infer searcher's preferences on documents and then learned to adapt the ranking function. Results indicates, that clicking decision is influenced by the relevance of the result, but they are biased by the order they are presented.

Leung and Lee [7] developed several profiling methods to get the searcher's positive and negative preferences. By considering each search query individually, the profiling methods capture the searcher's preference with fine granularity and shows better performance than those based on document preferences. They find out, best results give mix of positive and negative searcher preferences. Negative preferences also help to separate similar and dissimilar queries into clusters.

Analyzing AOL search query log Jiang et al. [3] developed method to capture searcher's preferences with high accuracy. They also propose query session segmentation method. It consists of choosing right time threshold and finding right type of change in consequent query. Based on type of change they use proper click-through evaluating method and they got even better results.

Different sources of contextual data for his model analyzed Kramár [6]. They consist of temporal context in form of behavioral search patterns, activity-based context in form of past queries and social context in form of searcher similarity. Each source of contextual data increases precision of context model. Search context model captures the lightweight semantics of the documents and evaluates implicit feedback to get document's relevance. They also propose a method that detects search session change based on lightweight semantics with higher precision.

The most of related works evaluated searcher's activity context only in browser's environment. It may not be enough because of searcher activity outside internet browser may have much higher influence. Our contribution to the field of personalization's studies is in evaluating searcher's activity context also in desktop applications.

3 Method for activity context-aware search

We proposed a method which extends query by application's context connected to query. To get application's context we capture searcher's interaction with computer and each application's content. Our method connects specific application and its content to a query and chooses right terms based on knowledge of application purpose and its content to extend the query. It consists of the following steps:

1. Capturing searcher's activity and application's content
2. Connecting an application and query
3. Extend query by proper keywords from application's content

3.1 Capturing searcher's activity and application's content

An application's content consists of keywords that represent application's content when application lost its focus. To get searcher's activity, we developed an activity logger. To log the most important sections of searcher's activity and to get the largest application's content the activity logger consists of three independent parts each monitoring specific activity:

1. Tabber
2. Wordik
3. Annota

Tabber is a desktop application that captures searcher's interaction between applications. It uses `EVENT_SYSTEM_FOREGROUND` hook to notify when application switch event occurs. To get name of application, searcher switched to, we use searcher32.dll library. This is the only content information capturing for every possible application. Tabber also catches each copy and paste event and text that is being copy.

Audio and video players are locking files that are being actually played. We use third's party utility called *handler* to get list of all files locked by a process. If the file is video or audio, we log its name and bind it to an application. These files become application's content.

Wordik is Microsoft Word's addin. We consider that searcher search many times information about something he is writing about using this application. To get content of any document, addin is a must. It captures names of topics he is actually writing and text that is around actually editing position. Addin can also hook foreground change event, so we can save document's content when searcher leaves Microsoft Word application.

The last part is Annota. It is Mozzila Firefox extension. It captures URL address and title of pages searcher is actually browsing. It also catches web-page change event and links, searcher has clicked, so we can get searcher's click-through. It consists of search results searchers has clicked, active time spent on clicked pages and interesting parts, like text which searcher has copied.

These three modules are for purpose of prototype. New specific modules can be add to capture activity from others applications. There were attempts to hook each application's GUI text, but they failed.

Each information captured when application lose focus is split into keywords and properly weighted to show how much is the keyword significant, so application's context is represented by keywords with highest weights. Weighting methods are further described below.

3.2 Connecting an application and query

Important task here is choosing which application is the right one, i.e. one which has direct influence on the search intent. We assume that this application is still opened and it was recently used. We has chosen time threshold to cut off inactive application and reduce possible ones.

To determine, which application is related to the query, we propose several strategies that consider various types of connections between application and query: syntactic distance, semantic distance and interaction between search session and an application.

Syntactic distance is used to get numerical measure of syntactic differences between two words. We compare the query and each application's content. If we can find query keywords in one of application's content, we can consider with high probability that we have found the right application.

Semantic distance is numerical measure of semantic distance between two words. It is measured by *Wordnet.API*. We compare query and each application's content. We can find out, if query and an application's content have similar meaning, or if there is a topic that query and an application's content have in common. We assume application with the lowest number is connected to the query. An application and query must be similar, so there is threshold to cut off application that may have the lowest number, but don't have similar meaning.

If there is no syntactic or semantic connection between keywords and application's content, we try to get proper application by evaluating logged interaction between search session and each application. Previous two strategies can evaluate relatedness of application to the search query before starting actual search. This one can be evaluated only if search session is in progress. The longer search session is the better results we gain. We aim on switching between an application and a search session and copying between them. If there is a copy-paste event, we can assume the application, searcher has copied to, is connected to the query. We assume that application, searcher use to switch from search very often, is also connected to the query.

If no strategy can determine with some minimal confidence which application is connected to a query or each strategy gives different results, we calculate final results as sum of weighted strategies.

3.3 Extend query by right keywords from application's content

Thanks to finding right connection between an application and a query, we gained larger context of what searcher may be trying to find. Therefore we can specify searcher's query and make it less ambiguous for a search engine. Choosing right keywords characterizing application's content is crucial. An application may have many contents and we need only few keywords, so we need to assign weight to each keyword. There are two types of weight: weight in general and weight related to a query

Weight in general reflects keyword relevance to the current content. If actual application content is a huge text, weight is calculated by third party service *Metallurgy*[1]. It is a text metadata extraction service and it can extract weighted keywords. Otherwise, if the content is just short text (often name), each keyword has the same weight. Weight also affects how often the keyword has occurred in the past. It is function of time, so keywords that were stored longer time ago have significantly lower weight. Keywords stored before threshold are not considered at all.

Not every application's content is connected to actual query. Therefore we need to specify weight related to query. It reflects how much is query connected to content. We use first and second strategy from previous step to get weight of each considerable content. Each keyword's weight in specific content is equal and inherits its content's weight.

Special type of keyword is keyword connecting an existing keyword and query in semantic tree. It inherits existing keyword's weight in general, therefore its total weight is higher than existing keyword.

We reorder keywords by total weight. Total weight is weight in general multiplied by weight related to query. Finally, we choose top 3-5 keywords with highest weights to extend query and 5 other keywords as recommendation. For example, if user is writing a biology essay about jaguar and he wants to search its average age, after searching "average age jaguar" query our methods find out the query is related to Microsoft Word, because jaguar can be found many times in the document he is actually writing and add "animal biology" keywords to his query because they have the highest total weight.

[1] http://metallurgy.fiit.stuba.sk - text metadata extraction service from URLs or text content

4 Evaluation

We hypothesise that searcher's search need comes in many cases from an application used recently. To prove this, we have proposed an experiment with aim to find connection between an application and a query and verifying the success rate of the connection. It is based on explicit and implicit searcher feedback. We modified general search engine (we selected Google search) results page and allowed the searcher to select right application that activated the search (if any) by clicking on its name from the list of possible applications (see Figure 1). They are ordered by our strategies for purpose of evaluating precision of our method. The application connected to query should occur in this list. We record highest precision if the application that searcher has selected is first in the list and we record negative precision if the application is not in the list.

As a searcher interacts between applications and search sessions, third strategy for finding connection between an application and query may significantly change weights. It may results in change of application considered as the best activity context for particular search. We also compare new result to searcher's choice to see, if this strategy increases precision.

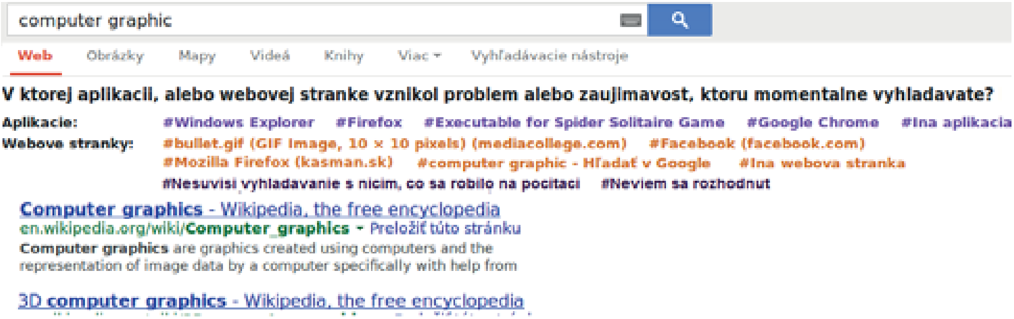
To evaluate precision of our method, i.e. evaluating whether selected keywords describe actual search context and result in better search results, we modified the search result's page to provide the most transparent environment. It highlights every change that has happened to the query. Moreover, it provides possibility to remove the keywords added by our methods or add another keyword, our methods have evaluated as suitable with ease.

This modifying ability also serves as explicit feedback. Searcher directly shows which keywords he doesn't like and which he does. If our methods add wrong keywords, searcher must react and remove them, otherwise he will get wrong results.

To verify if our methods are able to improve quality of search results, we propose an experiment based on implicit feedback. We compare search results of expanded query to not expanded results to verify, if search results of expanded query gives better results or not. To compare different search results, we show result list compound of results of expanded query and original results, similar to [6],[8]. However, behavior studies proves [1] that not only higher positions in search results rankings use to be clicked on, but also that lower positions are often not even examined at all. Therefore to achieve even fairer environment, each odd result belongs to owner that is selected at random and each even result belongs to its opponent. There is no visual difference between results and searcher has no chance to find out which search results are original.

To choose which search results are better we analyze searcher implicit feedback. We give each search result points based on interaction searcher has or has not made. Search engine, which search results brings more points is considered as better.

We give plus points to results, searcher has clicked to and minus points to results searcher has not. However, negative points can be only given to links, searcher has seen. We assume, that searcher has seen each link above last clicked result and in case that search session has not ended, searcher has also seen link just below last clicked result, like in [3].



computer graphic

V ktorej aplikacii, alebo webovej stranke vznikol problem alebo zaujimavost, ktoru momentalne vyhladavate?

Aplikacie: #Windows Explorer #Firefox #Executable for Spider Solitaire Game #Google Chrome #Ina aplikacia

Webove stranky: #bullet.gif (GIF Image, 10 × 10 pixels) (mediacollege.com) #Facebook (facebook.com) #Mozilla Firefox (kasman.sk) #computer graphic - Hľadať v Google #Ina webova stranka

#Nesuvisi vyhladavanie s nicim, co sa robilo na pocitaci #Neviem sa rozhodnut

Computer graphics - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Computer_graphics - Preložiť túto stránku
Computer graphics are graphics created using computers and the representation of image data by a computer specifically with help from

3D computer graphics - Wikipedia, the free encyclopedia

Figure 1. Selecting connection between an application and query.

Thanks to *Annota*, we are able to monitor searcher inside each search result page. We can detect if the search result helps him or if it was just waste of time. If it was just waste of time, searcher has left page without reading anything. We consider this search result as not clicked.

We can also detect, if searcher has copied or selected a piece of text. This means, that the web-page helps him and search result gets more points. Also reforming query based on a search result's text brings some points.

5 Conclusion and future work

We proposed a method for automatic activity context-aware query expansion. We proposed and implemented automatic activity context logger to capture activity context of each application. It consists of three independent logging applications and it uses third party keyword extractors to get content of an application. We also proposed three different strategies to find connection between query and one of applications' contexts. Each strategy return number that reflects how similar is query and an applications content. This numbers are weighted and the highest combination of weights is the application connected to query. These three strategies also help us to calculate weight of each application's context keyword. Weight of each keyword also depends on how much is the keyword relevant in application's context itself.

In further work we plan to evaluate success rate of connection finding strategies by implementing each one strategy and results comparing to searcher explicit feedback. We also plan to assign each weight to each application's keyword and extend query by few best.

Acknowledgement This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Granka, L. a., Joachims, T., Gay, G.: Eye-tracking analysis of user behavior in WWW search, In: *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04,* (2004), pp. 478

[2] Holub, M., Ševcech, J., Móro, R., Bieliková, M.: Annota: building linked data ser of documents and annotations, In: WIKT 2013 Proceedings : 8th Workshop on Intelligent and Knowledge oriented Technologies, Slovakia. Košice : CIT, (2013), pp. 13-18

[3] Jiang, D., Leung, K. W.-T., Ng, W.: Context-aware search personalization with concept preference, In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11,* (2011), pp. 56

[4] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search, In: *ACM Trans. Inf. Syst.*, vol. 25, no. 2, (2007), pp. 7

[5] Kamvar, M., Kellar, M., Patel, R., Xu, Y.: Computers and iphones and mobile phones, oh my!, In: *Proc. 18th Int. Conf. World wide we,* (2009), pp. 801

[6] Kramár, T.: Utilizing Lightweight Semantics for Search Context Acquisition in Personalized Search, In: *Information Sciences and Technologies Bulletin of the ACM Slovakia, 6,* (2014)

[7] Leung, K., Lee, D.: Deriving concept-based user profiles from search engine logs, In: *IEEE transactions of knowledge and data engineering journal*, (2010),

[8] Shen, X., Tan, B., Zhai, C.: Implicit user modeling for personalized search, In: Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05, (2005), pp. 824

Linking Slovak Entities from Educational Materials with English DBpedia

Ľuboš DEMOVIČ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
demovic@gmail.com

Abstract. Data published on the Web is largely unstructured, intended for people, without a clear definition of entities and relationships between them. In this paper we propose a method that allows automatic extraction of entities and facts from learning materials using search queries. Subsequently, we link all the acquired entities in the Slovak language with the English version of DBpedia. Our proposed method is almost completely language independent, which means, that we can link also entities in other languages to DBpedia. In the preliminary experiments we achieved the precision of linking of entities around 70%, which is a promising result continuation of our research.

1 Introduction

Currently, the Web provides a large amount of information, important facts and services to access them. It also has the potential to become the largest source of information and it is, therefore, desirable to be able to automatically gather information about entities and their relationships.

Data published on the Web is largely unstructured, without clear marking of entities and their relationships [1]. Linked Data describe a set of principles for publishing interlinked structured data. Links can be of various types and they create a graph representation of a selected domain.

There are a lot of specialized datasets that focus on entities from various domains, such as movies, books, popular music, geography, etc. In addition, there are also datasets describing the general principles of operation of the real world and the relationships between them. From the beginning, the CYC project was aimed to build a large knowledge base of general ontology [2].

Availability of vast amounts of interconnected data opens up great opportunities to create new generations of applications capable of using these Linked Data structures. With intelligent processing of the available data we can create useful methods aimed at: quick search, translation, personalization, recommendation, and context enrichment or user navigation.

In this paper, we propose a method that allows the identification and extraction of entities and facts in the Slovak language using search queries. Subsequently, we use the acquired facts for correct and automated linking with the English DBpedia dataset. As DBpedia is the core of the Linked Open Data cloud, such interconnection of entities enables us to use additional information

[*] Master degree study programme in field: Software Engineering
Supervisor: Michal Holub, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 137–142.

from various datasets. The results can be used for various tasks of personalized Web, e.g. for enriching the information presented to the user with additional facts.

2 Related work

Several researchers focus on processing unstructured data in the form of facts about entities from selected domains. From this work knowledge bases such as DBpedia, EntityCube, ReadTheWeb, YAGO-NAGA and other were created [3]. Their aim is to automatically build and maintain a comprehensive knowledge base of facts about named entities, their semantic classes, and their mutual relations with a requirement for high accuracy and relevance.

LinkedMDB project represents the first dataset that combines several popular web resources from the provision of comprehensive information about movies, i.e. IMDb, FreeBase, DBpedia Movies, RottenTomatoes.com, Stanford Movie Database, and more. The database provides millions of RDF triples with hundreds of thousands of RDF links to existing web resources. LinkedMDB implemented a way to create and maintain a large number of high quality links.

The mEducator – Linked Educational Resources provides structured data about various educational Web resources. An RDF schema is exploited together with clustering and enrichment techniques to progressively interlink educational resources with entities in available LOD datasets. A well-known LOD dataset suitable also for educational purposes is DBpedia. DBpedia extracts structured information from Wikipedia and makes this information more available on the Web.

The authors of the research in [4] applied entity extraction from unstructured documents to create links with the DBpedia dataset. They have proposed an end-to-end system that extracts RDF triples describing their properties and relationships with the text. Consequently, these entities are linked with DBpedia. All of the processed texts are in English.

In the research work [5], the authors describe an extraction of structured information from articles on Wikipedia. They have proposed a system called iPopulator that automatically processes info boxes for extracting information from Wikipedia articles.

The research paper [6] devoted its attention to generating Linked Data from unstructured text. The authors describe a method that combines deep semantic analysis with named entity recognition, word disambiguation and Semantic Web vocabularies. Extracted entities and relations between them are then linked with DBpedia and WordNet. Again, this method only works with the English language and does not utilize Linked Data to determine the ambiguity entities.

Based on the analysis, we conclude that nowadays there is no method that exploits the potential of Linked Data on the Web for determining ambiguity entities. This is also due to the fact that Linked Data are not yet as mature in the Web as other areas. It is a very young domain with a potential to be used in various Web adaptation tasks, such as context enrichment. Our proposed method also is almost completely language independent. We mainly focus on Slovak language, but we can simply apply out method for linking entities in other languages to DBpedia. These are our unique benefits compared to the state of art.

3 Entity extraction using search queries

While extracting entities and keywords from educational materials, we came across a problem of identifying complex entities, such as Slovak literary works “Keď báčik z Chochoľova umrie” by Martin Kukučín, or the first Slovak novel “René mládenca príhody a skúsenosti”. Obviously, it is necessary to have an automated way to obtain the following entities.

To solve this problem, we use the search queries that the users type when searching for suitable study materials. We also use a search engine that provides autocomplete of search results from Google and also ranks the results of search queries using the PageRank algorithm. Our hypothesis is that the users make extensive use of autocomplete, so the specified search terms are spelled correctly and without typos.

When obtaining search queries, we find the unique address of the website from which the user came to the current page. It is necessary to always verify that we came from a search engine on the domain with which we work, and not some other search engine that only has the same parameter in the web address. It is for this reason that our search engine uses sophisticated autocomplete, the great potential that we want to take full advantage of.

3.1 Algorithm for extracting entities

The algorithm for obtaining the search queries is as follows:

1. Check if the searched record exists to identify unique users with key search term
    a. If yes, check the existence of displayed study material among all recorded materials
        i. If such study material exists, ignore this search query.
        ii. Otherwise, add the ID of that study material into the array of all study materials.
2. Otherwise, create a unique record to identify the user with the ID of displayed study material as a single element of the array.

Table 1 contains examples of search queries with links to study materials that are stored in a database within the education portal. For search queries we store the following information: the date of the search query, the search query, the name and web address of a unique study material, category of the study material, order of clicks on study materials and the user ID. By this time we collected 66 180 search queries for more than 7 000 study materials.

Table 1. Example storing of search queries.

Search Query	Study material	Order
belgicko	Belgicko - konflikty v Európe	1
holandsko	Holandsko - konštitučná monarchia	1
antická literatúra	Antická literatúra	1
sto rokov samoty	Sto rokov samoty - Gabriel Márquez	1
rodina	Rodina – primárna, malá skupina	2
rodina	Rodina a jej funkcie	3

Through these search queries for each study material we determine the most popular search terms and these are then linked with DBpedia. We represent search queries in a graph scheme, where one type of vertices are search queries and the other type are unique identifiers to materials [7, 8].

4 Linking Slovak Entities with English DBpedia

For the purpose of linking Slovak entities, we use the Wikipedia API[1]. One of its services is the OpenSearch service[2] that returns the users' demand for the most popular Wikipedia articles. It is like the autocomplete for Wikipedia, thus for a specific term it returns several articles sorted by popularity. Next, we choose the best result from the offered choices. Now we have linked a Slovak entity with the Slovak Wikipedia. However, we need to link this entity with DBpedia.

We use another service from the Wikipedia API to link the Slovak Wikipedia article with all its available linguistic variations throughout Wikipedia. This is convenient, because most articles on Slovak Wikipedia have its English counterpart. At this point, when we have a Slovak entity link with the English Wikipedia, which means that we also have a link to the English DBpedia.

[1] WikiPedia API, http://sk.wikipedia.org/w/api.php

[2] Wikipedia API opensearch service, http://www.mediawiki.org/wiki/API:Opensearch

4.1 Algorithm for linking of entities

The algorithm for linking of entities in the Slovak language consists of five steps:

1. Search for entities via the Wikipedia OpenSearch API.
2. Find the English version of the entity for the selected result via Wikipedia API.
3. Disambiguation check of entities.
4. Choose the correct DBpedia URI
5. Link with the DBpedia dataset.

Figure 1 schematically shows how the algorithm works. We extract the entities using popular search queries from categorized study materials (e.g. rómeo a júlia, shakespeare, tragédia) and subsequently we link these entities with DBpedia (in our case: Romeo and Juliet, William Shakespeare, Tragedy). As a result, we get DBpedia's entity together with all of its connections.

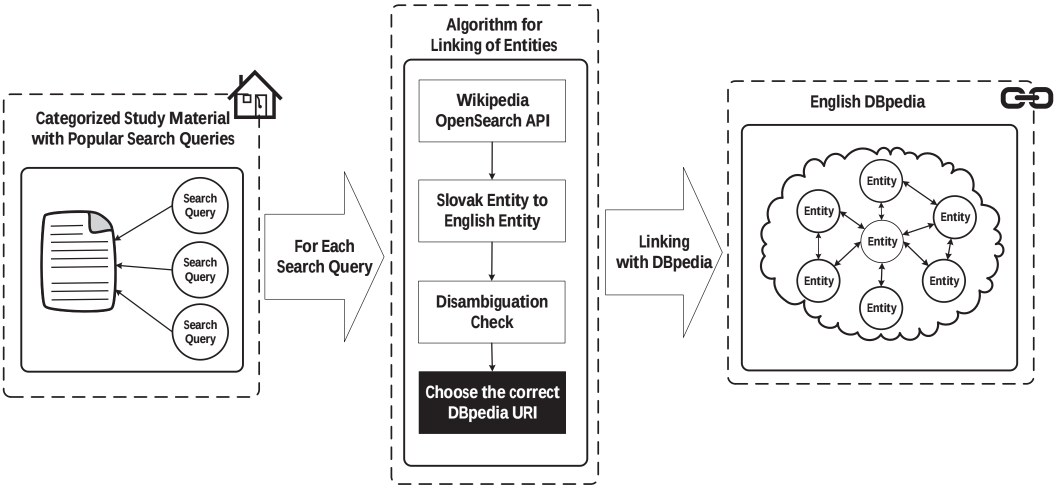


Figure 1. Diagram of the algorithm for linking of entities in Slovak language with English DBpedia.

4.2 Ambiguous entities and other problems related to entity linking

The algorithm of this method is related to the other challenges. One of the most significant challenges is the resolution of ambiguous entities. For example, we have a Slovak entity jaguar, and we cannot know if it is an animal or a car brand. To solve this problem we use categories.

We label each educational material with one or more categories (Biology, History, People, etc.) from this dataset. Given that each materials is already associated with some category in the educational portal, their connections with the categories in DBpedia are more or less automated. We categorize to specific categories manually. When we use categories, we assume that if we deal with the study material from category Biology, the entity Jaguar represents an animal.

Another solution for ambiguity of entities is to use properties and values of information from DBpedia. We observe the number of the properties and values found during the word-processing phase and on that basis we determine the clarity of the entity. Due to the fact that a number of properties of DBpedia are in the order of tens, then each property is manually translated with a suitable number of variations. For example, for the property populationTotal[3], we have added the version as: inhabitants, residents, population, people, and so on. This helps us to distinguish ambiguous entities, e.g. if we take the entity Martin, we do not know whether it is a person or a

[3] DBpedia, property populationTotal, http://dbpedia.org/ontology/populationTotal

city. Using the described procedure we find expressions such as: population, Mayor, Slovakia, Turiec river upon which we can certainly determine that it is a city.

Various language variations of Wikipedia do not contain the same articles. What if an English Wikipedia article is not in the Slovak Wikipedia? In this case we would have to translate the Slovak term to English, which is not easy for some more specific phrases.

5 Evaluation

Our main hypothesis is that we are able to automatically link Slovak entities with DBpedia with the accuracy of at least 85 %. We conducted experiments to determine how accurately we are able to link various types of entities. We have successfully linked about 80 % of Slovak cities to DBpedia. When linking entities without their disambiguation we achieved accuracy over 70 %.

In the next experiment we have tried to link all search queries regardless of the accuracy of results. As shown in Figure 1, the more popular search query has the greater precision of the linking search query (up to more than 50%).

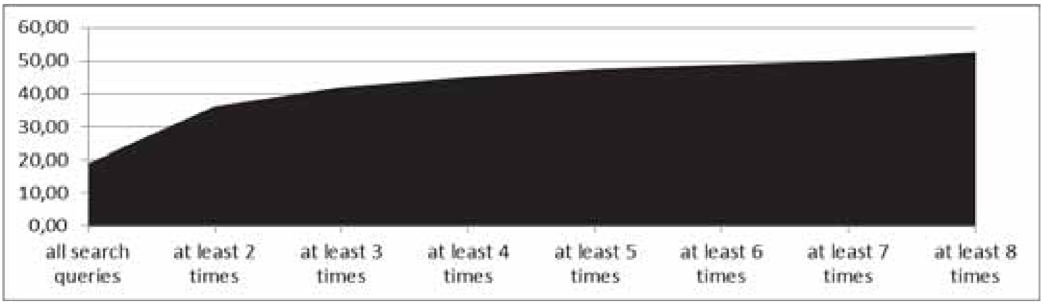


Figure 2. Precision of the linking search query based on count per unique search query [%].

We have more than 7 000 educational materials divided in more than 40 categories. We also focus on the problem of ambiguous entities. Within the method of extracting and linking entities we have prepared the next two experiments to verify these results. Some of the described methods are available online on the Web, so we can try them and compare our results with other methods.

5.1 The precision of automated linking Slovak entities with English DBpedia

The aim of this experiment is to verify the accuracy of linking entities or keyword in Slovak language with English DBpedia. The experiment will be performed on 100 selected users.

Every user in the experiment will see: 1) random study material from any category, 2) in average 3 extracted entities for each material that are linked to DBpedia, and 3) Wikipedia abstracts of the linked entities to correctly identify the connections between them.

Based on content of the study material, users will determine with explicit feedback, whether the connections were correct or incorrect. This means that the user reads the study material, then they read the abstracts of proposed links from Wikipedia and finally they click the correct / incorrect link. Based on this evaluation we will assess the precision of automated linking of entities with DBpedia. Thus, we count the number of all the assessments and the number of correct links and determine the resulting percentage accuracy.

5.2 The precision of the automated linking of popular search queries with DBpedia

The aim of this experiment is to verify the accuracy of linking popular search queries with the English DBpedia. From all acquired search queries, we choose the most popular queries, i.e. search queries that are the most frequently searched for.

Consequently, we will try to link the selected search queries with DBpedia and see how many of these search queries it will be possible link. Since not all requests are written in the appropriate format, it will be interesting to get these results. This part of the experiment can be accomplished without the need of user interaction. Subsequently, all successful linking of search queries will be checked on 100 selected users.

Based on the content of the study material, the users will determine with explicit feedback, whether the connections are correct or incorrect. This means that the user will click on a button with the correct or incorrect link on the relevance to the content entity.

6 Conclusions

The availability of large amounts of Linked Data opens up great opportunities to create new generations of applications capable of using these Linked Data. By intelligently extracting and linking the data on the Web, we can create a method that would be useful for recommendation or enrichment of information with an interesting content.

In this paper, we have proposed methods for extracting and linking entities and keywords in Slovak language with English version of DBpedia. Processing and linking of Slovak texts presents many challenges. We have proposed our own solution for linking entities in the Slovak language with English DBpedia. Slovak and Czech version of this dataset are currently not available, and our research work is one of the first initiatives in this direction. To confirm the functionality we have introduced new experiments to verify the method.

The proposed method appears to be promising and offers interesting results so far. Until now this area did not engage a lot of attention. This is also due to the fact that Linked Data are not yet in the Web as mature as other areas. Also for this reason, it is appropriate to focus on this problem and analyze unexplored possibilities.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] DeRose, Shen, Chen, Doan, Ramakrishnan. Building Structured Web Community Portals: A Top-down, Compositional, and Incremental Approach. *Proc. of the 33rd Int. Conf. on Very Large Data Bases.* Vienna, Austria: VLDB Endowment, 2007.

[2] Matuszek, Cabral, Witbrock, DeOliveira. An introduction to the syntax and content of Cyc. s.l. : AAAI Spring Symposium, 2006.

[3] Weikum, Theobald. From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. *Proc. of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.* Indianapolis, Indiana, USA: ACM Press, 2010.

[4] Exner, Nugues: Entity Extraction: From Unstructured Text to DBpedia RDF Triples. *The Web of Linked Entities Workshop.* Boston, USA: 2010.

[5] Lange, Böhm, Naumann: Extracting structured information from Wikipedia articles to populate infoboxes. *Proc. of the 19th ACM Int. Conf. on Information and Knowledge Management.* New York, NY, USA: ACM, 2010.

[6] Augenstein, Pado, Rudolph: LODifier: Generating Linked Data from Unstructured Text. *The Semantic Web: Research and Applications.* Springer Berlin Heidelberg, 2012.

[7] Berger, Doug Beeferman, Adam. Agglomerative Clustering of a Search Engine Query Log. New York, NY, USA: ACM, 2000.

[8] Hu, et. al. Mining query subtopics from search log data. New York, NY, USA: ACM, 2012.

Facilitating Learning on the Web

Martin GREGOR*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xgregorm@stuba.sk

Abstract. Web browsing is one of our everyday activities. Therefore web content enrichment with the potential to improve access to an information is an easy way to deliver new knowledge. There are several ways to enrich the web content. Personalization of web content enrichment is important for adaptation of a system to user individual needs. The problem is the inaccuracy of user modeling, causing inappropriate personalization resulting in inefficient web content enrichment. In this paper, we propose a method that will model the knowledge of the user about entities he encounters during browsing based on collected feedback on his behavior on the Web. The proposed method collects implicit feedback in a form of mouse clicks, moves, text selections, time spent over elements, time of visibility of each element and number of enters to areas of document. We predict a read level for whole document and deduce knowledge bound with terms of the document. We implement method in domain of language learning as a javascript library. We evaluate our approach as an extension for a web browser, where the user learns vocabulary of a foreign language.

1 Introduction

Today, we have many opportunities to learn something new. One of the most preferred ways of learning is Web, where we gain information through reading various types of articles. This type of learning is a part of life-long learning, which can be supported by computers and technology [8]. It is obvious that web browsing requires user interaction. User interaction can be observed and measured as feedback from a user. There are two types of the feedback that can be utilized for user modeling. The first type of the feedback, which does not require any additional effort from the user, is called implicit feedback. Typical implicit interactions on the Web are for example clicks, text selections, moves, scrolls, time spent on page. From implicit feedback we can deduce what the user reads and what he learns by reading.

We can deduce information from implicit feedback on two levels [3]. The lower level is an information deduction from one elemental interaction of the user. The higher and better level is the information deduction from a combination of some interactions which the user did. Implicit feedback is easily obtainable and trustworthy but it is sometimes really difficult to deduce an information about

[*] Master study programme in field: Software Engineering
Supervisor: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

the user [3]. The second type of feedback is explicit feedback which requires additional effort from the user, for example answering the questions, providing some ratings. An advantage of explicit feedback is that it can be used to check correctness of existing implicit feedback. On the other hand, we can consider untrustworthiness caused by wrongly motivated user among disadvantages.

The basis of every personalized educational systems is a user model [7], specifically the user knowledge model and collection of implicit and explicit feedback [1]. A system stores information about knowledge of the user in the model. Feedback collection and evaluation is one of approaches used to produce “data” for user modeling. The most important factor for quality personalization in a web-based educational system is the precision of the user model (e.g. a user knowledge model that is used for personalization of learning). Inaccuracy of the user knowledge model is a main problem of personalized web-based systems. Information in the user knowledge model is deduced from interaction data originating in implicit and explicit feedback. In this paper we proposed method which employs read level of terms in foreign language from implicit and explicit feedback and uses evaluations results to improve the precision of user knowledge modeling. We evaluate the method in the domain of foreign vocabulary learning, where considered feedback indicators are clicks, moves, text selections and time.

2 Related work

User knowledge modeling in e-learning systems is performed in the following steps [1] (1) data acquisition from user feedback; (2) data analysis, data processing and update of the user knowledge model; (3) e-learning system adaptation based on user knowledge stored in the user model. Data are obtained from implicit and explicit feedback from the user. The e-learning system chooses and processes the important data, updates user knowledge model and adapts own behavior according to knowledge stored in the user model.

User behavior is monitored and collected as data of implicit or explicit feedback. The analysis of the user behavior is important for understanding user needs and interests. User’s reading behavior study in [5] founds a new solution in a prediction of user gaze while reading without eye-tracker. A division of a document on the Web to k similar-height areas was a main idea of Hauger et all’s work. After the document preprocessing, the authors collect implicit feedback in a form of user’s clicks in areas, moves and text selections in areas, time spent with cursor in areas, time of visibility of areas and time of dwelling same y-position via cursor. The authors try to predict level of read for every area of the document. They propose 3 levels: area skipped, area glanced at, area read. In evaluation they discovered that moves and time spent in areas were the most significant implicit feedback indicator. Text selections and clicks were the least significant implicit feedback.

These results were used to propose an algorithm in [4]. Their solution predicts one of four levels of read of document areas. The authors proposed the algorithm predicting user read level of areas continuously via collecting and processing of implicit feedback modificators such as clicks, moves, text selections, time spent in areas, time of visibility of area. Each modificator has parameters as an internal weight, an external weight, time of activity of modificator. Their algorithm implemented as a javascript library, correctly predicts read level of document areas in 78,7% cases.

The read level prediction was also researched in [6]. The authors proposed a method enriching web document via translation of selected terms from a document to a foreign language with an aim to provide a user with “immersion” in the native context. The user was allowed to translate terms to native language via clicking the term or marking the translated term as an incorrect translation. The method was implemented as a web browser extension and collects implicit feedback from user behavior in form of clicks and time spent by document reading and explicit feedback from vocabulary tests. Data from feedback was used to model user knowledge. Text enhancement in the proposed method unobtrusively influences reading experience of each learning user.

We analyzed e-learning and user behavior monitoring on the Web and discovered the following problems: influence of imprecise user reading prediction on the precision of user knowledge modeling and resulting efficiency of (personalized) learning on the Web, and insufficient research on implicit feedback in the domain of term learning (e.g. when a learning wants to acquire new vocabulary). The aim of our work is two-fold: to improve state-of-the-art in reading prediction and to improve term-level knowledge modeling in a selected area of personalized web-based systems.

3 User knowledge modeling method

Our aim is to propose and evaluate a method of user knowledge modeling on the Web. In this work we focus on the area of term learning on the Web (this covers wide range of application, e.g. technical vocabulary acquisition, foreign language learning). Our solution extends existing model and method of user behavior monitoring [6] and collects and evaluates implicit feedback from generic implicit indicator inspired by the work of [4]. Our main idea is to monitor a number of mouse entries ("mouseenters") to area of interest of web a document. We use this number to predict what the user reads and what the user learns. In addition, we proposed a collection of specific implicit indicators for domain of term learning as well as a method for their processing into the user model. These two implicit indicators are translation of a term by the user and term exploration by the user. The proposed method consists of these steps (1) Document preprocessing; (2) Collection of implicit feedback; (3) Modeling of user knowledge.

3.1 Preprocessing

The first step is a division of the document into same-height areas of interest. The height of area is computed from the height of the document and the number of areas on the page. The height of the area has boundary minimum and maximum size. The number of areas depends on the height of document. Each area contains some tagged terms, usually visually enriched. The terms have educational meaning. Each term is marked with an attribute. The attribute contains value of user knowledge of the term.

3.2 Collection of implicit feedback

The second step is monitoring of user's interactions with the document while he is reading it. Our method collects data from interactions like clicks on terms, text selections of terms, a number of moves over the term, active time spent over the term, inactive time spent over the term, a number of entries to areas of the document and a translation of the term inserted by the user. Move is defined as a change of mouse pointer position over time. This interaction data are stored to so-called term model. The term model is a form of evidence layer [2] of user knowledge model. The term model contains data collected from implicit feedback related with the term. Data consists of following attributes in the Table 1.

3.3 Modeling user knowledge

The last step is an evaluation of learning process and updating of the user knowledge model accordingly. The user knowledge model represents the terms that the user had an opportunity to read in the document and their values of user knowledge in a range from zero to 100. Evaluation of learning process is based on rules described in Figure 1 below. The algorithm is performed for each enriched term from the document. The return value of the algorithm is the user knowledge K of the term. Numerical values α and β are specified from results of preliminary experiments. Firstly, the algorithm evaluates user entered translation of term. If the user entered translation of the term, that means he knows the term. The algorithm set knowledge of the term to a value α.

The next step is evaluation of a read level of the document areas that contain the term. A value of n is a division of a total number of the mouseenters and number of the mouseenters to the areas that contain the term. If n is greater than β, the algorithm increases knowledge of the term by the value of

Table 1. Term model attributes.

name	description
knowledge	a value of user knowledge of the term
position	a value of x and y position of the term in the document
area	a number of document area which contains the term
translation	entered translation of the term
entries	a number of entries to area of the term
clicks	a number of clicks performed on the term
moves	a number of position changes of mouse cursor performed in term element
selections	a number of text selections of the term
a_time	amount of a time spend with clicks, moves and selections performed on the term
v_time	an amount of a visibility time of the term
n_time	an amount of time spend without any activity performed on the term

Figure 1. Calculation of user knowledge of the terms.

```
if $translation \neq null \wedge K < \alpha$ then
    $K \leftarrow \alpha$
end if
$n \leftarrow N_i / N_a$
if $n > \beta$ then
    $K \leftarrow K + n * N_i$
else
    $K \leftarrow K + n$
end if
$T_E \leftarrow v_{time} * \frac{\sum_{i=1}^{N_M} w_{EXT_i} * T_{\%_i} * w_{T_i}}{\sum_{i=1}^{N_M} w_{EXT_i} * w_{INT_i}}$
$K \leftarrow K + T_E * \gamma$
return $K$
```

n multiplied with the number of mouseenters to the areas that contain the term. This multiplication expresses an increase of term knowledge based on the mouseenters to the areas.

The last important formula is a computation of T_E numerical value. The value of T_E represents a read level of the term. This formula is based on the algorithm from [4]. The result of T_E is computed from implicit feedback indicators defined in the Table 1 and performed on the terms. An explanation of the formula parameters: (1) time v_{time} of visibility of the term; (2) a percentage $T_{\%_i}$ of time occurrence of the indicator; (3) a weight of a time occurrence w_{T_i}; (4) an external weight w_{EXT_i} is determined statically for each indicator. The external weight denotes the relative significance of the indicator over others. The value of T_E is multiplied with γ number. A result of this multiplication is added to user knowledge of the term. A value of γ depends on a user reading speed in words per minute. The user reading speed is measured from time spent by reading and number of words in the document. The final values of the coefficients α,β and γ will be determined based on experiments we will conduct.

4 Evaluation

In order to evaluate our method we state the following hypothesis:

1. New proposed implicit feedback indicators (number of entries to area, a user-induced translation of term, term exploration) improve precision of user knowledge modelling.

2. Text read level prediction with number of entries to document areas improves user knowledge modeling.

Before proposing our method which extend model and collection of implicit feedback from the user we have made a preliminary experiment. This preliminary experiment helps us to discover new implicit feedback indicator we described above.

4.1 Preliminary experiment

Our preliminary experiment consists of six users where each user reads three different documents on the Web. The documents were divided to the same-height areas where each area contains information to learn. Reading process was monitored and implicit feedback indicators, such as clicks, moves, time spent in areas and entries to areas, were collected. Each user was asked question for each area after reading the article. An evaluation of reading behavior of each user shows high correlation 0,83 between the number of entries to areas and correct answers of areas. The click in the document was not frequent event. Moves and time spent in areas show low correlation (0.43 and 0.32) according to the correct answers. The preliminary experiment helps us to discover that the areas with the correct answers has a high number of entries. The high number of entries indicates that the user indeed read the area.

A result of the preliminary experiment is shown in the Figure 2. The graph shows two users and their entries to ten areas of articles over time. Dots represents entries to the areas. The Dots with circles represent the areas with the correct answers. We can see high number of entries to the areas with the correct answers. Axes represent time of enter to the area and the areas of the document.

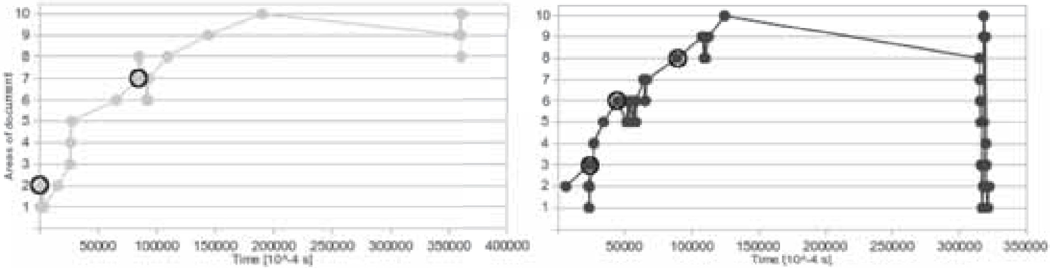


Figure 2. An example of measured mouseenters to the document area. Showing selected two participants.

4.2 Evaluation plan

We plan to evaluate our two hypothesis with browser extension of vocabulary learning proposed in [6]. Evaluation plan consists of two experiments. Our first open uncontrolled experiment will consist of 20 users. Each user has to read minimal 70 articles on the Web with a browser extension with our knowledge modeling method or without it. After reading of 10 articles each user will be examined with vocabulary test. We expect better results of vocabulary test with our knowledge modeling method as with original method. Better results proves that our method improves user knowledge modeling.

Second experiment will asses that prediction of user read level of document with our proposed entries to areas improve user knowledge modeling. Controlled closed experiment will consist of 13 participants with article about rules of game GO. This experiment is similar as [4] has done. One part of users will use extension of read level prediction with our implicit feedback indicator of mousenters to areas and second group without our proposed indicator. After reading each user will be examined with test of rules of game GO. The test will contain rules which user learned. Better results of the test of each group will reveal which knowledge modeling method is better.

5 Conclusion

In this paper we have presented our proposed method of user knowledge modeling for web-based systems aimed at term learning. In general, a drawback of current state-of-the-art approaches is inaccuracy of user modeling. This is often caused by the fact, that it is difficult to estimate whether a user really reads the presented content or not. This particularly applies if no special devices such as eye trackers are available and one can rely on mouse or keyboard only. Therefore, our goal was to improve user modeling based on implicit feedback originating from these devices only.

The main contribution of our approach is term-based user knowledge modeling based on a number of mousenters to an area of a document. The number of mousenters to document areas is a novel implicit feedback indicator which we have proposed as a result of our research including several small experiments. The preliminary results show that this indicator outperforms state-of-the-art indicators utilized for predicting user's reading behavior.

To date, we have implemented our method as a javascript library. In addition, we have already done a preliminary experiment to evaluate partial hypotheses of our approach. Our future work covers conducting a more complex experiment to evaluate the stated high-level hypotheses.

Acknowledgement: This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. 009STU-4/2014.

References

[1] Brusilovsky, P.: Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 1996, vol. 6, no. 2-3, pp. 87–129.

[2] Brusilovsky, P., Millán, E.: The Adaptive Web. Springer-Verlag, Berlin, Heidelberg, 2007, pp. 3–53.

[3] Claypool, M., Le, P., Wased, M., Brown, D.: Implicit interest indicators. In: *Proceedings of the 6th international conference on Intelligent user interfaces*. IUI '01, New York, NY, USA, ACM, 2001, pp. 33–40.

[4] Hauger, D., Paramythis, A., Weibelzahl, S.: Using Browser Interaction Data to Determine Page Reading Behavior. In: *User Modeling, Adaption and Personalization*. Volume 6787 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 147–158.

[5] Hauger, D., Van Velsen, L.: Analyzing Client-Side Interactions to Determine Reading Behavior. In: *UMAP 09:Adaptation and Personalization for Web 2.0. Trento, Italy (2009)*, 2009.

[6] Horváth, R., Simko, M.: Enriching the Web for Vocabulary Learning. In: *EC-TEL*, 2013, pp. 609–610.

[7] Tozman, R.: Learning in the Semantic Web. *eLearn*, 2012, vol. 2012, no. 3.

[8] Trilling, B., Fadel, C., for 21st Century Skills., P.: *21st century skills : learning for life in our times*. Jossey-Bass, San Francisco, 2009.

Using Linked Data for Exploratory Search in Digital Libraries

Michal CHYLIK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xchylik@is.stuba.sk

Abstract. The amount of data on the Web, which is published in structured way, is constantly increasing. Those data are published in accordance with the principles of the Semantic Web. One part of this idea is the Linked Data initiative. Data represented in this way allow more accurate answers to users' information needs. Due to the large number of data in particular problem area, the task to find a document that meets the needs of a user, is sometimes quite impossible. The goal this work is to design an algorithm that can choose the most relevant entities for the input entity and thus allow the user to perform exploratory search over a selected domain.

1 Introduction

Linked Data are an important part of the Semantic Web. They are publicly available structured data containing different types of entities and connections between them. Various methods of adaptation and personalization using the Linked Data have recently seen increased attention.

Marchionini distinguishes between three different strategies of searching [5]: lookup, learn and investigate. The last two strategies form exploratory search. Lookup is used when user wants to get exact information such as records in database or documents containing exact phrase or word. This strategy is still the one which is used mostly, but usually doesn't meet the real needs of users.

One of the most common problems for user who wants to gain knowledge on some topic is to exactly define his requirements and needs [5]. Oftentimes, this has to be done in keywords and terms which the user does not know, yet. This may be caused by his insufficient understanding of the problem area. The solution for this kind of problem is exploratory search [3, 4, 7, 9].

The domains composed of interconnected data that define relationships between entities are the best candidates for implementing exploratory search mechanisms. Using exploratory search we can extend the range of results by offering similar entities and navigate the user in the knowledge area. This allows him to explore a new domain, which is also the main goal of this paper. In particular, we aim at finding an algorithm which is able to select the most relevant entities for the starting entity. We perform our research and experiments in the domain of digital libraries.

[*] Master degree study programme in field: Software Engineering
Supervisor: Michal Holub, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 149–154.

This paper is organized as follows. Section 2 describes related work in the field of exploratory search with focus on Linked Data. Section 3 describes our proposed method and section 4 shows results of experiments on entities from a digital library. Section 5 concludes this paper with discussion of achieved results and proposals for future work.

2 Related work

So far, the area of exploratory search has been thoroughly researched. However, we are focused on exploratory search in domains, which are represented by Linked Data – a novel approach which has been proposed only in very few works.

Dimitrova and Lau [2] examine the role of semantic tags in the learning process. Presented case study with a semantic browser for the music domain indicates, that the semantic facets have positive influence on successfulness of exploratory search by allowing serendipitous learning.

Aemoo [6] is an exploratory search system showing filtered view of the DBpedia graph. The authors have tackled existing problems caused by preprocessing by designing a new real-time system. It solved the performance issues without the need of the preprocessing phase.

Another paper by Mirizzi and Azzurra introduces a tool named SWOC (Semantic Wonder Cloud), which allows users to perform exploratory search over DBpedia. SWOC contains two subsystems: the SWOC browser and a DBpedia ranking subsystem, which finds similarities between DBpedia nodes. SWOC enriches the graph with new associations calculated by combining formalised semantic knowledge represented in DBpedia with standard data from keyword search engines and social systems.

Project Yovisto [8] introduces a tool which allows semantic search in videos. Currently available video metadata are being mapped on Linked Data. Thus, the dataset is more appropriate for semantic search. They have introduced a navigational component that uses DBpedia connections between entities from the target domain. Because the amount of entities in DBpedia is too high to perform calculations in real time, a three-step algorithm is presented:

1. User request is mapped onto one or more DBpedia entities.
2. For every entity a defined number of connections with other entities is chosen.
3. The connected entities are mapped back to Yovisto repository in order to find related videos to the currently viewed one.

The possible next steps offered as exploratory search are calculated offline and stored in locally.
All existing research works have one feature which can be considered to be problem. They are tightly connected with a certain Linked Data dataset, usually DBpedia. They also prefer speed of calculation in favour of precision. Yovisto stops calculations of similar nodes once a certain amount of nodes meeting minimum coefficient of similarity is found. However, this can lead to overlooking other, more similar nodes, including of which could lead to increased precision.

3 Method for computation of the set of relevant entities

In order to accomplish our goal, which is to enable the user to explore the chosen domain represented using Linked Data, we need to obtain relevant entities to the starting one. Therefore, we propose a method, which finds a set of most relevant entities for a specified user who selects the starting entity. We aim our method to be domain independent.

The results of the algorithm contain entities of various types. The algorithm is composed of three main components:

- heuristics,
- weighting system,
- user model.

One of our contributions is the introduction of a weighting system which sets different importance to the heuristics which form the main algorithm. This helps us to get more accurate results. In particular, our proposed method works in these steps:

1. Get the starting entity as an input from a particular user.
2. Calculate values for connections using each of the designed heuristics.
3. Assign weights to the results of heuristics.
4. Order entities according to their coefficient gained by summing coefficient in heuristics.
5. Chooses top K entities and present them as the output. By default we select 15 entities.

The weighting system uses personalized weights for every user if possible. It is trained using the logs of visited entities by the user. If such data is not available, general weights are used.

3.1 Computing relevant entities using heuristics

The main idea is based on Yovisto project [27], but the heuristics are modified in several ways. We calculate the similarity of entities using 10 heuristics, all of them are domain independent:

- Frequency heuristic - It reflects the assumption that the more often property occurs for entity of a particular type, the more relevant for this type of entity is.
- Heuristic of same RDF type – it assumes that features connecting entities of same RDF type are more important and entities connected with these features are closely related.
- Heuristic of connection to the same place – it assumes that two entities connected to one place are more related.
- Heuristic of connection to the same event – the heuristic assumes that two entities connected to one event are more related.
- Heuristic of two-way connections – this heuristic assumes that two entities which have two way connection are more related.
- Heuristic of one-way outgoing connections – this heuristic uses outgoing connections. The important part of this heuristic is maximum depth for finding connections. This heuristic is derived from heuristic which originally involved only wikilinks.
- Heuristic of one-way ingoing connections – we use this heuristic to find all ingoing connections for entity. The depth of this heuristic is very important. It sets the maximum depth of searching for connected entities.
- Heuristic of lists – the heuristic uses lists of entities, which were created manually and configured for each domain separately. It helps to involve facts which are important but not reachable by direct connections in domain..
- Heuristic of categories – this heuristic uses categories of entities described in the domain. The feature which represents categories has to be configured manually for each domain of usage.
- Heuristic of ontology – this heuristic uses features defined by ontology of domain.. These features are more important than others.

Some of these heuristics have to be configured for domain in which the algorithm is used. The reasons are obvious. Definition of features representing, e.g. connection to place or connection to event can be different in each domain.

3.2 Determining the importance of heuristics using weighting system

Our algorithm uses weighting system to express different importance of the heuristics. We recommend using every heuristic, but it is also important to weight those using coefficients.

We have implemented a tool which can train these weights on the user defined training sets. This tool calculates the general weights for the domain, but also weights for all defined types of entities. In our experiments we have found out calculating of weights for types to be important.

Currently, we calculate the weights for each heuristic separately. The first step is to calculate the results of heuristic. Then, these results are compared with the training set and the recall is calculated, which is defined as the number of documents from the training set which can be found in the results of heuristics.

Our weighting system offers the use of personalized weights for users. These weights are calculated on the training sets, which were created based on the user model. If we collect a specific amount of data, which can be considered to be large enough to make some statements about user preferences, these data will be used as input for our training tool.

The usage of weights in our algorithm is then based on the user performing the exploratory search. If this user has personal weights, they will be used. If he does not, the weighting system will return general weights for the domain. They also depend on the type of the starting entity.

3.3 User model

We have designed user models, which will cover needs of our algorithm. System, which uses our algorithm for exploratory search should track movement of logged users and save it to populate this user model. It has to store three main values, which are important. These are unique identifier of user, source document and destination document of user exploring. These data will be used for calculation of personalized weights for users.

4 Evaluation

We performed experiments on a training dataset which was created manually. We used data from Annota[1], the service which improves the user experience while working with digital libraries. Annota [1] is a social bookmarking system enabling researchers to annotate, bookmark and share interesting research papers from libraries such as ACM[2]. Annota uses RDF for the representation of the data. We have randomly picked 32 entities as starting points for explorative search. These entities were chosen from 4 different RDF types, each type was represented eight times in the training set. For every starting point we have manually picked 15 most relevant entities.

We have divided the training set into two halves, one for training of the weights, other one for measuring the precision and recall. Every RDF type had the same amount of entities in both subsets. First, we used evenly distributed weights, which mean the same coefficient for every heuristic. Then, we used general weights calculated on the training set. As the last step we used special weights calculated for each RDF type. We have counted the top 20 results provided by our algorithm. The results of our experiment are presented in Table 1.

Table 1. Results of experiment.

Experiment	Precision	Recall
1 – Even weights	22%	27%
2- General weights	28%	32%
3 – RDF type weights	49%	72%

As we can see, the general weights do not improve precision and recall of algorithm significantly, compared to the use of evenly distributed weights. The main reason is that entities with different RDF types are very diverse in types of connections. Our domain for experiments does not have

[1] http://annota.fiit.stuba.sk
[2] http://dl.acm.org

defined any ontology, which would express this aspect. But once we have used weights calculated for each RDF type separately the precision and recall improve almost by 100%.

Let us describe the results in more detail. We have chosen an entity of type Person. Figure 1 shows top ten results for this entity returned by our algorithm. The set of results contains three papers which were written by this person. But the most relevant entity according to our algorithm is Mira Mezzini, researcher related to the starting entity in two ways: 1) they are members of same institution, and 2) they are co-authors of a paper. All of the given results are truly relevant for the starting entity and we cannot find more relevant entity described in our domain which was not part of our set of results. However, there are some entities which were omitted and their relevance is similar to some of the entities included. For example, B. Hoeltzener is also a co-author. His relevance coefficient is the same as for Jo Champeau and B. Baudry. This problem is caused by smaller number of connections between entities. The simplest solution is to show dynamic results, which means that the number of results will not be strictly defined.

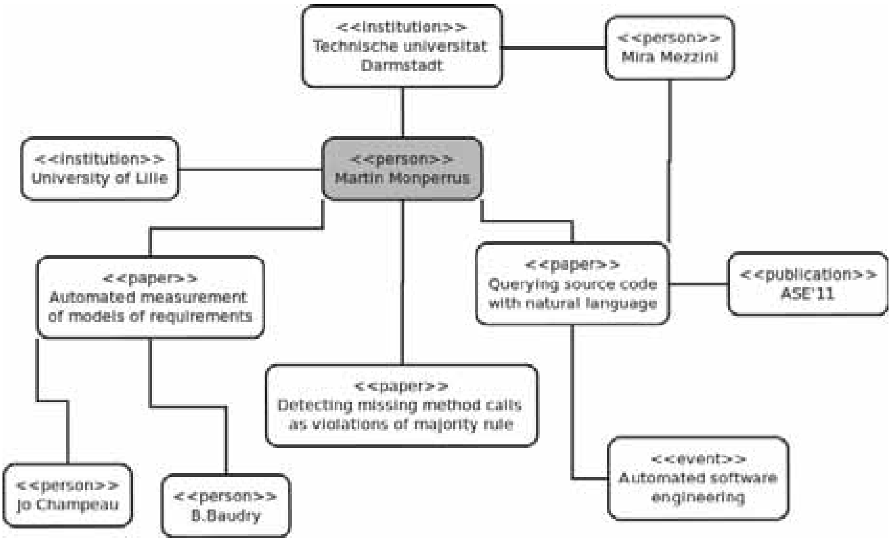

<<institution>>
Technische universitat Darmstadt
<<person>>
Mira Mezzini
<<institution>>
University of Lille
<<person>>
Martin Monperrus
<<paper>>
Automated measurement of models of requirements
<<paper>>
Querying source code with natural language
<<publication>>
ASE'11
<<paper>>
Detecting missing method calls as violations of majority rule
<<event>>
Automated software engineering
<<person>>
Jo Champeau
<<person>>
B.Baudry

Figure 1. Results of experiment for entity Martin Monperrus.

The performance of our algorithm is not that important in production mode, because we have designed preprocessing module to calculate results in advance. Real time calculations would cause many limitations, especially in depth of the following links in some of the heuristics. We tested our algorithm for depth up to 3. We have measured average time of execution for depths 2 and 3. We show the results in Table 2. However, it is sufficient to compute the results in advance.

Table 2. Time needed for preprocessing in different depths.

Depth	Time(s)
2	0.941
3	3.75

The experiment gave us really important knowledge about type of weights which should be used for our algorithm. It showed that in domains where the difference between entities of different RDF types in the types of their connections is big we should consider using of weights calculated for each RDF type. This experiment shows promising results, especially in the combination with weights based on the RDF type.

5 Conclusion

We have designed an algorithm for exploratory search in a domain represented by Linked Data. It helps users with exploring new area of knowledge by providing them with the most relevant entities for the entity which is currently displayed. This algorithm is domain independent unlike other existing works which are tightly connected to an existing public domain, usually DBpedia. We examine the contribution of different types of weights provided by weighting system designed in our paper. Weights can be evenly distributed, calculated for the whole domain, calculated according to RDF type of entities or personalized. To calculate personalized weights we have used user model described in our work.

We have examined usage of our algorithm on a dataset from a digital library. Besides this type of domain, the algorithm could be very useful in other types of domains typical for exploratory search, like domain with videos or pictures. Interesting usage could be implementation in online streaming audio service. However, the main prerequisites have to be met. The entities in such domain should have sufficient number of connections of various types.

As for the future work, we plan to perform more experiments to show real contribution of personalized weights. To examine this, we are planning to make experiments on user logs from system Annota and if the results will look promising we will make live experiments with real users to measure their satisfaction with sets of most relevant entities based on their personal weights. We are planning to improve the weighting system by implementing more sophisticated algorithm to calculate the weights of heuristics.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Bieliková, M., Ševcech J., Holub, M., Móro, R. Annota – poznámkovanie v prostredí digitálnych knižníc. *Datakon a znalosti 2013: Sborník konferencí – Ostrava.* Vysoká škola báňská – Technická univerzita, (2013). ISBN 978-80-248-3189-3, pp. 143-152.

[2] Dimitrova, V., Lau, L. Exploring Exploratory Search: User Study with Linked Semantic Data. *2nd Int. Workshop on Intelligent Exploration of Semantic Data*, (2013), pp. 0-7.

[3] Kammerer, Y., Nairn, R., Pirolli, P., and Chi, E.H. Signpost from the masses: learning effects in an exploratory social tag search browser. *In CHI '09,* ACM, (2009), pp. 625-634.

[4] Kules, B., Capra, R., Banta, M., Sierra, T. What Do Exploratory Searchers Look at in a Faceted Search Interface? *In proceedings of JCDL2009,* ACM, (2009).

[5] Marchionini G.: Exploratory search: from finding to understanding. Commun *ACM,* ACM, (2006), pp. 41-46.

[6] Musetti, A, Nuzzolese, G, Draicchio, F, Presutti, V, Blomqvist, E, Gangemi, A, Ciancarini, P.: Aemoo: Exploratory search based on knowledge patterns over the semantic web. *Semantic Web Challenge*, *ISWC2012*, ACM, (2012).

[7] Tvarožek, M. Factic : Personalized Exploratory Search in the Semantic Web, (2010), pp. 527-530.

[8] Waitelonis J, Sack H. Augmenting video search with Linked Open Data. *Proceedings of international conference on semantic systems*, (2009).

[9] White, R. W., Muresan, G., Marchionini, G. Report on ACM SIGIR 2006 workshop on evaluating exploratory search systems. SIGIR Forum, (2006), pp. 52-60.

Crowd-powered Evaluation Exercise

Marek LÁNI*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`mareklani@gmail.com`

Abstract. In past years, the Web began to be used largely for education purposes. There are many technology enhanced learning (TEL) or community question answering (CQA) portals, which are being used to gain knowledge and information. Therefore, the systems are beneficial to the users, however the users can be beneficial to the systems too. We can say it is a win-win relationship. The benefit for the systems comes from the content, which is often crowdsourced, i.e. generated by the users themselves. Since not everyone, who creates this content is expert in a specified area, it is necessary to filter it. The problem is that this filtering is a time consuming process and should be automated. In our work, we propose improvements to the existing exercise, which is a part of a TEL system. This exercise is used for evaluation of student-created answers to the questions. The evaluations are created by students and thus the exercise is built on a peer-grading principle, which can be used for filtering or determining the correctness of answers. We added new features to the exercise such as question answering or overview of evaluated answers, to enhance its educational potential. We discuss and qualitatively evaluate the exercise with students from the perspective of its benefits to the learning process. The outcome of this evaluation was that the exercise and its new features as part of learning process are regarded positively.

1 Introduction

Community question answering (CQA) systems such as Stackoverflow[1] or Answers.Yahoo[2] are becoming very popular. Thanks to these systems, it is possible to get an answer to almost any question. Nevertheless, it is important to filter and rate the answers, because the correctness of the user generated answers cannot be guaranteed. Filtering and rating are activities, which users can perform in the majority of CQA systems. These activities consist e.g. from assigning thumbs up/down or marking the answer as correct.

A basis of this CQA principle can also be used as an exercise in technology enhanced learning (TEL) systems. Teaching processes many times comprise scenarios, in which teachers ask questions and students (crowd) try to provide the correct answer. By this activity the students can clarify their understanding or gain new knowledge and the teacher can get an overview of the

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Jakub Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] http://stackoverflow.com/

[2] http://answers.yahoo.com/

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 155–160.

students' knowledge. The whole process can be also part of a TEL system as an exercise, in which it is possible to add additional features and enhance this process. Such a feature can be the evaluation of answer correctness. In case the students are involved in process of evaluation of answers, i.e. peer-grading principle is used, students can learn not only by answering, but also by evaluating. This principle of interactive exercise based on answer correctness evaluation was used in the related work on which our work is built. In our work, we took this basic principle and enhanced it with new features and developed a new interactive exercise, while we tried to provide the best possible contributions for improvement of the learning experience.

2 Related work

As we mentioned, our work is preceded by an experiment [4], which involved the usage of data collected in an interactive exercise, which is a part of an adaptive learning system called ALEF[3]. In this exercise the students evaluated the correctness of provided question-answer pairs on an interval from zero (totally wrong) to one (totally correct). This question-answer pairs were collected during tests in previous terms of a course, where the exercise is now being used. The exercise is now serving as a tool for learning for these tests. The acquired evaluations of answers were subsequently used for an experiment, in which the authors tried to interpret the global value of the evaluations of the answer. In this way, the authors wanted to find out, whether the crowd (students) is able to determine the correctness of answer similarly as a teacher (expert) and so if they can substitute the teacher in the specific cases of evaluation of answers.

Our work, unlike the preceding work, where the exercise served mainly as a tool for data collecting, is focused mainly on the quality of an exercise from the students' point of view. We wanted to improve the existing exercise, so it will provide students a better way to learn.

For our work it is very important that the peer-grading principle (or peer assessment) is helpful in educational process, as it is a base and large part of the exercise. Topping et al. [5] define peer assessment as "an arrangement for peers to consider the level, value, worth, quality or successfulness of the products or outcomes of learning of others of similar status". Peer assessment is considered to improve students' higher cognitive skills [1], because students use their knowledge and skills to interpret, analyze, evaluate and correct work of their peers. According to the work of Sadler and Good [2], peer-grading has different advantages, for instance:

Logistical: students can be grading simultaneously and they can take over the teacher's work, so it can save his/her time.

Pedagogical: when students evaluate and judge correctness of answers of their peers, it is a next possibility to obtain more information and knowledge about related topic. Many times they can also see another point of view on the problem, resulting from the question.

Metacognitive: grading and evaluating of answers as a part of educational process can push students beyond the provided study material in the sense of using an external materials and sources of information. In process of peer-grading students can also see their own strengths or gaps.

Sitthiworachart and Joy [3] further mention three important activities in peer assessment process, thanks to which, students are able to improve their knowledge:

Group discussion: students can exchange their ideas and knowledge.

Marking/evaluating: when students evaluate, they have to confront their ideas with ideas of their peers and have to evaluate the work of others, what improves their evaluation skills.

Providing feedback: students correct work of their peers and usually provide arguments and explanation of reasons of their evaluation.

Joy et al. also mention that peer-grading should be anonymous to avoid friendship grading or embarrassment for wrong or bad answers or evaluations. During the design of an exercise, based

[3] http://alef.fiit.stuba.sk

on peer-grading principle, it is important to provide all the mentioned activities and options, to create a tool, which the users will like to use and which will help them in learning process.

3 Design and implementation of a new version of the exercise

Question-answer pair evaluation exercise was originally part of the TEL system ALEF and its introduction to various subjects was difficult. According to this fact, we decided to create new version of the exercise as a standalone web application and to integrate it with the original system. We have defined a set of features, which we wanted the exercise to have, so we will be able to enlarge obtained dataset and to make it more interesting for students. These new features are:

Question answering - the most important new feature, which makes the exercise more interesting to use. By this, we have neared the exercise to CQA systems. The exercise is no more dependent on existing question answer pairs and the only necessity for usage of the exercise is a list of questions created by a teacher, so it becomes easy to use on any course.

Providing textual and graphical interpretation of evaluation - evaluation of the answers in the exercise is done by a slider. In the previous version the only interpretation of the evaluation was the position of the slider's handle. We added the next two interpretations: textual, which can take five different values and also graphical in form of coloring of the slider from red to green. By this we wanted to achieve higher degree of consistency of the user's evaluations, to make the data "cleaner".

Enhanced possibility to comment – in an interactive collaborative exercise, it is very important to provide users a possibility how to easily communicate. One way to do so, is by creating comments. In the previous version, users were allowed to create comments only after question answer pair evaluation and they were not able to look back at the comments or join the arisen discussion again. In the new version of the exercise we provide all these possibilities and so we strengthened its collaborative potential. The new version also provides the possibility to create an anonymous comment, so the author can express his opinions without a fear of embarrassment or influence on their reputation.

Displaying of evaluated answers - list of evaluated answers allows users to get back to the evaluated or created answers. This list provides a way, how to follow the process of the answer evaluation. On the list level, users are given information about the number of created evaluations, comments and textual and graphical (colored background) feedback about the average crowd evaluation. In the detail of the evaluated answer, users can additionally see distribution of all the created evaluations, their own evaluation and comments. They can also add new comments. Based on the feedback provided by users, we have also highlighted teacher's evaluation, if it was created.

Action and object selecting algorithm – selection of an action, which a user will do, is made automatically by an algorithm, which selects evaluation or answering according to the specified probability. During the selection of answers to evaluate, the greedy maximization principle is used as in the previous version of exercise. There are two levels of maximization. The first level is to achieve three evaluations, to provide quick and relatively relevant evaluation of answer. The second level is to achieve sixteen evaluations to provide data for interpretation of the evaluations experiment. We have enhanced the algorithm by rule, which selects only the questions, which the user has not seen in a specified period of time, to ensure variability of displayed content. User also cannot evaluate his own answer or answer one question more than once. Very important fact is that user can see the evaluations of an answer only after he created or evaluated the answer, so during the evaluation he is not influenced by the evaluations of other peers.

4 Exercise evaluation

The exercise was used for half of term on the course of Management of Information and Software Systems Projects. There were 238 questions created by a teacher and they were given to students based on content presented at lectures. Also, there were no initial answers in the exercise. The exercise was used by 96 out of 115 participants of the course and they together created 538 answers and 7719 evaluations.

We have evaluated the exercise qualitatively in terms of user experience and contribution to learning process. This evaluation was made in form of personal discussion with students, who have used the new version of the exercise. The interview scenarios were designed to answer our five main research questions: (1) How do students perceive the helpfulness of the exercise in learning process? (2) What was the motivation to use the exercise? (3) What were the most helpful parts of the exercise and how it was used? (4) How did students feel about their peers created content? (5) Did students tried to discuss and correct the wrong answers?

4.1 Participants and discussion

Discussion with students was made after the final exam from the Project Management in Software and Information Systems course. Discussion consisted of pre-formed questions related to the research questions and lasted 15 minutes on average. Discussion was made with five students, who have used the exercise. These student were chosen according to their level of activity and were distributed from very active to semi active users. We also conducted free discussion with two other students, who used the exercise only on minimal level, to get information what have discouraged them from using it.

4.2 Outcome

Outcome and answers to our research questions were created or derived from answers of students to the questions derived from the research ones. Next we will state these questions together with the collected answers. We also summarize the results from the research question point of view.

4.2.1 Helpfulness of exercise in learning process

When discovering if the exercise was helpful in learning process, we asked discussion participants four questions. First one was: *"What ways of learning for course have you used?"* Answer of all the five participants was that they used mainly slides from presentations and our exercise. Three of them stated they used exercise as a verification of their knowledge.

Second question was: *"Did the usage of the exercise help you on the final exam?"* All the participants answered that it helped them make an estimation of the range of questions on the exam, but as the questions on exam were of a different type (multi choice), there was no direct help.

Third question was as follows: *"Have you used external sources, when you evaluated an answer or answered a question?"* All the participants stated that they at least used the presentation slides. The usage of external web sites and the chatting with classmates were also mentioned.

Last question from this part was direct: *"Was the exercise helpful during the learning?"* The participants uniformly agreed that it was. One of the participants stated that well evaluated extensive answers were good source of information for him.

From these answers we can generally conclude that the exercise was helpful. During the answering and evaluating, the students have deepen their knowledge by studying from external sources, what is in our opinion the most important behavior of students related to the exercise.

4.2.2 Motivation to use the exercise

To get information about students' motivation to use the exercise, we asked them three questions.

The first question was: *"When have you used the exercise during the term?"* All the participants answered that they tried it, when it was introduced on one of the lectures and that they used it mainly during the preparation for the final exam.

The second question was: *"What were the reasons, why you used the exercise?"* Reasons stated were similar to the answers to the second question in the previous section. Students mentioned help with an estimation of the range of questions on the exam and that they used exercise as a verification of gained knowledge. Two students also stated that they were curious how their peers will evaluate the answers.

To the question: *"Is there any situation, besides before the final exam, when you would use exercise?"* the participants answered that before the midterm test and one if it was a homework.

These answers indicates that students are not using the exercise for a stackable learning or they simply do not learn stackable, but they use the exercise as a tool to learn for the exams.

4.2.3 The most helpful parts and ways of usage of the exercise

We also wanted the students to define, what were the most helpful features of the application and how did they use the exercise. To collect information about it, we asked them three questions.

First question: *"How did you use the exercise?"* Three participants answered that they were answering and evaluating for a longer period of time and after that, they went to the evaluated answers list to check the peers' evaluations. Three participants stated the usage of external sources in unsure cases and two stated that if they were not sure about the created evaluation, they immediately checked the detail and the evaluations of the answer in the evaluated answers list.

Second question was: *"Which part of the exercise would you identify as the most helpful for the learning process?"* In the answers, the participants specified different parts or features and there was no consensus of more than two participants. Two of them stated the highlight of teacher's evaluation, two of them stated the comments, by which it was able to correct the answer. Then one stated the list of evaluated answers.

Students also have possibility to skip the answering or evaluation, so we have asked participants: *"Have you skipped the answering or evaluating and if so why?"* Answers were similar and the reason was they did not want to create bad answers or evaluations, when they were not able to find information in external sources or when they did not cover the area of a question yet.

Although there was not specified a certain part of the exercise, which was the most helpful, every participant found his/her own part, which helped him the most. It is also very important that according to the participants, students were not creating junk evaluation or answers and they tried to provide as good entries as possible. From the answers also results that the list of evaluated answers is a good new feature, thanks to which students can strengthen their knowledge.

4.2.4 Trust in peers created content

During the usage of the exercise it is very important, how much students believe or trust their peers and how they behave in the act of evaluation of answers created by their peers. In this part we gave the participants four questions.

The first one was: *"Did you trust the average value of evaluations of the answer correctness?"* Two participants stated that they trusted it, but not completely. One participant stated that he did not trust it, because the evaluations were too positive. One participant stated that they were usually the same as hers, so she trusted them and one participant answered that she trusted them only when they were similar to hers.

The second question was: "*When you were evaluating, did you used the whole interval and gave also the extreme values or were you a reserved evaluator?*" Three participants said that if they were sure, they created extreme evaluations (totally wrong, totally correct) and two of them said they were keeping some reserve and used only a subinterval for evaluations.

The third question was: *"What did you do, when the average value of the evaluations was strongly different as yours? Was your trust to your peers lowered?"* Participants answered that in such a case, they tried to find right answer and find out who had the truth and that trust was not lowered. They also stated that if the peers were wrong, they added comments with right answer or a reason for their evaluation and that the comments were often source of helpful information. One participant stated that he always trusted himself more.

The fourth question was: *"How long did it take you, to create an evaluation on average"*. The answers to this question were uniform and the participants stated that if they were sure, they created quick evaluation in circa 15 seconds, but if they were not sure, they were thinking about it and were not creating evaluations according to their first feeling or random ones.

We can summarize results from these answers by a motto: *"Trust but verify"*. This fact is very good for the learning process, because the students think about the opinions of their peers, but they also verify them by studying external sources. This reserve was also reflected on the side of evaluation creation. Based on discussion we had with two students, who used the exercise only on a minimal level, we found out that the reason was they did not trust to content created by their peers and so did not want to use the exercise for learning. They said this problem should be solved by involving the teacher in the evaluation of each answer.

5 Conclusion

Based on the qualitative evaluation of the exercise we can conclude that the exercise was helpful in the learning process. Students were finding information in external sources during the usage of exercise, what is important for the learning process. Students also tried to create relevant answers and evaluations and appreciated the availability of the list of evaluated answers. A slightly discouraging factor to use the exercise may be the fact that the exercise is built on students created answers and evaluations and some students might not trust their peers. Regarding the future work, we have asked the participants after the main discussion to indicate possible improvements of the exercise. The opinion which resonated the most was that we should also add the evaluation to the comments in the form of thumbs up/down, because the comments are often the source of additional and correct information.

To sum up our work, we have designed and implemented a new version of interactive exercise based on some basic CQA principles. We have conducted the qualitative evaluation of this exercise, which confirmed its' benefits for the learning process.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Fallows, S., Chandramohan, B.: Multiple Approaches to Assessment: reflections on use of tutor, peer and self- assessment. In: *Teaching in Higher Education*, Vol. 6, Routledge, (2001), pp. 229-246.

[2] Sadler, P., Good, E.: The Impact of Self- and Peer-Grading on Student Learning. In: *Educational Assessment,* Vol. 11, Routledge, (2006), pp. 1–31.

[3] Sitthiworachart, J., Joy, M.: Effective Peer Assessment for Learning Computer Programming. In: *9th annual SIGCSE conf.e on Innovation and technology in computer science education*, ACM Press, (2004), pp. 122-126.

[4] Šimko, J., Šimko, M., Labaj, M., Bieliková, M.: Vzdelávacie objekty typu otázka-odpoveď : kolaboratívna validácia pomocou davu študentov. In: *Znalosti 2012 : Sborník příspěvků 11 ročník konference*, (2012), pp. 11-20.

[5] Topping, K., Smith, E., Swanson, I., Elliot, A.: Formative Peer Assessment of Academic Writing Between Postgraduate Students. In: *Assessment & Evaluation in Higher Education*, Vol. 25, Routledge, (2000), pp. 149-169

Collocation Extraction on the Web

Martin PLANK[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
plank.martin@gmail.com

Abstract. The main topic of this work is the extraction of collocations in the natural language. Natural language processing is important issue connected with metadata acquisition. The processing often involves identification of collocations in the text. There are various methods, which allows us to identify the collocations automatically. This work contains the analysis of existing methods for automatic collocation extraction. We characterize the issue of collocations in general and the properties of collocations, which can be utilized for the task of collocation extraction. One of these properties is limited modifiability of the collocation components. We propose a method for collocation extraction, which is based on this feature. Preliminary experiments show that performance of this method can be compared to other methods used in this area. Further evaluation and improvement of the method is the subject of our future work.

1 Introduction

Natural language is the main way of communication between the people. It is used for asking and answering questions, speaking about events etc. Web content is written is natural language, too. This causes problems for its intelligent automatic processing (e.g. machine translation or keyword extraction). One of well-known solutions is called The Semantic Web. It is based on an idea that web content should be expressed also in language, which machines can understand - in in the language of metadata. Creating metadata by the users of the Web is unfeasible. However, it is possible to use methods for automatic metadata acquisition. These methods involve natural language processing and handling its many specific features.

One of natural language features is that it cannot be simply reduced to vocabulary and syntax [8]. Individual words can be combined in various ways. Theses combinations of words are called collocations. Identifying collocations is very important in many applications of natural language processing, e.g. machine translation, keyword extraction, automatic text simplification, natural language generation, word sense disambiguation etc. Therefore, there are efforts to extract collocations automatically.

There are various definitions of the term collocation. Choueka [1] defines collocation as a syntactic and semantic unit, which exact meaning cannot be derived directly from the meaning of its

[*] Master study programme in field: Software Engineering
Supervisor: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

components. Other definition [4] says that collocation is a group of words, which occurs together more than often.

The most of the existing methods for collocation extraction is based on verifying features typical for collocations, described by Manning and Schütze [3] (see also [10]):

1. *Non-(or limited) compositionality.* The meaning of a collocation is not a straightforward composition of the meaning of its parts - e.g. *red tape*.

2. *Non-(or limited) substitutability.* The parts of a collocation cannot be substituted by semantically similar words - e.g. *guts* in collocation *to spill guts* (to confess) cannot be substituted by *intestines*.

3. *Non-(or limited) modifiability.* Many collocations cannot be supplemented by additional lexical material. For example, the noun in collocation *to kick the bucket* (to die) cannot be modified as *to kick the holey/plastic/water bucket*.

On the basis of mentioned features, collocations can be classified in four groups [11]. The classification describes the association strength between collocation components.

1. *Idiomatic collocations.* Idioms are fully non-compositional as its meaning cannot be predicted from the meaning of its components - e.g. *to climb a tree to catch a fish* (fruitless endeavour). The components of idioms are non-substituable and non-modifiable.

2. *Fixed collocations.* These collocations are non-substituable and non-modifiable. However, this type can be compositional - e.g. *diplomatic immunity*.

3. *Strong collocations.* This type has limited compositionality, substitutability and modifiability - e.g. *alliance formation*.

4. *Loose collocations.* These collocations have loose restrictions.

The first goal of the following study is to analyze existing methods for automatic extracting of collocations, as well as its characteristics, which are important for this task. The next goal is to design, implement and evaluate method for collocation extraction. Our work is divided into multiple parts. In the second section, we characterize related work in the topic of collocation extraction. In the third section we propose a method for collocation extraction. The fourth section is focused on evaluating of the proposed method. The last section contains conclusions and plans for future work.

2 Related work

The approaches for automatic collocation extraction can be divided into statistical and linguistic.

2.1 Statistical methods

The features of collocations can be mathematically described and then the strength of association between its components can be computed. These formulas are called association measures and they compute the probability that a word combination is a collocation. Association measures used for the task of collocation extraction includes frequency-based measures, hypothesis-based measures, and information theoretic-based measures. The overview of the most important measures is presented in [3] or [7]. Extensive comparison of these measures was done by Pecina [8]. He proposes an approach, where the best association measures are combined together to achieve the best results.

The list of best measures contains for example *pointwise mutual information* or *mutual expectation*. In the experiment with the combination of measures, the collocation were extracted with precision 0.85 and recall 0.90, or precision 0.90 and recall 0.56. In this approach, also some linguistic features, e.g. the part of speech or the context of bigrams is considered.

Another statistical method is called Latent Semantic Analysis. It allows extracting latent semantic relations from natural language texts [5]. It is based on a mathematical procedure of singular value decomposition of a matrix describing word frequencies. The result is a matrix of smaller rank, which reduces the information noise. The words that are similar (according to the matrix) can form collocations. But this is not the only assumption. Similar words are e.g. *car* and *automobile* do not form a collocation. To filter these words, only similar words, which distance in the sentence equals 1 are considered as collocations.

2.2 Linguistic methods

Linguistic methods are based on evaluating of typical characteristics of collocations on the basis of linguistic features. Pearce [6] presented, how limited substitutability can be evaluated. His approach uses WordNet[1], lexical database which is the source of synonyms. The synonyms, e.g. *strong* and *powerful* cannot be replaced in collocations, which are part of. Replacing the word *strong* in collocation *strong coffee* with *powerful* leads to senseless combination of words *powerful coffee*. Comparing the frequencies in the bigrams, where one word is replaced by its synonym can be used to compute the probability that a bigram is a collocation.

Other approach [10] verifies the limited modifiability of collocations. This approach is focused on typical combination of words in german language - the combination of preposition, noun and verb. These triples are annotated with frequencies and with their context called lexical supplement. The assumption is that a triple is less modifiable (and thus more likely to be a collocation) if it has a lexical supplement which, compared to all others, is particularly characteristic. In the experiment, this method was compared with some of the association measures and achieved better results. When considering only 1% of the most probably collocations, the precision was equal 0.84.

Another method [2] adapts the bilingual word alignment algorithm (used in the field of machine translation) into a monolingual scenario. This can be done, because collocations occur in similar context as well as bilingual pairs of words. With this approach, not only adjacent collocations are extracted, but also collocations, which are distant in the sentence. The precision of this method in the experiment was 0.62.

2.3 Summary

We described basic approaches used for automatic collocation extraction. According to the analysis, we identified four main problems in this area. First, only few methods are able to extract multi-word collocations. Second, there are many unexplored linguistic features of collocations that can be utilized for their extraction. Then, collocations are domain-specific [9]. Nevertheless, there is no suitable solution, which focuses on extracting of collocations in various domains. And at last, many methods are language-independent, but there is no service or tool for extracting of collocations in Slovak language.

3 Collocation extraction based on modifiability Of N-Grams

In this section, we propose a new method for collocation extraction. It is based on the assumption of non-(or limited) modifiability of collocations. The modifiability of a combination of words can

[1] http://wordnet.princeton.edu/

be computed according to the frequencies of n-grams. We design and evaluate this approach for Slovak language, so we utilize the frequency statistics of Slovak National Corpus[2]. The method can be explained on a simple example of collocation *to pull my leg* (to tell me something untrue):

1. The frequencies of collocation candidate and its components are computed. All frequencies are computed for lemmatized words. Also, the headword (important in the next steps) is identified (*leg*). Headword is one of the collocation components, which has a high semantic significance (e.g. noun or verb, not a preposition).

2. Frequent bigrams, which contain the headword are identified (*right leg, long leg*dots). We call the certain number of the most frequent bigrams *headword supplements*. Their frequencies are computed, too.

3. The candidate is modified by the headword supplements. The result is a list of *candidate modifications*: *to pull my right leg, to pull my long legdots*. Their frequencies are computed, too.

4. The frequencies of headword supplements and candidate modifications are compared and the modifiability of a collocation candidate is computed.

The process of judging the collocation candidate is based on the following hypothesis: If the frequencies of candidate modifications are significantly lower than the frequencies of the original candidate, its components and the headword supplements, the candidate is probably a collocation. In other words, if the candidate can be supplied by another lexical information, it has high modifiability. Otherwise, if it cannot be supplied, it has low modifiability and is probably a collocation. So our method looks at the modifiability in a different way, compared to approach described in [10].

3.1 The algorithm specification

To compute the modifiability of a collocation candidate, we compute a *modifiability score*. This score, in combination with other attributes is used to calculate the modifiability. The modifiability score S is computed as a sum of logarithms of candidate modifications frequencies $F(M)$ divided by frequencies of the corresponding headword supplements (from which the modification was created) $F(Sup)$:

$$S = \sum_{i=1}^{n} log \frac{F(M_i)}{F(Sup_i)} \qquad (1)$$

Then, the modifiability of collocation candidate $MOD(c)$ is computed as the modifiability score S multiplied by the product of *document frequencies*[3] of the candidate components $w_2 dots w_n$ (with the exception of a headword w_1) and divided by the number of not-null frequencies of a candidate modifications NF and by the candidate frequency $F(c)$:

$$MOD(c) = \frac{S \cdot \prod_{i=2}^{n} DF(w_i)}{NF \cdot F(c)} \qquad (2)$$

[2] http://korpus.juls.savba.sk/

[3] Document frequency DF of unigram w is the number of bigrams, which contains the word w.

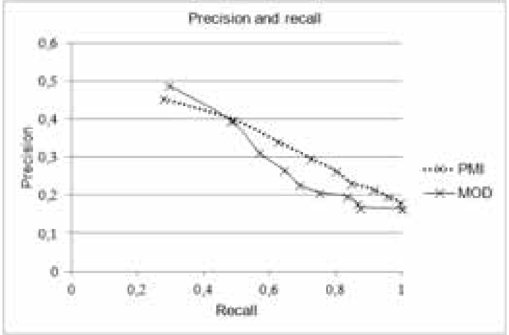


Figure 1. Precision and recall.

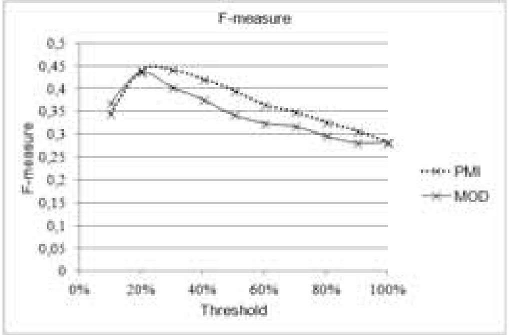


Figure 2. F-measure.

4 Experimental results

In the experiment, we compared our method with one of the most successful association measures - pointwise mutual information (PMI). It is computed as the logarithm of the collocation candidate frequency divided by the product of the candidate components frequencies. We specified a hypothesis: The performance of method for collocation extraction that considers the feature of limited modifiability of collocations is better than the performance of association measure pointwise mutual information. The performance of methods for collocation extraction can be measured according to the measures of precision, recall and their combination: F_1 - measure. The sets of true collocations (which are real collocations, judged by people) and collocations identified by the automatic method are compared.

As a dataset for the experiment, we created 4139 bigrams, which contains one of the 11 words, for which we know, what collocations they form from a limited dictionary of Slovak collocations created manually[4]. In the experiment, the modifiability based on our method, as well as the pointwise mutual information was computed for each of the 4139 collocation candidates. Then multiple thresholds were set to evaluate these methods. Firstly, we marked 10% of the candidates with the lowest modifiability (or the highest PMI) as collocations and then we increased this number multiple times by 10% to reach 100%. We set the number of headword supplements that are used in the calculation with similar experimental setup. The resulting number that we used in the main experiment was 55.

The comparison of our method and the state-of-the-art method pointwise mutual information is shown on Figures 1 and 2. The highest f-measure was 0.44 for PMI and 0.43 for the modifiability method. Our method achieved better precision and recall, when only 10% of candidates were considered as collocations. In conclusion, both methods achieved comparable results, but there are not enough evidences that confirm our hypothesis yet. Therefore, further evaluation is required. It is important to evaluate the extraction of multi-word collocations, because our method is able to extract them. The detailed evaluation includes also focusing on the collocation categories. Our method is related with modifiability, so it is more suitable for collocations with very limited or no modifiability - idiomatic and fixed collocations. The other types of collocations, strong and loose, can be partially modified.

[4] `http://www.vronk.net/wicol`

5 Conclusions and future work

In this article, we described several methods for automatic collocation extraction. We analyzed various open problems in this area and proposed a new method that can overcome some of them. Our method is linguistic, because it explores the feature of limited modifiability of collocations. However, it is combined with statistical approach - the modifiability is computed according to frequency statistics. We also evaluated our approach in a simple experiment, in which it achieved results that are comparable to another method used in this area - pointwise mutual information.

When compared to some other methods for collocation extraction, our method has several advantages. It is able to identify multi-word collocations. It is language independent and able to extract collocations of all types, though some types are probably identified more successfully. The method is likely to perform better on collocations with strong association - idiomatic and fixed (these collocations have very limited modifiability).

The goal of our future work is to improve the presented method and to supplement the knowledge in the area of collocation extraction, including the successful extraction of different types of collocations, as well as the domain-specific extraction. And the next goal is to develop a web service for collocation extraction in Slovak language.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Choueka, Y.: Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. In: *Proceedings of the RIAO*, 1988, pp. 609–624.

[2] Liu, Z., Wang, H., Wu, H., Li, S.: Two-Word Collocation Extraction Using Monolingual Word Alignment Method. *ACM Transactions on Intelligent Systems and Technology*, 2011, vol. 3, no. 1, pp. 16:1–16:29.

[3] Manning, C.D., Schütze, H.: *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999.

[4] McKeown, K.R., Radev, D.R.: Collocations. In: *A Handbook of Natural Language Processing*, Marcel Dekker, 2000, pp. 507–523.

[5] Nugumanova, A., Bessmertny, I.: Applying the Latent Semantic Analysis to the Issue of Automatic Extraction of Collocations from the Domain Texts. In: *Knowledge Engineering and the Semantic Web*. Volume 394., Springer Berlin Heidelberg, 2013, pp. 92–101.

[6] Pearce, D.: Synonymy in Collocation Extraction. In: *Proceedings of NAACL Workshop on Wordnet and Other Lexical Resources: Applications, Extensions and Customizations*, 2001, pp. 41–46.

[7] Pearce, D.: A Comparative Evaluation of Collocation Extraction Techniques. In: *Third International Conference on Language Resources and Evaluation*, 2002.

[8] Pecina, P.: An extensive empirical study of collocation extraction methods. In: *Proceedings of the ACL Student Research Workshop*. ACLstudent '05, Association for Computational Linguistics, 2005, pp. 13–18.

[9] Smadja, F.: Retrieving collocations from text: Xtract. *Computational Linguistics*, 1993, vol. 19, no. 1, pp. 143–177.

[10] Wermter, J., Hahn, U.: Collocation extraction based on modifiability statistics. In: *Proceedings of the 20th international conference on Computational Linguistics*. COLING '04, Association for Computational Linguistics, 2004.

[11] Xu, R., Lu, Q., Wong, K.F., Li, W.: Annotating Chinese collocations with multi information. In: *Proceedings of the Linguistic Annotation Workshop*. LAW '07, Association for Computational Linguistics, 2007, pp. 61–68.

Discovering Identity Links between Entities on the Semantic Web

Ondrej PROKSA[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
ondrej.proksa@gmail.com

Abstract. Linked Data are structured data containing entities and relationships among them that are available through the Web. The main problems are finding new relationships between entities and verification of the existing ones. In this paper we propose a novel method for discovering the relationship of identity (also known as owl:sameAs) between two entities in the LOD cloud. It is based on the analysis of sub-graphs formed by entities and existing relationships. Our method can be used for verification of existing relationships or to connect a new dataset in the LOD cloud. We present an experiment in which we evaluate our approach by discovering duplicate authors in the domain of digital libraries.

1 Introduction

In our times, few millions of specific webpages increase on the World Wide Web daily, from which Linked Data can be obtained. Linked Data appear on the Web in different form and goal of gathering structured data is a better possibility of device processing.

Linked Data are structured data containing real-world entities and relationships between them, which are published on the Web. They contain various types of relationships and form a graph of data describing selected domain. Linked Data produce a large graph, which links entities not only within a dataset, but also across datasets. Using the graphs gives a possibility to make more demanding and more complicated search queries, because they use the graph algorithms like the breadth-first search, the depth first search, etc. Thus, we are able formulate more complicated queries than using traditional approaches (e.g. keyword-based search engines). Apart from benefits, automated extraction of structured data introduces few problems, one of which is is linking identical entities across various datasets.

Currently, for the purpose of connecting entities between different datasets, the relationship of identity (also known as *owl:sameAs*) is widely used. However, it turns out, that in some cases the claim, that two entities are identical, is incorrect [5]. The problem might be caused when we have people who create the relationship with their namesakes or there are different entities that are connected with relationship of identity, but their properties are not fully identical. In the case, when

[*] Master study programme in field: Software Engineering
Supervisor: Michal Holub, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

two entities are identical, it is possible to create new relationships, which enrich Linked Open Data cloud with relationships across the datasets.

Our main goal is to discover relationships of identity between entities in order to create new connections between existing data sources and datasets in the Linked Data Cloud. Our goal is to propose a method, that will automatically search for connections and *owl:sameAs* relationships using graph algorithms and specific rules. We use similarities from sub-graphs of properties and classes for determination of identity between two entities. In the case that entities are identical, it is possible to enrich new relationships, which are across the datasets in sub-graphs.

Our proposed method is universal to any particular domain, because it uses sub-graph of properties and classes for entities comparison. Method is applicable on any particular graph, in which similarity relationships between the entities exist. It is possible to use the method in the search for identical organizations among the public government data as well as in discovering duplicate authors with data from digital libraries and also for connecting a new dataset to the cloud of Linked Open Data.

This paper is organized as follows: in Section 2 we provide related work to our research. In Section 3 we give a detailed description of our method which is based on the graphs' similarity, which produce entities from properties and classes. In Section 4 we describe the experiment discovering duplicate authors in the Annota dataset. The final Section contains discussion of implications of our research and a proposal of future work.

2 Related work

Linked Data enable us to interconnect entities across various datasets, thus creating one large global data space. The main goal of the Linked Data initiative is to have open data publicly available on the Web, which would be processable by software agents and robots. The secondary goal is to interlink various data sources.

In order to successfully use the Linked Data for the purposes mentioned, we need to have a good representation, which was analyzed in [6]. The main problem related to data representation is the discovery of similarity and identity relations between entities [1, 4, 9].

There are miscellaneous datasources in the Linked Data Cloud, which brings the problem of having duplicates of the same entity. One real-world entity can be represented by its various instances in multiple datasources. Every instance has a unique URI so it may appear that these instances belong to different objects. However, OWL standard defines useful relationship labelled *owl:sameAs*, which can be used for connecting multiple instances of the same real-world entity.

The problem is that this relationship does not always connect instances which are really identical [5]. Sometimes, it is used also to connect similar entities. Other times it is mistakenly used to connect entities, which appear to be identical, although they are not. This may be caused by errors, typos or noise in the source unstructured data from which the Linked Data are being extracted, lack of information about entities which would help us to distinguish them, or by namesakes in the source data (e.g. persons, organizations). These mistakes may lead to the same real-world entity being represented by multiple URIs in the same dataset, or to creating false *owl:sameAs* relationships between entities (from one or multiple datasets). In [5] the authors discovered 4 different cases in which the *owl:sameAs* relationship is used incorrectly. Sometimes, it is difficult to tell such entities apart and they are mistakenly linked as identical.

Because the Linked Data datasets use various ontologies to describe their content, there is a problem of ontology diversity, which could cause that identical entities are not connected using *owl:sameAs*. In [10] the authors propose a method for addressing this problem using graph algorithms. They defined the graph patterns which are subgraphs of two graphs with identical vertices and edges. First step of the method is to integrate two or more datasets and to detect the graph patterns in the resulting graph. Then, the method performs ontology alignment [7, 8] on each of the found graph patterns. Finally, it aggregates similar ontology classes and properties. The method was evaluated on

4 datasets form the Linked Open Data cloud and using this method the authors were able to discover new, missing relationships between the datasets.

When discovering new relationships of identity between entities, we need to take these aspects into account: the properties of the entities, existing relationships between entities and their distance in the graph, and the structure of the subgraphs the entities belong to.

3 Detecting similarity between entities

We propose a method for finding similarities between entities in a graph. We use this similarity to determine whether two entities are identical or not. This determines whether they should be linked using *owl:sameAs* relationship. We based our method on a hypothesis that the matching of entities is reflected in the similarity between the sub-graphs composed of classes and properties of the individual entities. This approach was also explored in the ontology matching problem [2].

The similarity between entities depends on the similarity of their properties, graph distance between entities and graph distance between neighboring entities. We define the total similarity as a sum of similarities of its individual components:

$$SGN = \frac{SNP \times W_{SNP} + ND \times W_{ND} + DRN \times W_{DRN}}{W_{SNP} + W_{ND} + W_{DRN}} \tag{1}$$

where
SGN - similarity of graph nodes, final similarity between two entities, SNP - similarity of properties between entities, ND - graph distance between entities, DRN - average graph distance between adjacent entities

The resulting values of the similarity are from the interval $[0, 1]$. $SGN = 1.0$ means that the two entities are 100% similar (i.e. identical), where as $SGN = 0.0$ means that the two entities are not similar at all (their similarity is 0%). All components of the Equation 1 have associated a weight (W_{SNP}, W_{DRN}, W_{ND}), which determines the component's contribution to the total similarity.

The weights may be adjusted for each dataset individually. As an example, in the domain of public government data it turns out that the entities' properties are more important than relationships between them, so the particular weights should reflect this. On the other hand, in digital libraries the most important are the connections between entities. In cases where we know the training set, the can train weights using machine learning eg. SVM.

SGN defines the similarity between the given entities a, b from the interval $SGN_{a,b} \in [0, 1]$. To determine whether these entities are identical and should be connected using *owl:sameAs* link, we need to find a threshold of similarity (denoted S) from the same interval. If the computed similarity of two entities is large enough (greater than S), we consider them to be identical, which implies that they can be connected with an identity relationship (or that the existing relationship is correct). Otherwise, we consider them to be distinct.

The threshold (S) should be determined for each domain differently. For its value we expect the following: $S \in [0.6, 0.8]$. This is based solely on our observations done on the data used for experiments. This value is slightly larger than the half and for various domains it may differ. We verified this claim in experiments described in the evaluation section. Next, we are going to describe each of the components of Equation 1 in more detail.

3.1 SNP - The similarity of properties between entities

If the entities are in the same dataset or have the characteristics of a dictionary from the same dictionary, so we always make it clear that we have to compare the properties. As is the case with the main similarity between entities (SGN), we also introduce the similarity between the properties (SNP), which also has its weight. The reason is that each feature may have different importance. The equation reflects the weighted average of the individual similarities between the properties. Sometimes it is possible that some properties will be ignored (we set their weight $W = 0$).

As we analyzed in ref we define the similarity of properties: Text similarity and Numerical similarity.

Text similarity between the properties define under ref as the arithmetic mean between the Levenshtein distance and 3-gram similarity as follows: Numerical similarity is defined as the arithmetic average between the text similarity properties and normalized numerical distance as follows: For other properties can be calculated by using the text similarity.

When computing similarity of the properties of two entities we use the fact that the entities are described using the same vocabulary, therefore the names of properties will match. The problem occurs when the properties of entities are defined using different vocabularies (e.g. the entities are from different datasets). In this case we compare the properties names' using: textual similarity and semantic similarity.

Once at least one of these similarities is above the given threshold, we can say that the names represent the same property. We compute the textual similarity as described previously. However, the semantic similarity between words is not easily computable as in general we do not know exact meaning of the words. In such case dictionaries or thesauruses might help.

3.2 Distance between the entities (ND)

When calculating the distance between two entities in a graph ($dist(A\ B)$) we use the breadth-first search. Our intention is to find the shortest path between the entities. We consider every edge to have the length of 1. The final distance is normalized to the interval $[0, 1]$. Normalization is based on the minimum and maximum distance in the given dataset. We define the distance between the entities (sig. ND) as follows:

$$ND(A,\ B) = [MAX_{dist} - dist(A,\ B)]\ /\ [MAX_{dist} - MIN_{dist}] \tag{2}$$

Maximum and minimum distances for the normalization must be set for each dataset separately. E.g. in the domain of public government data the minimum distance between organizations is at least 2, whereas in the domain of digital libraries the minimum and maximum distances between papers' authors are different (see evaluation).

3.3 Average distance between adjacent entities (DRN)

The average distance between two entities (similarity of which we are computing) is defined as the normalized average distance computed from the smallest distances to the neighboring entities. Given two entities A and B we compute their DRN using their direct neighbors.

Entity A has three neighboring entities - RN_{A1}, RN_{A2} and RN_{A3}. Entity B has four neighboring entities - RN_{B1}, RN_{B2}, RN_{B3} and RN_{B4}. For each of the neighboring entities of A we find exactly one neighboring entity of B which is the nearest to it. We get 3 pairs of entities. We then compute the shortest path for each pair of neighboring entities. The value of DRN between A and B is equal to the average of these shortest paths. Finally, we normalize the average distance using maximum and minimum average distances. Values of maximum and minimum have to be set separately for every dataset used. This is expressed in the Equation 3.

$$DRN(A,\ B) = [MAX_{drn} - avg\_dist(A,\ B)]\ /\ [MAX_{drn} - MIN_{drn}] \tag{3}$$

$$avg\_dist(A,\ B) = \sum \forall RN_{Bj} \in RN_B\ :\ min(RN_{Ai},\ RN_{Bj}) /\mid RN_A \mid \tag{4}$$

where
$DRN(A,\ B)$ - normalized average distance between A and B, $DRN(A\ B) \in [0, 1]$
$avg\_dist(A,\ B)$ - average distance between entities A and B

Table 1. Results of experiments no. 1.

Count of pairs	773210
Average	0.42313
Standard deviation	0.13381
Minimum	0.00000
Maximum	0.88571

$SGN \leq \mathbf{0.50}$	527194	68.18%
$SGN > \mathbf{0.50}$	246016	31.82%
$SGN > \mathbf{0.60}$	15211	1.97%
$SGN > \mathbf{0.70}$	1225	0.15%
$SGN > \mathbf{0.80}$	162	0.02%
$SGN > \mathbf{0.90}$	0	0.00%

Table 2. Results of experiments no. 2.

Count of pairs	773210
Average	0.45089
Standard deviation	0.08723
Minimum	0.17347
Maximum	0.84127

$SGN \leq \mathbf{0.50}$	560832	72.53%
$SGN > \mathbf{0.50}$	212378	27.47%
$SGN > \mathbf{0.60}$	15240	1.97%
$SGN > \mathbf{0.70}$	381	0.49%
$SGN > \mathbf{0.80}$	99	0.01%
$SGN > \mathbf{0.90}$	0	0.00%

4 Evaluation

In the experiments, we analyzed a dataset created in the Annota project [3], that contains entities from the domain of digital libraries in RDF form. Annota tool is an extension to the browser and helps to annotate, bookmark and share research articles and publications.

Our aim was to analyze the duplicate authors. For each author, we tried to find a few authors, which could be the same. We performed two experiments. In the first one, we tried to find random candidates. In the second experiment, we tried to find targeted candidates.

Dataset contained the following entities: authors, publications, articles, organizations, institutions and others. Dataset contains 154 642 authors, 26 170 articles, 6 organizations, 6 880 institutions and 7 019 other entities. In total, the graph has: 205 520 nodes, 884 380 properties and 336 525 relationships.

In the first experiment, for each author we found 5 random candidates and for each pair we calculated the value of SGN. Table 1 shows the results of this experiment. We have discovered the similarities between 773 210 pairs. The average value of SGN was 0.42 and the maximum value was 0.88. The results show that if we set the degree of similarity $S = 0.5$ (two entities are identical if SGN is least than 0.5) we are able to discover 31.82% of the identical pairs. If we set a degree of similarity $S = 0.6$, it would be only 1.97% of the same pairs.

In the second experiment, for each author, we found 5 nearest authors and for each pair, we calculated the value of SGN. In Table 2 we can see the results of the experiment no. 2. Again, we have discovered the similarities between 773 210 pairs. The average value of SGN was 0.45 and the maximum value was 0.84. From the result shows that if we set the degree of similarity $S = 0.5$ (two entities are same if SGN is least than 0.5) so we discover 27.47% same pairs. If we set a degree of $S = 0.6$, it would be only 1.97% same pairs.

5 Discussion and conclusion

In this paper we proposed a novel approach for discovering the *owl:sameAs* relationships between entities. We based our approach on the hypothesis that the similarity between entities depends on the similarity of their properties and relationships between them, which we proved to be right. This approach has the advantage that it is independent from the target domain, i.e. it can be used with any Linked Data dataset. The applications of our method are multiple: 1) it can be used to find duplicities in a dataset and thus help to clean the data, 2) it can be used to verify the existing identity relations in a dataset, and 3) it can be used to connect a newly created dataset to the Linked Data Cloud.

We have developed a prototype and evaluated it on a dataset from domain digital libraries. We analyzed the new dataset Annota, that was created using the principles of linked data. We tried to find a duplicities authors comparing our method. We have proposed two possibilities find candidates for same authors. The results of both experiments were approximately the same. Our method works stably, but it necessary compare with existing methods and try generate test data for evaluate.

In the future work we plan to evaluate our method by connecting our RDF dataset created from a digital library to the Linked Data Cloud using *owl:sameAs* relations. This area of research is very young and there are not many approaches with which we could compare our method. We will be evaluate not only the method as whole, but also the contribution of particular method parts (SNP, ND, DRN) separately in next experiment. Next, we will generate test data from existing dataset. We will create duplicate authors with changed attributes. Our method should discover duplicates.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Auer, S., Lehmann, J., Ngomo, A.C.N.: Introduction to Linked Data and Its Lifecycle on the Web. In: *Proceedings of the 7th International Conference on Reasoning Web: Semantic Technologies for the Web of Data*. RW'11, Berlin, Heidelberg, Springer-Verlag, 2011, pp. 1–75.

[2] Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and Ontology Matching with COMA++. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. SIGMOD '05, New York, NY, USA, ACM, 2005, pp. 906–908.

[3] Bieliková, M., Ševcech, J., Holub, M., Móro, M.: Annota - poznámkovanie dokumentov v prostredí digitálnych knižníc. *Proc. of the Annual Database Conf.*.

[4] Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web (LDOW2008). In: *Proceedings of the 17th international conference on World Wide Web*. WWW '08, New York, NY, USA, ACM, 2008, pp. 1265–1266.

[5] Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl: sameAs isn't the same: an analysis of identity in linked data. In: *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*. ISWC'10, Berlin, Heidelberg, Springer-Verlag, 2010, pp. 305–320.

[6] Harth, A., Hose, K., Schenkel, R.: Database Techniques for Linked Data Management. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. SIGMOD '12, New York, NY, USA, ACM, 2012, pp. 597–600.

[7] Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In: *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I*. ISWC'10, Berlin, Heidelberg, Springer-Verlag, 2010, pp. 402–417.

[8] Kang, D., Chen, H., Xu, B., Lu, J., Li, K., Chu, W.C.: Approximate Information Retrieval for Heterogeneity Ontologies. In: *Proceedings of the 2005 International Conference on Cyberworlds*. CW '05, Washington, DC, USA, IEEE Computer Society, 2005, pp. 539–544.

[9] Weikum, G., Theobald, M.: From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In: *Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS '10, New York, NY, USA, ACM, 2010, pp. 65–76.

[10] Zhao, L., Ichise, R.: Graph-based Ontology Analysis in the Linked Open Data. In: *Proceedings of the 8th International Conference on Semantic Systems*. I-SEMANTICS '12, New York, NY, USA, ACM, 2012, pp. 56–63.

Automatic Web Content Enrichment Using Parallel Web Browsing

Michal RAČKO[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xracko@stuba.sk

Abstract. In the domain of education, it is important to know relevant information about learning objects and relations among them. Adaptive learning systems extend the area of personalizing the learning needs of individual users. Assuming that the user behaviour is the same while following the same goal, e.g. searching for additional information to some topic, we propose a method for automatic web content enrichment based on user actions in the domain of open web. User actions are preprocessed, transformed to sessions and loops, and classified for further information extraction based on defined behaviour models. In this paper, we propose a method for implicit web content enrichment and relationship discovery based on user actions in the process of looking for information.

1 Introduction

Creating links between resources on the Web is now an acute problem due to the large amount of various content that is created or uploaded daily. In the past, content could only be created by the authors of pages, but now, in the times of Web 2.0, users can also add content themselves. This is the main cause of improper structure of such content and weak or no links between similar sources. Pages are no longer linked to other similar ones or the number of such interconnected sites is low.

Creation of clear and long-term sustainable structure of the Web is important for easier navigation when browsing or searching. Meta-information and semantics of each page provides the search engines with better picture of Web structure and thus allow more accurate searching. Based on the information about content, it is also possible that virtual links between similar resources exist without physical links. Therefore it is necessary to propose a method able to provide us with additional information about web resources and thus allow discovery of these links between sites.

Currently, there is a large number of web browsers allowing tabbed browsing. The most common ones are Internet Explorer, Chrome and Firefox. All of them in latest versions support this kind of browsing [1]. Some of them allow to persistently maintain the selected tabs, or renew their state

[*] Master study programme in field: Software Engineering
Supervisor: Martin Labaj, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

even after the user closes the application. Various researchers found that users use tabs in a large number of ways, while the majority of the usage ways was not explicitly planned [1].

The aim of this project is relationship discovery between sites frequently visited by users using multiple tabs. What are the relations between them, or whether they can be linked together based on the way the user has accessed them. These links may not be dependent on the physical hyperlinks between sites, but they are based on the user browsing behaviour. Our hypothesis is that while taking user actions into account in the process of browsing in tabs, we should be able to create connections between related websites.

2 Parallel browsing

We can think of the user browsing session as a sequence of actions performed during browsing [4]. Tabbing actions among browser tabs represent a problem, since each tab can be seen as a separate dimension and thus a page in one tab do not necessary has information about the pages in other tabs. The page itself does not receive information about the tabbing events among these dimensions. Therefore we have to propose a method that will flatten those dimensions into one sequence of user actions. Then, in our scenario of external sources discovery, we can identify sequences that led to adding an external resource to the adaptive learning system. When analysing the parallel browsing behaviour data, we need to consider different ways of switching among tabs. Users sometimes view the tab content for only a very short period of time, or they open the tab and do not look at it at all.

User actions with tabs could be following:

- opening a page in a new tab without switching into it,
- opening a page in a new tab while switching into it,
- opening a page in existing tab,
- closing active/inactive tab,
- switching between tabs.

Each of these actions is used irregularly. Research has shown [7] that re-visitation rate of tabs is much larger than the number of sites visited, thus the number of page loads is smaller than the number of tab switching actions. In some cases, the tabs created in the browser were not actually opened by the user. They could be ads that are opened in inactive windows, respectively tabs. Those records are difficult to distinguish from relevant ones [3].

Another important information when examining the user browsing behaviour is tracking how much time he/she spent browsing the content of the page [4] and thus we can reconstruct the actual viewing. Tab switching actions signal transitions between pages, even though the user has not followed a hyperlink from one page to another. We can also view the action of closing the active tab as an alternative to the *back* button in the browser.

3 User action logging and processing

For the purpose of parallel web browsing reconstruction, it is necessary to obtain sufficiently large amount of accurate user browsing data. First of all, we need to assign an unique identifier to individual users. Consequently, we need to identify each browser tab and the pages it contains. Through the use of a browser extension, we can observe important information, for example actions of switching between the existing tabs.

In general, there are two possible ways of recording user actions in the domain of open web. First option is to record the user actions on the server side [2]. In this case, it is not possible to capture user actions among multiple pages of different web domains, we can only track actions over web pages stored on the server where the tracking is deployed. The data are scattered among many servers. The second option is to create a browser extension to be installed by each user [6].

Such extension provides the ability to record browsing data otherwise invisible to the server. These may include, e.g., switching between tabs or windows, or page visibility status changes. Using an extension, it is also possible to observe additional information about the browsing behaviour such as whether the page has been displayed in a normal or private tab.

In our research, we record actions using an extension for Mozilla Firefox and Google Chrome browsers called *brUMo* [5], which was developed at the Faculty of Informatics and Information Technologies for the purposes of user modelling. This solution allows us to anonymously track user actions in the domain of the open Web. We subsequently use these logs for relationships discovery between similar pages.

4 Web content enrichment

In automatic web content enrichment, it is important to know what kind of actions has the user done while browsing the web. These actions may lead to creation of virtual links between pages that can be later transformed to hyperlinks. It is common in adaptive learning systems to allow the users to add external resources additional to learning objects present within the system. The usual scenario of adding an external source may start with opening a search engine and entering keywords corresponding to the information the user is looking for. Subsequently the user selects few suitable candidates from the list of search results by viewing their contents. The user works with the relevant web pages in a different way than with less relevant ones. These characteristics of user behaviour are crucial in detecting relationships between web resources according to their actions.

Current solutions allow websites to link each other based on content similarity, or the common characteristics of users who viewed them. For websites with large amount of text content, it is easier to use the content based methods, where keywords are selected and similar sites are found. In the second method, we consider the characteristics of the user and his/her fitness in the group of similar users. The proposed method takes user behaviour into account rather than the content and thus makes it possible to link related sites with little or no text content.

A proposed scenario of applying the method in an adaptive educational system is divided into two sub-scenarios. The first consists of users browsing through pages, while the second is processing the collected data with the proposed method. The first sub-scenario could be:

1. The user is studying a learning object in the system.
2. He/she needs additional information for better understanding of a problem mentioned within the object. The user opens a search engine in a new tab and using the keywords inferred from the learning object, he/she searches for resources that potentially explain the problem.
3. The user opens a few search results he/she deems relevant in the new tabs and reads through them.
4. After finding the necessary information, the user returns back to tab containing the system.

After collecting sufficient data, the next step of method realization is the following:

1. User actions are preprocessed into the form in which they are ready to be analysed and searched for behaviour patterns.
2. The method selects potential candidates for pages containing relevant information to the studied learning object. The best candidate is recommended for linking in the learning system.

4.1 Preprocessing

When browsing, in addition to ordinary actions (reading the content, switching between tabs) the users are also performing actions that have to be removed in preprocessing, such as actions of fast switching between tabs. Pages accessed via POST method should be considered carefully, since they

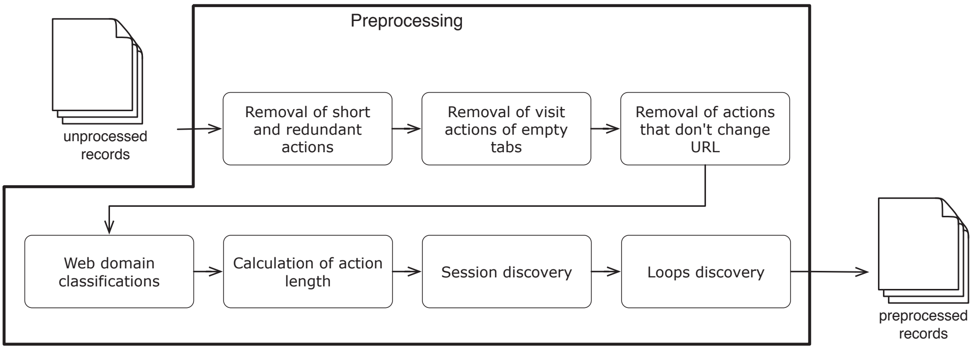

Preprocessing
unprocessed records
Removal of short and redundant actions
Removal of visit actions of empty tabs
Removal of actions that don't change URL
Web domain classifications
Calculation of action length
Session discovery
Loops discovery
preprocessed records

Figure 1. Preprocessing of web browser usage records.

have parameters hidden in requests and they can perform persistent actions. We classify pages into categories:

- adaptive system,
- digital library,
- search engine,
- other.

Another important information is how much time has the user spent on a specific site. This information is calculated based on the timestamps of two consecutive switch actions, which do not change the user context, such as opening a page in an inactive tab. Web browser usage data is a continuous record, so we have to be able to identify user sessions. The records may contain action with time distance even more than one day, but they are not considered as continuation of previous work and thus we mark them as a new session. Sessions can be determined based on the number of existing tabs, and when the number reaches zero, it means the end of session. The preprocessing is shown in Figure 1.

In the last phase of preprocessing, we divide the continuous record into sessions and these sessions into loops. A loop is defined as the smallest sequence of actions, which starts and ends in the same learning object. There must be at least one other page switch action between the initial and final action, otherwise we consider such loop incomplete and continue in expanding it. The loops which are not closed by the end of a session are deleted. The resulting loops contain potentially relevant external resources to the learning object that began the loop.

4.2 External resources extraction

After preprocessing, we have a group of actions without irrelevant ones. We then rate the significance of tabs with defined patterns. Weighting is based on multiple aspects:

- the category of the source and destination web page,
- the domain of the source and destination web page,
- the time spent visiting the page in a tab.

We adjust weighting of the web pages with a function defining the ratio between the active time spent on the site and its significance. We empirically determined that the category, assigned from 0 to 1, represents 70% of final page weight. Browsing time weight multiplier is logarithmically dependent on the time spent on page, where the upper limit is empirically defined to be 20 minutes. Final formula for weight calculation is $w = 0.7 * w_c + 0.3 * log(t)$, where w_c is category importance and t

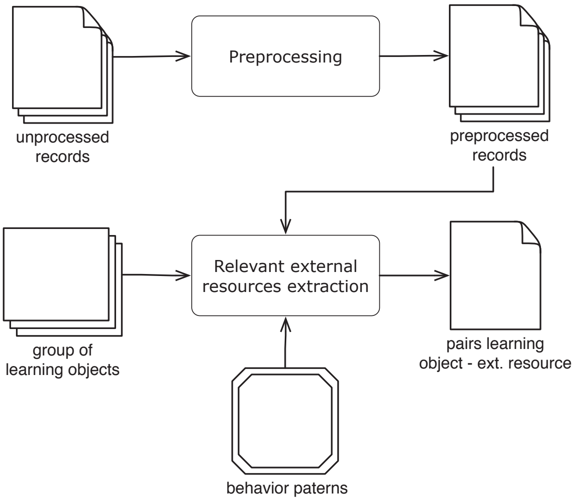

unprocessed records
Preprocessing
preprocessed records
group of learning objects
Relevant external resources extraction
pairs learning object - ext. resource
behavior paterns

Figure 2. Process of relevant resources extraction.

is browsing time in seconds. The method process is shown in Figure 2. Its output is a set of learning object–external resource pairs.

We use the resulting set of page pairs obtained after applying the method to pre-processed data to enrich learning objects in an adaptive educational system. External resources are automatically added to the learning objects and made available to users. The relationship between learning objects and external sources are important in the terms of content. We assume that the external source contains information similar to the learning object, thus concepts defined for the learning object within the domain model in the education system are describing the content of the resource.

5 Results and further experiments

We made several experiments to prove viability of the proposed method for identification of loops containing external resources potentially relevant to learning objects in ALEF system, using Principles of Software Engineering course. User actions data from brUMo browser extension were preprocessed in the aforementioned way and sessions and loops were identified. For the purpose of loop classification, each loop was assigned to the user action of physically adding an external resource to the learning object. The external resources were manually assigned into two groups by the domain expert - approved and not approved. Then we chose three binary classifiers and classified the loops into two groups:

- Loops with external resource that have been added to ALEF and have been approved,
- Loops with external resource that have been added to ALEF and have not been approved.

The purpose of this classification was to find a classifier with the best recall of the first class (approved), because these loops are going to be searched for relevant external resources and it is important to search only potentially relevant loops. In this experiment we have chosen the following classification methods: Naive Bayes, SVM and kNN. Results are shown in Table 1.

The best method for discovery of loops potentially containing external resources relevant to the learning object was Naive Bayes that has 93.24% recall rate for the chosen class. The trained classification method is used for further loop classification without information from the domain expert. Discovered external resources are then added to the appropriate learning object.

Table 1. Recall for class containing accepted external resources for evaluated classification methods.

Classification method	Recall
Naive Bayes	**0.9324**
SVM	0.8233
kNN	0.2414

We plan further experiments to prove method viability in web content enrichment. These are aimed at rating the external resources that are added to the learning objects in ALEF adaptive learning system. Students will be encouraged to comparatively rate external resources, both those found by the method and those added manually. We assume that resources found by our method will be at least as good as manually added ones.

6 Conclusion

In this paper, we proposed a method for automatic web content enrichment based on user behaviour in the domain of open Web and it is aimed to be domain independent. We also made several experiments based on which we were able to differentiate between interesting and non-interesting loops which were then divided into independent pages switch actions.

The method is implemented as a server side script and used in the Principles of Software Engineering course as an alternate way of adding external resources in ALEF adaptive learning system. In the future, we plan to improve the process of sessions and loops extraction and also measure performance of different weighting method settings.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0233-10.

References

[1] Dubroy, P., Balakrishnan, R.: A study of tabbed browsing among mozilla firefox users. *Proceedings of the 28th international conference on Human factors in computing systems CHI 10*, 2010, p. 673.

[2] Grace, L., Maheswari, V., Nagamalai, D.: Web Log Data Analysis and Mining. In Meghanathan, N., Kaushik, B., Nagamalai, D., eds.: *Advanced Computing*. Volume 133 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2011, pp. 459–469.

[3] Huang, J., Lin, T., White, R.W.: No search result left behind: branching behavior with browser tabs. In: *Proceedings of the fifth ACM international conference on Web search and data mining*. WSDM '12, New York, NY, USA, ACM, 2012, pp. 203–212.

[4] Labaj, M., Bieliková, M.: Modeling parallel web browsing behavior for web-based educational systems. In: *Emerging eLearning Technologies Applications (ICETA), 2012 IEEE 10th International Conference on*, 2012, pp. 229–234.

[5] Šajgalík, M.: Decentralizované modelovanie používateľa a personalizácia. Master thesis, Slovak technical university in Bratislava, Slovak republic, 2012.

[6] von der Weth, C., Hauswirth, M.: DOBBS: Towards a Comprehensive Dataset to Study the Browsing Behavior of Online Users. *CoRR*, 2013, vol. abs/1307.1542.

[7] Zhang, H., Zhao, S.: Measuring web page revisitation in tabbed browsing. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11, New York, NY, USA, ACM, 2011, pp. 1831–1834.

General Language Interface for Adaptable Semantic Search Engine

Marek ŠUREK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`marek_surek@yahoo.co.uk`

Abstract. Intelligent information retrieval becomes popular theme for business customers. The main challenge is extracting semantics and context of user request in natural language. Our research aim is devoted to embedding vector representation of words created by Word2Vec tool and use it for word annotation in search engine. We propose architecture which profits from properties of word vector model to create general interface for multilingual search engine. Therefore we trained and analysed multiple different vector models for Slovak language, which proof usability of word vectors for morphologically rich languages. We evaluate the models and propose their usefulness in search engine architecture.

1 Introduction

Nowadays companies invest lot of effort to intelligent information retrieval. Even though the interest raises rapidly, the current knowledge is still not sufficient. Many methods which are fully integrated into today's commercial systems like Apache Lucene, starts to run into limits. The problem is more obvious when companies try to deal with morphologically rich languages. The necessity to improve current search methods opens the space for new ideas and techniques which can improve quality of search results and enrich search methods as we know them today.

In our studies, we concentrate on developing adaptable semantic search engine. As the robustness of the work is high, in this paper, we deeply describe one particular part of search engine, which is responsible for improving accuracy and modularity of natural language query processor. We use results of word2vec [1, 2, 3] model to identify words and transform them into SPARQL query. The outstanding performance of the word2vec was largely tested on English language. As we origin from country which has rich language with many morphological constrain, we devoted our work to Slovak language. By many techniques we realized many interesting facts and areas where we can use word2vec in a new way it was not originally purposed.

Even though the stress of the work is detailed description of one specific part of search engine, the basic principles of our engine is introduced as well. In chapter 2, we briefly describe the most important features of the engine which creates architecture of the search system.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Miroslav Líška, Consultant: Dr. Marián Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

2 Related work

In scientific area, there is a numerous works, which tries to rewrite native natural language query into SPARQL query. We identify many areas, where we see a big potential for our contribution. One of the main disadvantages of current methods is their aim only to English language without any visible interface which can extend their work directly. Improvement in natural language processing is the key theme in this work. We want to present our novel approach of dealing with multilingual architecture problem. At the same time, the method solves lack of freely available dictionaries and WORDNETs, which are rarely available for less populated countries and their languages.

NAGA [4] represents language lock methods, which can be hardly adapted into different languages. This method is based on using URI ending as entity label which is typically in English language. Because of this mechanism, we can not simply extend NAGA model and add support for different languages. Except the fact, other problem is that authors of the project created their own query language.

The next implementation which has very similar characteristics with our search engine proposal is PANTO [5]. The only identified problem is that they also focus attention only to English. The methods implemented in to work are mainly design for English. OwlPath [6] is another representative of artificial language interface. The focus is concentrated on English language.

Single language methods restrict the possible usage of design search engine abroad. At the same time, artificial query language is another identified weakness of analysed works. The vector model should help us to deal with those problems on acceptable level of abstraction.

3 Adaptable search engine architecture

We build our search engine architecture on a few but very powerful technologies and search methods, which has big potential for great performance. The key feature which vastly forms the whole architecture and methods is decision of the database technology. We embrace native semantic repositories as they offer multiple features which are very important for adaptable search engine.

The key benefits are:

- Ontology which represents one of the most descriptive database scheme
- Reasoner, which helps us to infer new information
- Standardize identifiers for commonly used relationships

Adaptability of final design is very important for future reusability of our work. Except software modularity based on well design interfaces, we aim our attention on creating reusable workflow structure. Therefore we decided to integrate ontology processing directly to our search engine. The application is able to read ontologies from input and reconstruct the whole relationship graph. Gained graph is used for heuristic purposes during identifying entities in natural language user query.

Low latency of answer is one of the features, which is necessary for practical usage of our engine. A user reacts on the latency time and it can highly influence their final satisfaction with the engine. Appling numerous indexing and caching techniques help to minimize answering time and increase user satisfaction.

Many words have their meaning depended on the context in which they are used. Ideal example is word cheap. When we are talking about shoes the word cheap has different value range as when we are talking about cheap apartment. Therefore we design the software architecture in the way, that domain architect can easily set up which words influence the meaning words. The real meaning is based on statistics in the concepts. As we declared, our search engine is tested on relatively large data 20+ millions of triples we identify necessity to precompute the statistics before their usage.

Important part of any search engine is identifying named entity and ontology relationships in user query. Here we use word2vec as synonymic and morphological helper. Created model should be able to provide appropriate synonymic and semantic alternatives. At the same time, we use Lucene index to process faster named entity lookup. Another benefit of word2vec usage is ability to create licence free dictionary. Official dictionary are usually connected to copyrights. We see potential in analogy modelling which were shown in previous work related to word2vec [2, 3].

4 Word2Vec

The word2vec tool is a new approach in natural language processing. It is important to understand computational model to understand possible use case scenarios. Researchers can see the possibility and usability of the trained model in their scientific work.

The basic idea of word2vec model is vector in multidimensional space. All words are precisely represented in the space and have well defined relationship with other words. The actual weight of similarity is counted with cosine similarity measure. The idea of vector representation of word is not a new one. The real benefit, except the outstanding performance, is training time. Ability to train models in couple of hours instead of weeks, which are very usual for neural networks, underlined the strength of this newly developed model.

The whole method is a set of complex optimization method. In our work we train vector model by skip gram model. Skip gram models works on principle of skipping words in sentence. The number of skips is called windows. Using this technique, we can deal with sparse of word co-occurrence. It is very important for neural network language models, which is directly depended on word co-occurrence in sentence to estimate appropriate relationship weight. The model can be defined with following formula [2]:

$$p(w_0 \mid w_1) = \frac{\exp\left({v'_{w0}}^{T} v_{w1}\right)}{\sum_{w=1}^{W} \exp\left({v'_{w}}^{T} v_{w1}\right)} \tag{1}$$

The whole skip gram model is then applied to simple neural network which uses softmax activation function:

$$soft\max_i(a) = \frac{\exp(a_i)}{\sum_{j=1}^{n} \exp(a_j)} \tag{2}$$

To improve performance of training time, authors improve softmax function into hierarchical structure. They use knowledge of Huffman trees and applied it. The optimization cause significant reduction of training time and therefore we use it during tests.

5 Experiment

5.1 Dataset and preprocessing

The quality of word2vec output is highly dependent on large and well pre-processed corpora. The scientist from Google which originally introduce this new method had enormous data banks which they could use for the purposes. As the Google sources for Slovak language are not available, we extracted 3 different Slovak datasets. All datasets have their specifics. Quality of word2vec model with usage of different corpora indicates domain independence of the method. At the same time, we use multiple dataset size for real estate datasets to see how size of dataset improves accuracy.

5.1.1 Real estate

We consider this dataset as referential, because we evaluate the performance of our search engine in the real estate domain. Therefore we took special attention to analysis of the corpora. The first dataset was taken from 80 000 offers. It consists of 13 900 unique words and 2.6 million words in total. Next dataset size was based on 150 000 descriptions of offers. We parsed 39 000 unique words with almost 14 million words in total. The largest train corpora consists of 360 000 textual description of estate offers. It represents 57 000 unique words and almost 30 million words at total.

We devoted special attention to pre-processing the corpora. Preprocessing of unformatted natural language sentences which contains multiple patterns of the same thing and spell errors is much harder. Except more advanced techniques, we embraced the standard ones. In first phase, we converted letters to lower case and replaced letters with diacritics with their ASCII equivalent.

One disadvantage of word2vec model is that it can not handle continuous values and specific identifiers(emails, phone numbers) correctly because of their sparse co-occurrence with other words. Therefore we applied special kind of text annotation, which we consider as novel in the area. We recognize patterns like total number of floors in apartment, phone numbers, email addresses, areas etc. Identified text areas were replaced with URI representation e.g number 20 is replaced with http://www.w3.org/2001/XMLSchema#integer. Continuous values are only one aspect of text annotation. We applied also annotation of ontology entities in corpora. This can help us to use word2vec as predictor in query creating. Base on word we can ask word2vec model what word occurs in similar situations. Results could help us navigate in ontology graph and create more precise SPARQL query.

5.1.2 Other

The real estate dataset was not the only corpora we used. We collected and preprocessed raw text from different domains as well. Our next corporas were Slovak Wikipedia and Europarl corpus. Wikipedia corpus needs specific kind of text preprocessing as we can rely on almost correct formatted text. The Slovak Wikipedia consists of 250 000 unique words and 32 millions of words in total. Europarl corpus was smaller as it consists only from 62 000 unique words and 13 millions of words in total.

5.2 Performance Results

The testing environment uses 4 cores of Intel Xenon server with 50GB RAM. We decided to use Linux in RHEL distribution as operating system where all tests ran.

Table 1. Training time.

	d=100,k=5	d=200,k=5	d=800,k=5	d=100,k=7	d=200,k=7	d=800,k=7
reality -13k	1m 24s	2m 47s	10m 51s	1m 50s	3m 42s	14m 17s
reality – 39k	9m 17s	23m 37s	1h 29m	17m 21s	34m 43s	2h 4m
reality – 57k	18m 1s	33m 12s	1h 57m	21m 49s	41m 13s	2h 45m
wiki	18m 17s	36m 14s	2h 19m	24m 16s	46m 40s	3h 4m
europarl	6m 27s	12m 38s	49m 20s	8m 3s	15m 58	1h 2m
combination	29m 7s	57m 32s	6h 50m	54m 48s	109m 30s	7h 10m

The first test describes speed and performance of different dataset. We did 36 different tests and measure training time. The results are written in Table 1. The following results were achieved using hierarchical softmax optimization, without subsampling. The parameter d in the table represents dimension of resulting vector space. Parameter k represents window size in k-skip-n-gram training model.

The results from Table 1 show almost linear scaling of method independent from dataset. The method scales linearly with increasing dimensionality of model. Parameter k is not linearly scalable. In most cases, it scales as one half of percentage gain of parameter k. It means that when we raise parameter k by 40%, the time increased in about 20%.

Training time is only one side of gaining results. We have to analyze time of receiving answer from trained model. The standard test consists of receiving 10 most related words. Such task took 60ms on real estate corpora with 13000 unique words. We consider the result as very promising for usage in search engine, which emphasis low response latency.

5.3 Usability analysis

The trained vector model has for us multiple areas of application. The first benefit is that we do not have to rely on official dictionaries as we mentioned before. Instead, it creates domain specific language model, which suits the best for domain specific search engine. For languages where area of research is not so developed is this new approach very important.

Natural user queries are very sparse. It is common that people use synonyms to represent the same meaning. Our trained model is able to provide the synonyms which are available in the selected domain. At the same time, we identified that is possible in some limited sense to use the model for morphological resolution. This helps to our search engine annotator to recognize user queries and translate them to SPARQL.

The work with real natural language without formatting helped us to discover totally new area of usage. We identified, that our model is able to handle misspellings and shortcuts and provide correct word form. As an example we can point to Slovak word “poschodie”. Except the typical synonym like “podlazie” the model offers numerous other interesting responses like : poschode, pochode, posch which have the same meaning like original word even though they are not part of any dictionary because they represent typical human mistakes and shortcuts. With such knowledge, we are able to automatically correct human mistakes in natural language queries. Other words, which presents very interesting results are rozloha and predaj.

Results for rozloha word:

vymera, rozlohu, plocha, rozlohou, rozhloha, rozlohe, vymere, rozl, vymeru

Results for predaj word :

napredaj, predam, pradaj, ponuame, ponukama, pedaj

As we can see, we can get multiple morphological forms of word rozloha. Therefore we believe, we will be able to increase accuracy of recognition words in natural user queries. Another resolution is that we can deal with shortcuts. It is very important as in human language, we use shortcuts often. We have to mention that we were not able to solve misspellings with the smallest realestate dataset. It proofs the necessity of large corpora to get accurate results.

The list of possible usage applications share one common feature. Suggested words from trained model are defined through cosine similarity. Cosine distance between words can be directly connected with probability model which is present in search engine. The size of likelihood measure can influence the final natural language translation.

6 Conclusion and future work

In the work we presented brief picture of our proposed architecture of adaptable multilingual search engine. We aim reader’s attention to better understanding of principles how word2vec works. To proof usability of word2vec as general interface for multilingual engine, we test the performance on multiple Slovak datasets from various domains. The speed of training shows usability of neural network like method even in commercial application on standard hardware.

Based on training results, we see that trained model can deal except synonyms also with morphological forms and unexpectedly spelling errors. This can have big impact on accuracy of our search engine as it will certainly improve word recognition in query. The tool word2vec is freely available and therefore results of our work are transparent as anyone can reproduce them. We believe, that with larger corpora, we can substitute official dictionaries in future.

In future we plan to evaluate benefits of word2vec implementation into our search engine. As this paper indicates, we can expects positive impacts of the model to final word annotation. The experiment should contain detailed comparison of trained model with another word source – dictionary. Another area for improvement of this work is to increase size of current dataset. As we mention in the beginning, accuracy of the model is depended on the size of provided corpora. It will play key aspect in our evaluation in the paper.

We are aware of lack text for processing in Slovak language. Therefore we want to deeply investigate new methods which are built upon word2vec. They were recently presented at NIPS conference [7]. The results from the works show, that we can get the same quality vectors with ¼ of corpora size.

Acknowkedgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Mikolov, Tomas and Deoras, Anoop and Kombrink, Stefan and Burget, Lukáš and Černocký, Ján, Empirical Evaluation and Combination of Advanced Language Modeling Techniques, p 605-608. ISCA, 2011

[2] Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey, Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[3] Mikolov, Tomas and Sutskeyer, Ilya and Chen, Kai and Corrado, Greg and Dean, Jeffrey., Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

[4] Weikum, Gjergji Kasneci and Ramanath Maya and Suchanek Fabian and Gerhard. The YAGO-NAGA approach to knowledge discovery. s.l. : SIGMOD Rec.

[5] Chong Wang and Miao Xiong and Qi Zhou and Yong Yu. PANTO -- a portable natural language interface to ontologies. In: 4th ESWC, Innsbruck. 2007, s. 473--487.

[6] Valencia-García, Rafael and García-Sánchez, Francisco and Castellanos-Nieves, Dagoberto. OWLPath: An OWL Ontology-Guided Query Editor. IEEE Transactions on systems, man, and cybernetics—Part A: Systems And Humans, V. 1, 2011, Zv. 41.

[7] Mnih, Andriy and Kavukcuoglu, Koray. Learning word embeddings efficiently with noise-contrastive estimation. In Proceedings of NIPS, 2013

Method for Novelty Recommendation Using Topic Modelling

Matúš TOMLEIN[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
matus@tomlein.org

Abstract. The web content has a very dynamic nature, it frequently changes and spreads over various information channels on the Web. The huge amounts of content on the Web make it difficult to find web pages that provide novel information. In our work, we aim to model the information value of web content in order to recommend novel articles to the user. To this end, we use topic modeling to work with information in web content at a higher level. We focus our method in the domain of news articles. We introduce a method for clustering articles based on their relevancy and also a method for ranking topics based on their novelty. We evaluate and compare our method with two other baseline methods for novelty detection. The evaluation shows that articles recommended by our method have much improved relevancy to the interests of the user while maintaining comparable novelty to the baseline methods.

1 Introduction

There is a large number of news and other articles being published on the web by various large and small portals every day. However, the content in these articles is often repeated among them and most of them contain little novel information.

When a person reads an article about a specific topic, it is very likely that they might find dozens of similar articles on the Web, giving the same information in a different way. Such articles are not interesting to the reader. However, there might also be numerous related articles that contain significant novel and interesting information. The problem we deal with is to identify articles with novel and relevant information.

To illustrate this problem, let us look at the following example. A person interested in smartphones, reads an article titled *"Windows Phone wins bigger chunk of smartphone market"*. There are two other articles, similar to this one:

Windows Phone gains, but Android rules (A)

[*] Master study programme in field: Software Engineering
Supervisor: Dr. Jozef Tvarožek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Windows Phone grows, but so do non-Samsung Android vendors (B)

Articles *A* and *B* give very little new information compared to the one the reader has already read and so they are of small interest to them. On the other hand, an article titled *"21 percent of smartphones shipped in Q3 were big-screened behemoths"* might be interesting to the reader as it probably contains novel information about screen sizes, which is relevant to the topic of smartphones. It is therefore a good candidate for recommendation to the reader. In our work, we want to contribute to the field of novelty news recommendation by proposing a method for novelty recommendation based on topic modeling. This article describes our method for novelty recommendation and an experiment we conducted to evaluate it, along with the results of the experiment.

2 Related work

There were three workshops, called TREC Novelty tracks, focusing on novelty detection. For each workshop, there was a manually created dataset that the researchers tested their methods against. The novelty tracks focused on sentence-level novelty detection and the datasets consisted of sentences rated by their novelty and relevancy from a set of documents [5].

There were also attempts to create news recommender systems that applied novelty detection methods to provide an interface for users to find articles with novel information [1,2]. They applied various difference metrics for novelty detection, like inverse cosine similarity, Kullback-Leibler divergence, density of previously unseen named entities, quantifiers and quotes. Using similarity measures as the basis for novelty detection is also common in other works [6].

So far, the use of topic modeling in novelty detection has not been widely explored. There has been a research comparing the Latent Dirichlet Allocation and Pachinko Allocation Model methods for topic modeling with cosine similarity in novelty detection and it showed promising results in favor of topic modeling [4].

3 Method for novelty recommendation based on topic modeling

Our goal is to design and evaluate a method for news article recommendation that recommends articles based on their novelty to the reader. The articles should be relevant to other articles that the user read. We work with the following hypothesis:

- Taking the novelty of articles into account leads to better results of a recommendation
- Topic modeling is a better approach to detecting relevant novel information than using an inverse similarity or divergence measure

The overview of the method is shown in Figure 1.

3.1 Topic modeling

Our method uses topic modeling in order to calculate the novelty and relevancy of articles. Topics are sets of relevant words with some probabilistic degree of distribution with them [4]. We use the Latent Dirichlet Allocation algorithm for topic modeling. The reason why we think topic modeling can be useful in novelty recommendation is that it provides a way to work with the information in articles on a higher level of abstraction. It allows us to work with information using topics as opposed to using keywords. This is particularly useful if we want to track similar information across articles and find novel groups of information in them.

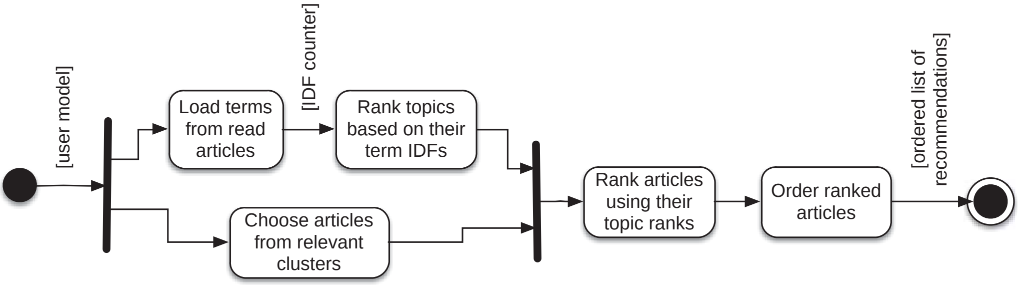

[user model]
Load terms from read articles
[IDF counter]
Rank topics based on their term IDFs
Choose articles from relevant clusters
Rank articles using their topic ranks
Order ranked articles
[ordered list of recommendations]

Figure 1. Overview of the method. First chooses relevant articles and ranks the topics using the user model. Then it ranks the relevant articles based on their topics and orders them by their rank.

3.2 Relevancy of the recommended articles

It is important to ensure that the recommended articles are relevant to the interests of the users, i.e. to what they previously read about. To achieve this, we perform the recommendation in two steps:

1. Create clusters of similar articles
2. Recommend novel articles within the clusters

It means that we recommend articles by their novelty only from the clusters of articles the user previously read. We designed a simple method for clustering articles based on their topics. We decided to design and implement our own method because we wanted to make use of our topic model and for its simplicity. The method creates clusters based on the topic pairs of articles - it creates a cluster for each topic pair, which is any of all the possible 2-combinations of topics. Then it assigns articles to clusters based on their topic pairs. Afterward, it removes weak clusters (clusters with less than 4 articles) and weak articles from clusters (articles with small probabilities).

3.3 User model

The main purpose of our user model is to store information about the articles the user read. It contains the following information:

– List of read articles

– List of topics of the read articles along with their probabilities retrieved from the topic model

3.4 Topic ranking

Topics retrieved from LDA have various qualities. Some contain important information, some are just groups of words without significant importance or meaning. These less important topics can have an impact on the performance of our method and so it is useful to give them a lesser importance when considering their contribution. We also want to give a lesser importance to topics that group information the user already read about. This is a crucial part of our method that ensures the novelty in our recommendation. To meet this goal, we employ topic ranking. We give each topic a numeric rank that represents its importance and novelty to the user. The rank of a topic is calculated separately for each user based on their user model.

We use an algorithm inspired by the method proposed in [3] that calculates the novelty of an article based on the Inverse Document Frequency (IDF) of its terms. We use the average IDF of the 100 best terms of a topic to calculate its rank. As the corpus of documents for calculating the IDF against, we use the articles the user read. The rank of a topic is calculated using the formula 1, where *T* is the collection of terms and their probabilities in the topic, *t* is a term, *w* is the weight of the term, *idf* is the function for computing the IDF of a term and *k* is a constant which we set to 100.

$$TR(T) = 1 - k * \frac{\sum_{t,w \in T} idf(t) * w}{|T|} \tag{1}$$

By using the read articles as the corpus for calculating IDF, we address both of our goals mentioned above:

1. We give a lesser importance to topics containing non-important terms – terms that are frequent in the other read articles

2. We give a higher rank to topics that contain novel terms that user didn't previously read about

4 Evaluation

We evaluated our method in a preliminary experiment. The goal of the experiment was to find out the advantages and disadvantages of our method compared to two other commonly used methods for novelty detection. Our expectation preceding the experiment was that the recommendations produced by our method would be the most interesting to the user out of the three compared methods. We expected this outcome because our method doesn't blindly recommend articles that are the most dissimilar to the ones the user read, but the ones that contain new topics.

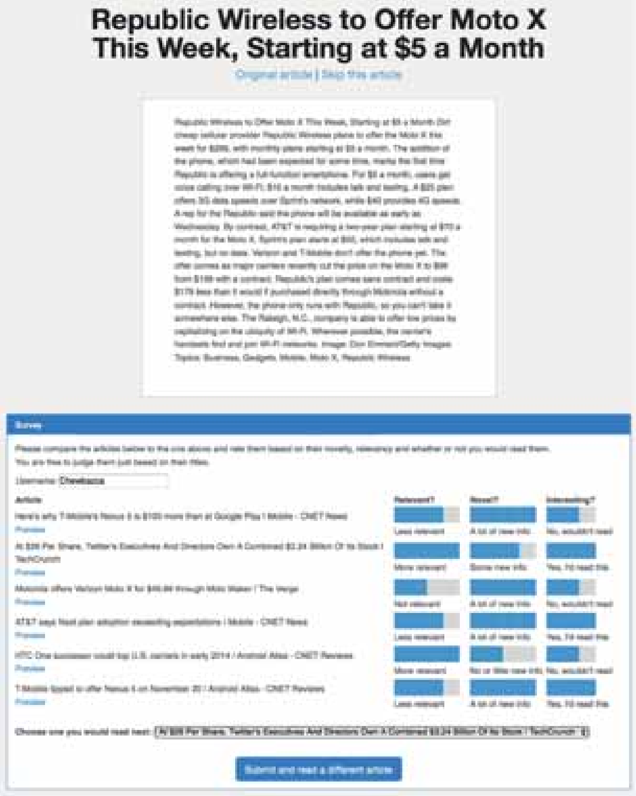
4.1 Description of the experiment

The user interface of our experiment is shown in Figure 2. It was a web page that showed an article at the top and a feedback form at the bottom. The form consisted of 4–10 other articles that were related to the article above. The task of the participants of the experiment was to compare the listed articles to the one above based on their novelty, relevancy and how interesting they were, on a scale of 3. We also asked them to choose one article that they would like to read next. We collected this explicit feedback and used it to compare our method with two baseline methods used for novelty recommendation: inverse cosine similarity and IDF scored novelty detection [3]. The experiment went on for a day and a half and 5 subjects (university students) took part in it. They compared 152 pairs of articles. The articles being compared were retrieved from several well-known tech blogs.

4.2 Results

The experiment has shown that the perception of what is novel information and what is not is very subjective. The participants used different scales for rating the novelty and relevancy of the articles, some of them rarely using the option *"A lot of new information"*. We also received feedback that the rating of interestingness of articles was unclear as it could have been influenced by various factors. The evaluation of the first part of the experiment went as follows. If the user rated article A as more novel (relevant, interesting) than article B, we tested if a given method also ranked article A higher than article B. To evaluate the choice of one article that the user picked to read next, we considered an algorithm successful if it listed the chosen article among the first 3 recommendations.

The results are shown in Figure 3. As the chart shows, our method ranked by far the highest in relevancy of its recommendations. In novelty as well as in interestingness of its results, the IDF scored novelty detection scored the highest. Our method was the most successful in recommending



Republic Wireless to Offer Moto X This Week, Starting at $5 a Month

Original article | Skip this article

Republic Wireless to Offer Moto X This Week, Starting at $5 a Month Dirt cheap cellular provider Republic Wireless plans to offer the Moto X this week for $299, with monthly plans starting at $5 a month. The addition of the phone, which had been expected for some time, marks the first time Republic is offering a full-function smartphone. For $5 a month, users get voice calling over Wi-Fi; $10 a month includes talk and texting. A $25 plan offers 3G data speeds over Sprint's network, while $40 provides 4G speeds. A rep for the Republic said the phone will be available as early as Wednesday. By contrast, AT&T is requiring a two-year plan starting at $70 a month for the Moto X. Sprint's plan starts at $80, which includes talk and texting, but no data. Verizon and T-Mobile don't offer the phone yet. The offer comes as major carriers recently cut the price on the Moto X to $99 from $199 with a contract. Republic's plan comes sans contract and costs $179 less than it would if purchased directly through Motorola without a contract. However, the phone only runs with Republic, so you can't take it somewhere else. The Raleigh, N.C., company is able to offer low prices by capitalizing on the ubiquity of Wi-Fi. Wherever possible, the carrier's handsets find and join Wi-Fi networks. Image: Don Emmert/Getty Images Topics: Business, Gadgets, Mobile, Moto X, Republic Wireless

Survey

Please compare the articles below to the one above and rate them based on their novelty, relevancy and whether or not you would read them.
You are free to judge them just based on their titles.

Username: Chewbacca

Article	Relevant?	Novel?	Interesting?
Here's why T-Mobile's Nexus 5 is $100 more than at Google Play | Mobile - CNET News Preview	Less relevant	A lot of new info	No, wouldn't read
At $26 Per Share, Twitter's Executives And Directors Own A Combined $3.24 Billion Of Its Stock | TechCrunch Preview	More relevant	Some new info	Yes, I'd read this
Motorola offers Verizon Moto X for $49.99 through Moto Maker / The Verge Preview	Not relevant	A lot of new info	No, wouldn't read
AT&T says Next plan adoption exceeding expectations | Mobile - CNET News Preview	Less relevant	A lot of new info	Yes, I'd read this
HTC One successor could top U.S. carriers in early 2014 / Android Atlas - CNET Reviews Preview	More relevant	No or little new info	No, wouldn't read
T-Mobile tipped to offer Nexus 5 on November 20 / Android Atlas - CNET Reviews Preview	Less relevant	A lot of new info	Yes, I'd read this

Choose one you would read next: At $26 Per Share, Twitter's Executives And Directors Own A Combined $3.24 Billion Of Its Stock / TechCrunch

Figure 2. The user interface of our experiment, showing the base article at the top and relevant articles at the bottom. The task of the participants was to compare the articles by their novelty, relevancy and interestingness.

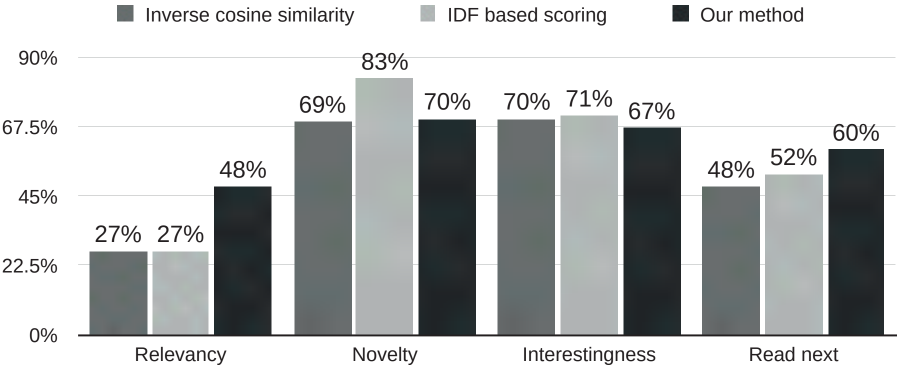


Figure 3. Comparison of our method with two baseline methods. It shows that our method was by far the most successful in the relevancy of its results and in recommending articles chosen to be read next. The IDF based scoring method showed the best results in terms of novelty.

articles the participants chose to read next. From the results of this experiment, we can conclude that our method is a valid approach to recommendation based on novelty, that has its strengths mainly in the relevancy of its results to the users interests. The recommendations based on IDF scoring had better attributes of novelty, however at the cost of their relevancy.

5 Conclusions

We proposed a method for recommending articles based on their novelty. Our method makes use of topic modeling to represent groups of similar information. It creates clusters of similar articles using the topic model. It ranks topics by their novelty to the user using an algorithm that employs IDF based scoring.

We evaluated our method in an experiment, where we collected explicit feedback from several users. We compared our method with two other commonly used methods. The results were optimistic, giving better results in terms of relevancy of articles and also in recommending articles the participants chose to read next. The method for novelty detection using IDF scoring of terms gave better results in terms of novelty, however their relevancy was poor.

In future work, we plan to improve the relevancy scoring of our algorithm by using a commonly used clustering algorithm, such as K-means or hierarchical clustering. We will focus on the scalability of our method and consider replacing the LDA algorithm for topic modeling with Latent Semantic Indexing (LSI).

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] Gabrilovich, E., Dumais, S., Horvitz, E.: Newsjunkie : Providing Personalized Newsfeeds via Analysis of Information Novelty.

[2] Iacobelli, F., Birnbaum, L., Hammond, K.J.: Tell me more, not just more of the same. *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10*, 2010, p. 81.

[3] Karkali, M., Rousseau, F., Ntoulas, A.: Efficient Online Novelty Detection in News Streams.

[4] Sendhilkumar, S., Nandhini, N., Mahalakshmi, G.: NOVELTY DETECTION VIA TOPIC MODELING IN RESEARCH ARTICLES. *airccj.org*, 2013, pp. 401–410.

[5] Soboroff, I., Harman, D.: Novelty Detection: The TREC Experience. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics, 2005, pp. 105–112.

[6] Tsai, F.S., Zhang, Y.: D2S: Document-to-sentence Framework for Novelty Detection. *Knowl. Inf. Syst.*, 2011, vol. 29, no. 2, pp. 419–433.

Method for Navigation in Research Papers with User's Preferences

Zuzana UJHELYIOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
Zuzana.ujhelyiova@gmail.com

Abstract. Amount of the research papers available on the internet can be confusing for young researchers. It is hard to identify significant articles for user's topic of interest without reading them. We propose new navigation method for research papers based on qualitative attributes of the article to define their quality. It is personalized through the weighing coefficients set by user. Model values of weighing coefficients are provided to user. Quality to the articles influence decision making and define order of the articles in result set. Multiple inputs are used to define the most suitable result set in decision mechanism.

1 Introduction

There is a huge amount of research papers available for researchers in digital libraries and all over the internet. It can be really challenging for young researcher to find the right sources of information to the exactly defined theme. Two main ways for researchers to find the papers are using web search engines (like Google Scholar[1]) and through the digital libraries.

Using the web search engines bears the risk of insufficient quality of found papers. Papers are aggregated from both peer-reviewed and non-peer-reviewed journals. There is no quality guarantee with non-peer-reviewed journals. The main advantage of this way is that links to full texts of the articles are available for free.

By using digital library you have guaranteed the quality of found articles. Articles available in digital libraries (like IEEExplore[2] or ACM Digital Library[3]) are published by important publishers that guarantee the quality of the content using the peer-review system. The main disadvantage of using digital libraries is that most of the articles available here are paid for full text. Only abstract is available for free.

Metadata of the research papers that can help define their quality. All of the articles can be characterised by their attributes like number of citations (total or average), quality of authors (h-index or average number of citations), quality of source (journal impact factor or other metrics)

[*] Master degree study programme in field: Information Systems
Supervisor: Dr. Jozef Tvarožek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] http://scholar.google.com

[2] http://ieeexplore.ieee.org/Xplore/home.jsp

[3] http://dl.acm.org

and references. Using these attributes can help quantitatively evaluate the research papers. Each of these metrics has their disadvantages, so it is important to dedicate significant time to choosing the right combination of them [6, 7].

2 Related work

Most of the approaches used for search and navigation in research papers are based on semantic analysis of the text as well as content of the text and dynamical taxonomy. Navigation models are used in dynamical taxonomy to reduce the size of the resulted set [1]. Reduction of the resulted set assumes to return less amount but more accurate result data. Navigation models are based on queries in natural language where keywords are combined with navigation operators defined in advance that are interpreted into concrete navigation models. They change size of the result set as well as its order. Application of navigation models is differently difficult depending on the specified domain.

Approaches based on the semantic text analysis used in research papers navigation are similar to that used in the web navigation. Most of them try to find some similarities or patterns in data to define categories or clusters that minimize the result set for defined query [2]. Similarity in web search is mainly based on topics of the data. There can be found more tight links in research papers than in the internet. Research papers are connected by topic of the content, co-authorship, citing, source they were published in, etc.

Clustering can be used for finding the similarities in data without defining the classification classes in advance. There are multiple types of clustering algorithms with different approaches. Recently the clustering approaches with best results are based on the frequent item sets of words [3, 4].

All of the non-semantic attributes of the article can be used to define quality of it. There are metrics for article citations (total amount of citations, average amount of citations per year, etc.), for authors (value of h-index, total number of citations, number of citations per year), for sources they were published in (impact factor, modified page rank, modified hits), for references of article (number of citations) [5, 6, 7]. Right combination of defined quality of the article (relevance of the article) with semantic analysis of the text can return better result sets for defined query.

3 Method for personalized navigation based on attributes

We propose new navigation method for research papers based on combination of article quality and semantic analysis of the text. The main steps of the algorithm are defined in Figure 1. Parts with person pictogram represent user’s inputs like setting up weighing coefficients, defining the query or browsing through the data represented in graphs. Parts with letter A are the significant segments of the method like decision making and setting up quality of the article. Quality is defined as combination of quantitatively evaluated metadata of the article. Decision making takes multiple inputs into process of defining the right result set for the user’s query. Decision making and setting up quality values are discussed in detail in sections 3.1 and 3.2.

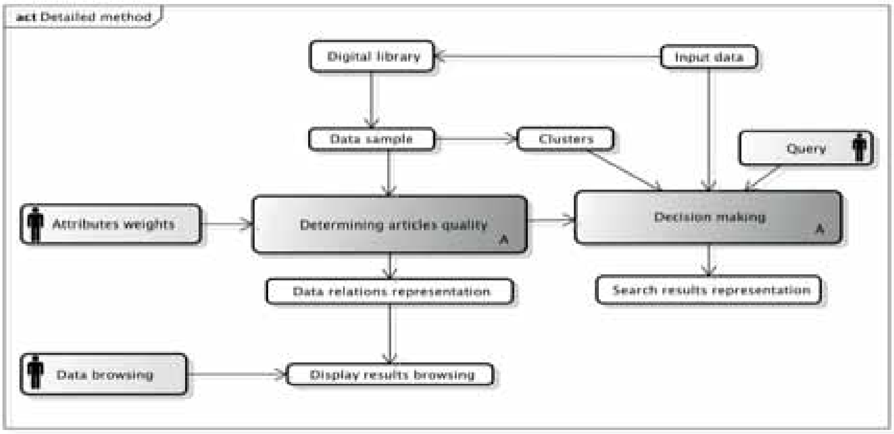


Figure 1. Detailed method proposal.

Representative data sample is needed for evaluation of the proposed method. Appropriate queries need to be chosen to get the data sample from the digital library. Data sample includes metadata about research papers (articles) as well as their full texts. Quality value will be determined based on combination of the attributes of the article (metadata) and the weights defined by the user. Model weights will be provided to the user defined by the researchers understanding the domain. User can use either the model weights or define his own. There are multiple relations and dependencies between research papers. These can be presented in graphs like relations between authors, sources or year of the publishing of the article. Data can be grouped by the quality values in the graph as well. All of these representations help user to find information in the data sample. He can use graphs to browse through the data which is more natural and comprehensible than using the lists.

Selected clustering algorithm based on frequent item sets is used to define similarities in the data sample (represented by found clusters). It helps to improve performance of the method by grouping the similar data into one cluster. These clusters are one of the inputs of the decision mechanism. Research papers with defined quality values are input of the decision making as well. User defines query to get the result for. It is compared with input data used to get data sample from the digital library and clusters with frequented item sets. Articles are selected to the result set. They are sorted based on the quality values to increase user's impact into the decision making process. Result set will be represented as ordered list to user as is used to from the digital libraries.

3.1 Determining quality of the article

Quality value is defined as combination of selected quantitative attributes of the article. Selected metadata used to gain value of quality of each article are

- total amount of citations of the article,
- values of h-indices of the authors,
- impact factor of the source of the article,
- numbers of citations of used references,
- age of the article.

We suppose that the right combination of these attributes can prefer articles with the higher values (like with better authors, better source, etc.). Each of these attributes has its own weighing coefficient that is used as user's input to process of finding the quality of the article. Quality is defined by equation

$$R = \frac{\alpha \ln P_C}{\beta (R_A - R_V)} + \sqrt[2]{\gamma \sum R_A} + \delta R_Z + \varepsilon \sum R_{CR}$$

where P_C is number of citations of the article. R_A-R_V is difference between the actual year and the year when article was published (age of the article). $\sum R_A$ represents sum of values of h-indices of the authors. R_Z is value of impact factor of the source were article was published and $\sum R_{CR}$ is sum of total amount of the citations of references. Logarithmical function used for total citation amount as well as square root function used for authors were used to balance the impact of each attribute (values were too high in comparison to other attributes). Age of the article is used in denominator to decrease disadvantage of less amount of citations for new articles.

Coefficients α, β, ɣ, δ, ε are used as personalization input of the user that can prioritize attributes by them. Default values of all weighing coefficients will be set to 1.0. Model values of the weighing coefficients will be presented to user based on settings defined by researchers that understand the analyzed domain. User can either use model values or try his own settings to get the different order of the relevant articles. We suggest that researcher can better define which articles in the result set are relevant to the query than the search engine. Order of the returned data based on quality values will be compared with the results from the digital library as well as with the order created by researcher.

3.2 Decision making

Decision mechanism is the most complicated part of the whole system. It combines multiple inputs to create the best and most relevant result set for user. Inputs of this part are

- query defined by user,
- created clusters with frequent item sets of words,
- input data used as queries for creating the data sample,
- articles from the data sample with quality values.

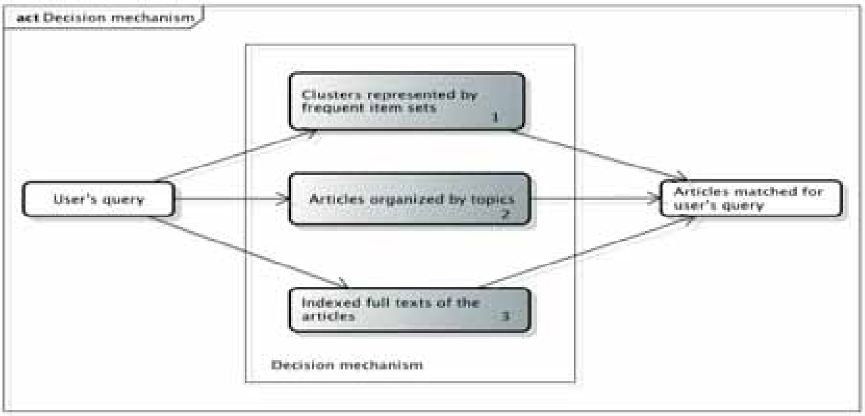

act Decision mechanism
Clusters represented by frequent item sets
1
User's query
Articles organized by topics
2
Articles matched for user's query
Indexed full texts of the articles
3
Decision mechanism

Figure 2. Decision mechanism parts.

Query defined by user will be preprocessed, stop words removed and enriched by the synonyms for the terms from input. Porter algorithm will be used for stemming of the input. Prepared query will be compared to different types of data as defined in Figure 2. Firstly, query will be compared to frequent item sets that represent created clusters (represented by 1 in Figure 2). If query matches either of the item set (or its part) selected cluster will be returned. Articles from the selected cluster will be ordered by the quality values. Ordering by the quality values ensures that result set is organized by user's preferences.

Second method will use input data used for creating data sample as is represented by 2 in Figure 2. These data represents topic of the articles. Each topic will be also preprocessed to ensure covering more inputs for the same meaning. After matching topic with query, articles from this topic will be sorted by quality values and best of them will be returned to the user.

Indexed full texts of the articles with the important attributes will be compared to the query in the third method (part 3 in Figure 2). Articles with the best match will be selected to the result set and ordered by quality values.

Size of the result set in each method will be limited to ten best matching articles. Limited size of the result set prevents overloading of the user. These different methods of choosing data to the result set will be evaluated by the experiments and combined to get the most relevant results. As user change the weighing coefficients data are retuned in real time.

4 Evaluation

We created corpus of 6000 research papers with full texts from Google Scholar for evaluation of out method. Queries for the data collecting were based on titles of the papers for IIT.SRC 2013[4]. First experiments based on quality values have been made. We compared our order based on quality values with order defined by Google Scholar results and order created by researcher.

We used articles gained for topic named *Method for social programming and code review* that were published in 2008. Ten articles were returned in the result set from Google Scholar and seventeen were referenced by them. Returned articles with titles and abstracts were provided to the researcher to define their order based on the relevance to the query. We made another experiment giving more impact to the citations of the article (coefficient α: 1.0 → 2.0), quality of the authors (coefficient ɣ: 1.0 → 1.5) and less impact to references of the article (coefficient ε: 1.0 → 0.7). Order created with our method (with default and changed weights) is compared to order from Google Scholar and order created by researcher in Figure 3.

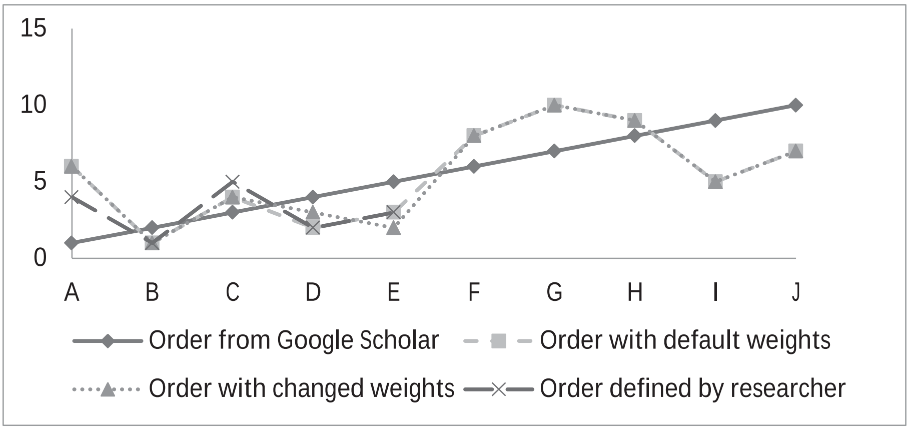


Figure 3. Comparing Scholar's order to order created using the proposed method.

Order created by quality values is more similar to order created by researcher than the order from Google Scholar. As researcher can better define if the article is relevant to the query this order is more significant for us than the one based on Google Scholar. We need to say that at this point there are only quantitative attributes used in our approach without using of decision mechanism.

[4] http://www.fiit.stuba.sk/generate_page.php?page_id=3786

Once the semantic analysis of the text will be used in decision making we assume that the results will be even better.

Results returned from the decision making will be compared to both results from digital library as well as order created by researcher. Decision making is dependent on the personalized quality values while order of the result set is based on them.

5 Conclusion

In this paper we presented new navigation method based on personalized quality values for the articles. Quality value depends on personalized weighing coefficient of the selected attributes that can user change to influence the order of the articles in the result set. Articles are selected to the result set with decision mechanism. It combines pre-processed user's query, topics of the articles in the data sample, created clusters with frequent item sets of the words and full text of the articles with quality values. Order of the articles in the result set is defined by quality values.

Articles are connected with different types of relations like references, citations, co-authorship or source that they were published in. All of these relations can be presented to the user in graphical representation to simplify the navigation in the domain. Quality values will be represented graphically as well.

We applied selected clustering algorithm on the data sample and created 82 clusters with frequent item sets of words. All of the inputs can be now combined in decision making. Results of the decision mechanism will be compared with the selected digital library. Quality of the returned result set will be evaluated against order defined by the researcher. We assume that partial results we got can be even better with using semantic analysis of the articles. Decision mechanism is including semantic analysis in the process of creating the result set. The last step will be creating user interface for the navigation system based on our method.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Ferre, S., Spangler, W.S.: *Agile Browsing of a Document Collection with Dynamic Taxonomies*. 19th International Conference on Database and Expert Systems Application, Turin, Italy (2008).

[2] Wongsuphasawat, K., Plaisant, C., Taieb-Maimon, M., Shneiderman, B.: *Querying event sequences by exact match or similarity search: Design and empirical evaluation.* Interacting with Computers, (March 2012).

[3] Fung, Benjamin CM, Ke, W., Ester, M.: *Hierarchical document clustering using frequent itemsets.* Proceedings of SIAM international conference on data mining, (2003).

[4] Qun, L., Xinyuan, H.: *Research on Text Clustering Algorithms*. 2nd International Workshop on Database Technology and Applications, (November 2010).

[5] Cheng, S., Yuntao, P., Junpeng, Y., Hong, G., ZhengLu, Y., ZhiYu, H.: *PageRank, HITS and Impact Factor for Journal Ranking*. Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, (2009).

[6] Bornmann, L., Mutz, R., Daniel, H.: *Are there better indices for evaluation purposes than the h index? A comparison of nine different variants of the h index using data from biomedicine*. Journal of the American Society for Information Science and Technology, (March 2008).

[7] Fassoulaki, A., et al.: *Self-citations in six anaesthesia journals and their significance in determining the impact factor*. British Journal of Anaesthesia, (2000).

Patterns in Browsing the Web: Distinguishing Computer Mouse Usage Characteristics

Peter KRÁTKY*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
kratky@fiit.stuba.sk

Abstract[1]

Every person is unique in the way he/she uses a computer, an operating system, programs and even input devices such a keyboard or a computer mouse. Patterns produced by usage of standard input devices could be utilized to identify an unauthorized user. Especially, adaptive and personalized systems accessed by multiple users might benefit from this feature as tailoring the content heavily depends on previous activity of the user. In our work, we focus on patterns in browsing the web, a common activity nowadays. We propose a new approach to identify a user browsing the web solely based on his/her behavior apart from the solutions identifying rather the browser than the person behind it. The process of user identification consists of three stages - acquisition of the data while browsing, construction of a user model and matching the model with template models to get the identity of the user. In our work we focus on the essential element of the identification process design that is a relevant user model. Thus, we seek characteristics the user model should hold in order to provide high suitability for the process of identification.

The user modelling starts with the acquisition of the data performed in an unobtrusive way. Four computer mouse events are tracked: mouse movement, click, mouse wheel movement, scrolling of a page, all of them assigned time and coordinates. In the next phase, characteristics representing user's mouse usage are extracted from the data (tangential velocity, curvature, etc.). The user model is represented by a vector holding mean and deviation for all the characteristics.

We conducted our first study and present preliminary results of distinctiveness of individual characteristics and their performance in the user identification. We acquired data from 17 users browsing a real running e-shop. The experiment design was enriched with gamification concepts in order to fully encourage users to perform intended actions. From the view of suitability for identification we examined how distinguishing the values of the characteristics are by quantifying the distinctiveness. The most distinctive characteristic is duration of a single click with rate of 0.77. We also evaluated how suitable the presented user model is for the identification process. The achieved result of the method is 63% success rate.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

[*] Doctoral degree study programme in field: Software Engineering
Supervisor: Assoc. Professor Daniela Chudá, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 197-203.

Users' Web Browsing Behaviour inside and outside a TEL System

Martin LABAJ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
martin.labaj@stuba.sk

Abstract[1]

Web browsing behaviour is used in various fields – from web usage mining, to its applications in adaptive systems, for example in user and domain modelling. Current browsers allow for parallel browsing (also called tabbing) – opening multiple web pages at once and switching between them. We describe an experiment on observing the parallel browsing behaviour, both in a technology enhanced learning (TEL) system and on the open Web, while using the TEL system as a recruitment and motivation tool for the participants to install the tracking extension.

For the purpose of this experiment, we implemented parallel browsing tracking as an extension in the Brumo platform. We captured user actions in individual tabs and combined them using our browsing reconstruction algorithm. The output is a browsing tree describing users' actions (linear visits or branching) and how they switched tabs or out of/into the browser – effectively reconstructing the entire session. In order to study participation and different approaches in browsing per different users, we created a dataset of 249 users. 80 users participated in the browsing study (installed the extension) during an experiment with ALEF system, where students were asked to augment the course content with external links discovered while browsing the open Web. The users were motivated with user score awarded for inserted links. 144 other users did not participate in the study, but used the ALEF during the course of the experiment and 25 more users participated in the browsing study outside of the experiment. These are essentially 3 groups of users, those: (1) who participated in the motivated study, (2) who were also motivated in the TEL system, but chose not to participate, and (3) who participated outside of the experiment. We computed churn data (when the user joined, left, etc.) for the participants and also included other traits, including academic performance, learning styles, and personality traits.

We then analysed relations between available attributes, expecting to predict whether a user would participate or not, or even better, when they would leave the experiment. In such cases, one could modify or improve the motivation in order to collect more data from/about the participants. Our results include correlations of whether and how actively has the user participated in this experiment/study with other activities in the TEL system.

Acknowledgement: This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. 009STU-4/2014.

[*] Doctoral degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 204-211.

Influence of Navigation Leads' Visualization on Digital Libraries Exploration

Róbert MÓRO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
robert.moro@stuba.sk

Abstract[1]

The most natural way of search and navigation seems browsing, which does not force users to split their attention between the navigation interface and the search results. In addition, it supports the idea of navigation-aided retrieval which understands the search results as mere starting points for further exploration. In order to emulate this behaviour and support users when researching a new domain in a digital library, we provide them with navigation leads, i.e. links to relevant documents, with which we enrich the documents' summaries (or abstracts).

We examine the influence of the navigation leads' different visualizations on the users' performance of exploratory search tasks in a digital libraries domain. We proposed three types of visualization, namely visualization in text of an abstract, under the text and in a cloud of terms next to the list of search results.

Advantage of the first approach is that the leads can be viewed in their context, on the other hand only words or phrases already present in abstracts can be used as navigation leads. When visualizing the leads under the text, they lose they immediate context, but do not get in the way of reading. Lastly, visualization in a cloud tries to mimic the tag cloud with one exception – only leads for currently retrieved set of documents are selected into the cloud.

We conducted a user study with five participants – bachelor and master students, whose task was to explore a new domain using the provided navigation interface in a web-based bookmarking system Annota. We hypothesized that visualizing leads in the text or under it will prove to be more immersive, thus resulting in more interaction with the search results in comparison with the visualization in a cloud and that consequently the users will acquire better understanding of the domain and the problem at hand with less (extraneous) cognitive load.

In order to evaluate our hypotheses we used different metrics, such as task success or subjective ratings. We also collected gaze tracking data using the eye-tracker Tobii X2. Our results show that some users prefer in-text visualization appreciating the leads' context, while others prefer cloud visualization, because of its ability to differentiate relevance of the terms. As a result we proposed a compromise solution that merges the advantages of these approaches together.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

* Doctoral degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 212-219.

Knowledge Acquisition in Community Question Answering

Ivan SRBA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`{name.surname}@stuba.sk`

Abstract[1]

Besides standard search engines, current possibilities of the Web allow us to employ many supplementary sources of information. These nontraditional sources of knowledge are often based on collective intelligence. Concept of collective intelligence refers to a shared knowledge which emerges from common collaboration of community of users that share common practice, interests or goals. Collective intelligence is present in many popular web systems, such as forums, social networking sites or wikis. In the recent years, the new forms of systems based on collective intelligence has appeared. One of them is Community Question Answering.

Community Question Answering (CQA) is a service where people can seek information by asking a question and share knowledge by providing an answer on the particular questions. One kind of CQA systems is providing users with a possibility to ask any general question without any topic restriction (e.g. Yahoo! Answers or Wiki Answers). On the other hand, there are also topic-focused CQA systems dedicated to specific domains (e.g. Stack Overflow where users concern with questions related to the programming).

We suppose that besides receiving an answer on a question, an acquisition of a new knowledge is another motivational factor in CQA systems, which has not been fully discovered yet. On the basis of the dataset from Stack Overflow, which is one of the most popular CQA systems, we proposed post and user profiles which allow us to model question answering process on more abstract level. Consequently, we employed these profiles in analyzes which confirmed our hypotheses that 4 specified scenarios in question answering process positively leads to acquisition of a new knowledge.

The results open a novel perspective on CQA systems as non-traditional learning environments. We can derive several important implications from this finding. Especially in educational organizations, concept of CQA systems can be employed as a supplement to formal learning in particular courses or even as an immediate component of learning process where community of students together with teachers can participate on solving questions related to students' learning.

Acknowledgement: This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. 009STU-4/2014.

[*] Doctoral degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 220-227.

User’s Interest Detection through Eye Tracking for Related Document Retrieval

Jakub ŠEVCECH*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
sevcech@fiit.stuba.sk

Abstract[1]

Eye trackers are becoming more and more affordable and in the near future it is possible that every home computer or mobile device will be equipped with such a device. Eye movement tracking may become another mean of interaction of ordinary people with computers.

Many studies used various types of implicit user feedback such as mouse movement, clickthrough data, text selection or bookmarking as a source of information for user modelling, recommendation, user interface adaptation etc. They were used to quantify document quality and to identify user’s interest in document content on both document and sub-document level. In this study we use eye movement data captured while reading a document to identify sections of the document, user is most interested in. We describe multiple patterns in eye movement of users reading documents that can be used to identify important document fragments. By studying gaze data collected using Tobii X2-30 eye tracker while reading several articles, we identified seven base patterns in eye movement on the level of fixations and saccades that can be used to identify document fragments readers are most interested in. For every identified pattern we defined possible meanings for the importance of associated document fragments.

We used the identified patterns to determine the user’s interest in text fragments in studied documents and to compare effectiveness of extracted document fragments with user created in-text highlights in the task of related document retrieval. We were interested in qualitative study and comparison of properties of extracted interesting document fragments and highlights users attached to the documents. Using gaze data and identified eye movement patterns, we were able to extract smaller number of longer, important document sections compared to manual annotations. However, user interest identification using gaze data is prone for false positive errors due to misinterpretation of external interruption such as reading distractions or complicated text structure. To improve precision of pattern interpretation in gaze data, more external signals such as document content or other forms of user feedback would be necessary. Using gaze data as interest indicators we were able to obtain comparable, even slightly better results in related document retrieval as when using in-text highlight as important section identification.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11.

[*] Doctoral degree study programme in field: Software Systems
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 228-235.

Computer Science and Artificial Intelligence

Tree Structure Design of Boolean Expressions for Purpose of Genetic Programming

Peter FILÍPEK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
pfilipek.fiit@gmail.com

Abstract. In this paper we discuss a usage of genetic programming and tree structure for the approximation of Boolean functions. Boolean functions will be approximated by techniques used in genetic algorithms such as selection, reproduction, crossover and mutation, which are suitably applied for example in terms of symbolic regression. We used a tree structure for representation of Boolean expressions. To find the expressions representing Boolean functions we used unsupervised learning for rating the individuals in population.

1 Introduction

Design of Boolean functions is in the world of Electrical Engineering and Computer Science very common and important thing. The most commonly used method for design of Boolean functions is the use of conjunctive normal form or disjunctive normal form. Boolean functions designed by using this formula contain only three basic Boolean operators: conjunction, disjunction and negation. Through genetic programming we can simply design a Boolean function that may have a larger diversity of operators.

Genetic programming is an evolutionary algorithm which is used to optimize complex problems. One of the main authors in using of genetic programing is John R. Koza [1] who in 1988 got the patent on it. Characteristic for genetic programming is data representation by tree structures. Tree structure is suitable for representing functions and therefore it is easy to apply main operations for genetic programing such as mutation and crossover. This makes genetic programming very useful for approximation of Boolean functions.

In this contribution we closely look on some key information about Boolean functions which are important for our project. Then we take a look on the main operations and methods which are used in genetic programming.

2 Boolean functions

Boolean functions transform binary inputs to binary outputs. This transformation is constructed by Boolean operators. Boolean function is just an expression which merges Boolean operators with

[*] Bachelor degree study programme in field: Informatics
Supervisor: Ladislav Clementis, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 239–244.

binary inputs to get required outputs. One of challenging task is to find this expressions, if desired outputs for given inputs are known.

To complete our task by using genetic programming, it is essential to know some information about Boolean operators and Boolean variables. But there is much more about Boolean functions what is appropriate to know and most of it we can learn from knowledge about Boolean algebra [2, 3].

3 Genetic programming

Main mechanisms and methods of genetic programming are inspired by nature. Nature and genetic programming are using very similar evolutionary methods to approximate desired solutions [4].

3.1 Data structures

Tree structures are useful for storing most of mathematical expressions, for Boolean function are appropriate. Furthermore, all genetic procedures are applicable easily. Tree structures of Boolean functions are very similar to classic binary trees. Difference is only in node structures, which are extended for some information about functions. Each node includes references to parent and children. Several nodes together create one solution - individual.

3.1.1 Node structure

In our solution, there are very simple node structures. We add two new properties for storing extra information. First one is for storing information about operator type and second one is storing information about variable type. Other items are just used for storing references to parent node and left and right child, as usually.

Information about operators and variables are stored by integer variables. There we use a simple number substitution. For each Boolean operator we use different number to represent them. Same idea is used for variables with a little difference in values now represents variable. Zero is a special number for both: operators and variables. When operator or variable is equal to zero it indicates, that this node is not operator or variable. Operator and variable can't both equal to zero in the same node. This number substitution is giving as a good opportunity to expand our solution for another Boolean operator if needed in future.

In our solutions we don't work with constant for simplification.

3.2 Valid individual

Creating valid individuals at the population initialization is important for whole run of genetic algorithm. These first individuals must be created randomly, but correctly. It is very important to create a first generation of individuals diverse in all directions, size, composition of the operators etc.

In our solution creation of valid individual is very simple and it starts from bottom of the tree. First we start with randomly generated variables which are saved in stack. The quantity of generated variables is determined by a randomly generated number from interval $\langle 1, \frac{n}{2} \rangle$, n is the maximum tree size, more in the Crossover. Then we randomly generate operator node and randomly choose one or two another nodes from the stack, one for unary operator and two for binary operator. Then we join those nodes to new one as children. This will create a sub-tree whose root is a margin newly generated operator node. Root of this sub-tree we save back to stack. This cycle we repeat until the stack contains only one node. This one node in stack is root of our new individual which is ready to be added to first generation.

3.3 New generation

Process of creating new generation is very important for genetic programing. In this process we use all methods characteristic for genetic programing to get final solution.

First we randomly generate 256 individuals. These individuals compose population of our first generation. We rate each individual by corresponding fitness. If we don't find final solution, we save the best and worst one for statistics.

Then we start with crating new population for next generation. Firs we select 128 pairs of individuals from current generation. Individuals are selected by roulette selection. Roulette is using fitness values for selection. Then all 128 pairs are sent to the crossover. By the crossover we try to create new individuals with better fitness values. New individuals are sent to the mutation for little correction of their nodes. After the crossover and the mutation performed, we rate new individuals with fitness and add them to the new generation. We repeat this cycle until we find the solution, or reach generation limit, which is 100 generations work perfectly with 3 variable inputs.

For functions of 3 input variables we have found the solution from 5 to 20 generations in average. When we test functions of 4 input variables we increase population to 512 individuals and solutions were found between 20 and 150 generation in average.

3.3.1 Fitness

Fitness is very important value. Each individual have this value. Fitness represents success of individual compare to wanted solution. This value helps as to find theoretical potential of individuals [4].

In Hamming distance numbers of different elements on same position in two different strings are counted. In our solution fitness is calculated by reversed Hamming distance. This means, that we count number of same elements on same position in two different strings. In our solution we compare truth values of actual individual to wanted outputs.

Truth values are calculated for each possible variant of input variables. They are calculated by recursive method. Root sends request for result to his children and they send this request to their children and so on until this request reach leafs – the bottom nodes of the tree. Then they start sending results back to the root.

3.3.2 Selection of individuals

The selection is important for creating new generation. The selection method must give a chance to all individual to be selected. But individuals with better fitness must have better chance to be selected, as others with lower fitness.

We chose the roulette as selection method for our solution. First we need to create interval from which the numbers will be generated. The interval is created by sum of fitness raised to a power of two.

$$\left\langle 1, \sum_{i=0}^{n} Fitness_i^2 \right\rangle$$

Then we randomly generate a number from this interval. This number is representing one individual in the population. To get the index of individual we must gradually decrement this number by fitness raised to a power of two of all individuals until is not less than zero. Then we create a clone of this individual. This clone is stored in a list for next population.

This cycle we repeat until the list has the same size as the actual population. Fitness raised to a power of two exponentially increases size of interval for each individual. This way we achieve better prioritization of individuals with better fitness.

3.3.3 Crossover

The crossover is one of main methods used in genetic programing to create new individuals. This method is based on exchange block of information between two individuals to create two new ones [5].

In our solution we store only root of the tree. So in first step we create list of all nodes for both individuals which were send to the crossover. Lists are created simple by pre-order riding. Then we randomly chose one node form each list. These nodes are used as rood of sub-trees. These sub-trees are the blocks of information to exchange. It is important to correctly exchange references to parent's nodes and in each parent node the reference to his child nodes (see Figure 1 and Figure 2).

Our solution is little bit different as standardly used. In our solution we don't work with block of the same size. We make this decision because final expression could have different length than the expressions which were generated at the beginning. By this change of the standard algorithm we can get form the crossover individuals with different size. But this has one negative impact. With each passing generation we could get bigger and bigger individuals. This happens when one individual exchange smaller blocks for bigger blocks many times. The calculations of truth values for extremely large individuals are very time consuming. We limit the size of individuals to 100 nodes to avoid this problem. If after the crossover one of the new individuals will have more than 100 nodes, both are not added to the population and the procedure of the crossover will start again.

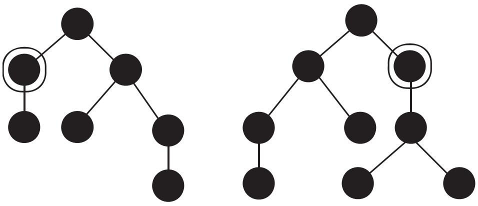


Figure 1. Two individuals before the crossover [5].

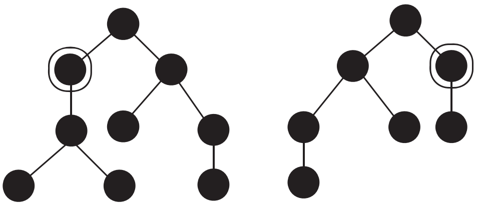


Figure 2. Two individuals after the crossover [5].

3.3.4 Mutation

The mutation is important for preserving genetic diversity. There is a possibility that we don't generate all possible operators or variables at the beginning. But another big risk is that some operators or variables can be lost because some individuals for next population are not selected. These lost operators and variables could be needed to successfully find a final solution. With the mutation all nodes have little chance to be changed to another.

We apply the mutation on new individuals created by the crossover. Individuals are read node by node and for each node is generated number from range 0 to 1000000. If the generated number is greater than 25000 we exchange actual node with new randomly generated. Chance to exchange is equal to 0,025. This chance is therefore so small, because there is not necessary a big frequency of the mutation. Before the exchange it is important to identify the type of the node, because we don't want to exchange operator node for variable node and vice versa. We work only with binary operator nodes, because replacing binary operator with unary operator is impossible. All these exchange limitations are necessary to preserve validity of individual.

4 Results

We create cycle to test all possible inputs for 3 variables and let it repeat 1000 times. Specification of cycle was the same as in Test 1. Program during the test successfully calculated 100% of inputs, what proofed correctness of our algorithm. In the Test 1 and the Test 2 we choose one input for three variables and one input for four variables. Those inputs have average difficulties to calculate. In the Figure 3 and the Figure 4 we can see best, average and worst fitness through all generations until we achieve correct result for those inputs.

4.1 Test 1

For this test we chose one of inputs with size of 3 variables. Final solution was found in 14 generation. All settings of this test are in the Table 1 and progress of fitness through all generations is in the Figure 3. The Figure 3 shows how the population was filled up by the algorithm with better and better individuals.

Table 1. Settings for Test 1.

Input	[true, false, false, true, false, true, false, false]
Number of generate nodes[1]	(1,50)
Population size	256
Maximum tree size	100
Mutation chance	0.025

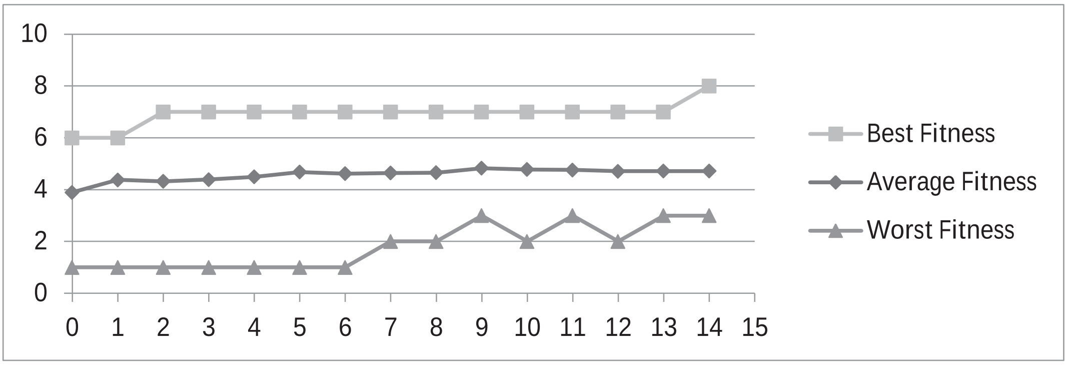


Figure 3. Progress of fitness through all generations in Test 1.

4.2 Test 2

This test was run with 4 variables input. We increase population form 256 to 512, full settings in the Table 2. Increased population give us bigger variability of generated functions at beginning of algorithm. Final solution was found in 43 generation. Full progress of fitness is in the Figure 4. Compared to the Test 1 we can see difficulty of calculation between 3 a 4 variable inputs.

[1] This value defines how many variable nodes can be generated at beginning when we created valid individuals.

5 Conclusion

In this paper we have shown how to use genetic programing for approximation of Boolean functions. We used the idea of tree structure from John R. Koza [1] and modified it for Boolean functions. We also used knowledge about Boolean functions to create specific node structure. We used these nodes in tree structures for representation of Boolean operators and variables. In our program we created method for randomly generating trees for representation of Boolean functions. We modified crossover and mutation for tree structures and specific Boolean conditions.

Our solution is good alternative to classic method of design Boolean functions such as conjunctive normal form (CNF) or disjunctive normal form (DNF). Our solution works with much larger range of Boolean operators comparing to CNF and DNF. It works perfect with Boolean functions which have up to 4 input variables. With 5 and more input variables it could be time consuming.

Table 2. Settings of Test 2.

Input	[false, true, false, false, false, true, false, false, true, true, true, false, true, false, false, false]
Number of generate nodes	(1,50)
Population size	512
Maximum tree size	100
Mutation chance	0.025

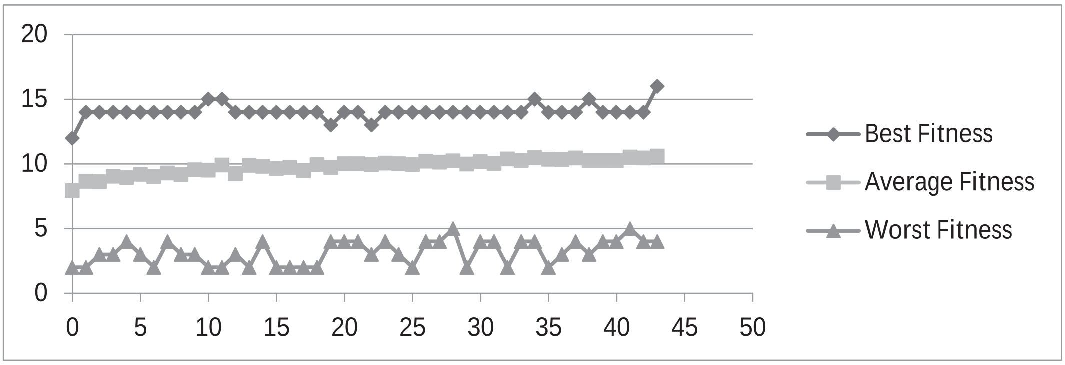


Figure 4. Progress of fitness through all generations in Test 2.

References

[1] Koza, J., R.: *Genetic Programming, On the Programming of Computers by Means of Natural Selection*, Sixth printing, MIT Press, Cambridge, 1998, pp. 17-288.

[2] Kvasnička, V., Pospíchal, J.: *Algebra a diskrétna matematika*, Slovenská technická univerzita v Bratislave, 2008, pp. 143-173.

[3] Kvasnička, V., Pospíchal, J.: *Matematická logika*, Slovenská technická univerzita v Bratislave, 2006, pp. 37-48.

[4] Kvasnička, V., Pospíchal, J., Kozák, Š., Návrat, P., Paroulek, P.: *Umelá inteligencia a kognitívna veda I*, Slovenská technická univerzita v Bratislave, 2009, pp. 335-353.

[5] Kvasnička, V., Pospíchal, J., Tiňo, P.: *Evolučné algoritmy*, Slovenská technická univerzita v Bratislave, 2000, pp. 31-76.

Interactive Evolutionary Music Composing

Matúš PIKULIAK[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
empiko@gmail.com

Abstract. This paper discusses utilization of evolutionary algorithms as subfield of artificial intelligence in creating musical compositions. We introduce our proposed interactive evolutionary algorithm for generating simple melodies. This algorithm uses comparative rating-based selection, heap-based population sorting and online interface accessible in all modern browsers.

1 Introduction

The subjectivity of art and problems that come with a difficult formal expression of it represent a complex challenge for contemporary research in informatics. There are plenty of attempts to create, perceive or criticize different artistic forms with computers, usually by the means of artificial intelligence. One of the subfields of artificial intelligence, evolutionary computing, was used in numerous systems for generating musical compositions, statues, industrial design or paintings [2, 8].

Evolutionary algorithms inspired by the concept of Darwinian evolution are based on assumption that by making small steps, we can keep achieving better and better results [1]. This improvement can be perceived as state space searching and evolutionary algorithms are considered to be heuristic searching algorithms. There are also other similarities with biological evolution: use of genotype-fenotype model, crossover of genes, sexual reproduction and others. These are all techniques used in evolutionary computing in order to improve their results and computation times.

There are preconditions for use of evolutionary algorithms in musical composing. Even human composers do not tend to write whole composition at once, there are usually lots of corrections and changes in the process that can be viewed as an evolution. Another obvious similarity is the use of genotype-fenotype model in Western music, where genotype is traditional sheet music while fenotype is the musical piece played by musicians.

In next chapter we will look at other systems using evolutionary algorithms in generating melodies. In third chapter we will present an outlook of our very own algorithm. In fourth chapter we will disscuss experiments we are carrying out with our system. And finally the fifth chapter is the conclusion of this paper.

[*] Bachelor study programme in field: Applied Informatics
Supervisor: David Chalupa, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

2 Related work

Perhaps the most well known interactive evolutionary system for composing musical elements is GenJam developed by Al Biles [3]. This system utilizes simple genotype to fenotype mapping, where every gene represents one measure. It also employs musically meaningful mutations, which means that the structure of composition will not fall apart during the evolution. Every composition consists of several phrases and every phrase consists of several measures. This forms a two-level genotype system. The songs are presented to listeners who indicate which parts of the song they like and which parts they do not like. That is how the fitness of individual phrases and whole song is computed. GenJam can also handle jazz-like improvisation with human player playing on real instrument.

There are also other ways of computing fitness function of melodies than interactive algorithms. One of them is implicit fitness function inside the system independent on musical realization. This is usually used in systems, where the evolution itself is interpreted as musical piece, rather than the results of evolution. These evolutions usually form self-organizing systems [5] with its own orders and structures. One of these systems is the birdsong ecosystem generator developed by José Fornari [6]. Individuals in this system are short birdsongs. As they are playing they influence other birdsongs in their proximity. As the time progresses there are more and more interactions between these songs and whole system is creating some kind of natural soundscape.

There are also systems that used computed generated fitness or fitness based on corpus of human composed songs. The theory of evolutionary art is also object of intensive research. There are numerous papers concerning classification of “evart” systems [7, 11] or papers dedicated to evolutionary art as an art movement with history and social consequences [9, 10].

3 Our algorithm

We decided to use an interactive evolution as a model for gaining fitness values. Fitness is assigned to individuals by human listeners. Individuals are available online on webpage of our project where users are able to listen to them and rate them[1]. This approach ensures high quality fitness calculations as human ear is still the best judge of music. The drawback of this approach is that individual ratings are very time consuming. Human listener has to physically listen to every individual before rating it while computer based music critics can rate a large amount of songs in a very short time.

Most of the interactive systems generating music are using some kind of numeric rating scale. Every individual in these systems is assigned certain amount of points, stars or something similar. This way of rating is quite demanding for human listener because he usually does not know how to precisely evaluate such a subjective object as musical piece. We chose different approach and in our system users are selecting the worst song from the triplet of samples. According to these selections we are sorting our individuals in population to heap with the worst individual on the top. We are naturally assuming that the quality of the songs is transitive property. In other words, if individual A is better than individual B and individual B is better than individual C, that mean that individual A is better than individual C.

We have also decided to create just really small populations containing only 15 individuals. They create four-level heap and it takes 3 ratings at the most to rate every generation. This way we have eliminated the time of rating of each generation to 1-3 minutes. We have also aimed at high accessibility of our system. Our system supports every modern browser and users can come and go at any time without registration or any other distracting actions. Instead of loading few people with a lot of demands, we have decided to open our system to anyone who is willing to participate in ratings.

[1] Available at *http://95.85.32.201*.

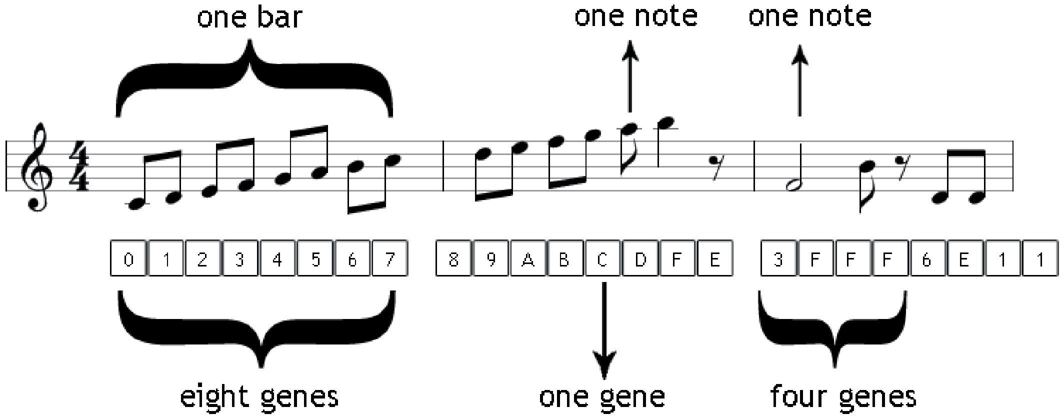


Figure 1. Genotype example.

Individuals in our evolution are musical melodies. Every melody has its own genotype that is transformed into fenotype as an mp3 file accessible for critics. Genotype is a sequence of characters from hexadecimal system (0-9, A-F) divided into groups of eight. Every group of eight represents one bar of song, while every character constitutes as one measure of this bar. The meaning of each character in genotype is as follows:

- *0-D* - one of the notes of two consequencing C-dur scale,
- *E* - pause, no sound,
- *F* - hold, the note prior to this characters is held.

Example of this genome with fenotype representation in the form of a sheet music can be seen in Figure 1.

Individuals living in our system were set to have 8 bars and are approximately 18 seconds long when played. After every rating, the worst individual on top of the heap dies and new one takes his place. This is so called steady-state model of evolution. The new individual is a child of two of the remaining individuals in the population. These parents are selected by weighted change based on their position in the heap according to the following formula:

$$f(n) = \frac{(h(n) - 1)^2}{90} \times 100\% \tag{1}$$

where *f (n)* is a chance individual *n* will be selected for reproduction and *h (n)* is a level of individual *n* in heap. These parents then undergo so called crossover creating new individual in the process. Crossover is genetic operation taking genotypes of two individuals and combining them into new genotype in a manner illustrated in Figure 2. Every new individual created by crossover then passes through a series of mutations after which final genotype is created. Mutations are random minor changes in genotype executed in an effort to bring new genetic material to evolution. Completely random mutations could however destroy structure of song created so far. That is why we have designed a set of so called musically meaningful mutations. These mutations are designed to preserve at least basic musicality of individuals. Some of these mutations are for example setting notes higher or inverting the order in which they are played.

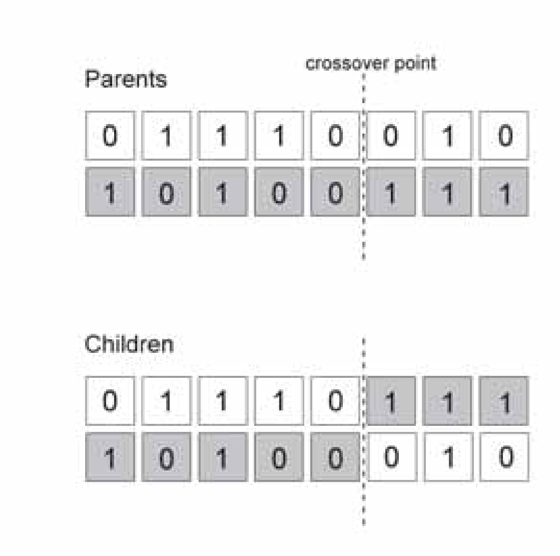


Figure 2. Illustration of crossover operation.

This new individual is then put on the top of the heap and this newly created generation is presented to some user to rate. This cycle can go on indefinitely. Successive generations are however preferably rated by the same user. This way one user's musical taste can really make an impact on an evolution and this evolution could became more unique. Basic scheme of this entire evolution can be seen in Figure 3.

We have developed our system as web application. Server side part is written in PHP using model-view-controller framework CodeIgniter while client side part is written in JavaScript. We are using SoundManager 2 JavaScript library for in-browser mp3 playing. As our database management system we have chosen MySQL. Generation of mp3 files is provided by *TiMidity++* and *FFmpeg*. Our web application is available for public basically all the time.

4 Experimental setup and further work

One of the objectives of our research is monitoring diversity in our system and how it influences quality of individuals in populations. We have implemented numerous techniques improving how the diversity of genotype is preserved as the evolution continues. We are assuming that this improvement will lead to better and more interesting results. It should also motivate users to continue to rate as the songs they will be presented will be more unique thus preventing the burnout of users in long term. These are the techniques we have implemented:

- *asymmetric crossover* [12] - changes how the crossover point works. Genotypes of two individuals have more possible ways to link to each other.

- *injection* - individuals from outside of the evolution are injected to population, if its diversity goes under critical limit.

- *inbreeding prevention* - related individuals can not have a child so their similar genotypes will not spread.

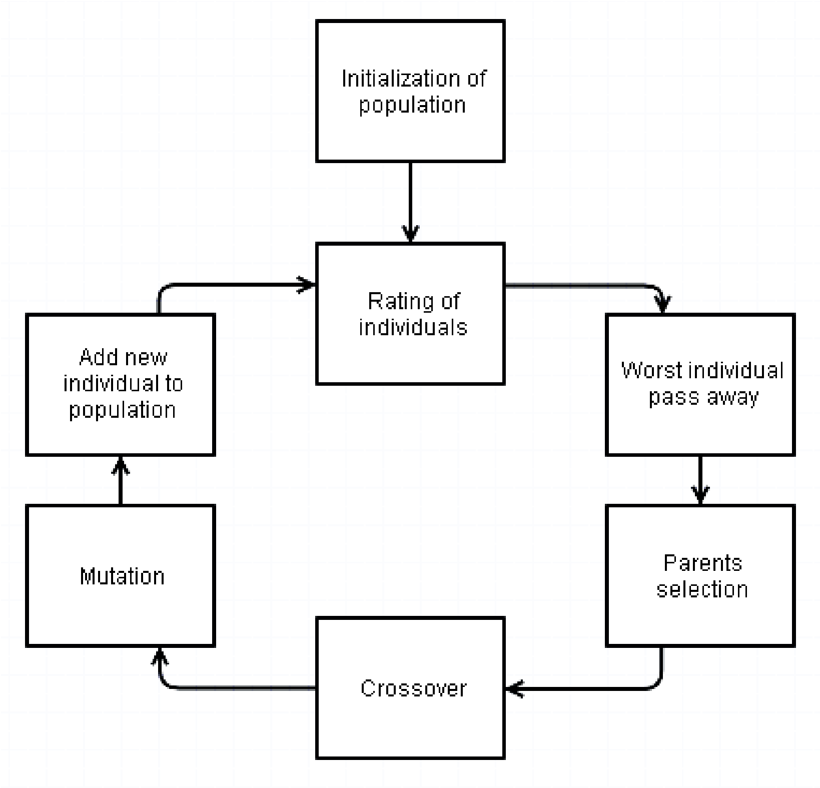

Initialization of population
Rating of individuals
Worst individual pass away
Parents selection
Crossover
Mutation
Add new individual to population

Figure 3. Scheme of our evolution.

- *clustering prevention* [4] - new individuals that are too similar to other individuals already established in population will not be included in this population.

We are planning to continue making these techniques better and experiment with different settings. We are also considering creating musically more complex compositions consisting of more melodies or introducing rhythm section. After we collect enough data, we could also attempt to build our own computer based critic based on some kind of artificial intelligence.

5 Conclusion

Evolutionary art is an interesting intersection of computer science and artistic aesthetics. Several attempts show that computer based composing using means of artificial intelligence is possible at least for simple melodies. Our system is one of those attempts and it aims to create interesting and unique songs using interactive evolutionary computing. We have utilized several improvements over previous systems, mainly in matter of ease of human interaction and reducing the time-consumption of ratings. Another dimension of our work is research of population diversity in this system. We believe that this research can help us create better and more satisfying melodies.

Acknowledgement: This contribution was supported by Grant Agency VEGA SR under the grants 1/0553/12 and 1/0458/13.

References

[1] Back, T., Fogel, D.B., Michalewicz, Z., eds.: *Handbook of Evolutionary Computation*. 1st edn. IOP Publishing Ltd., Bristol, UK, UK, 1997.

[2] Bentley, P.: *Evolutionary Design by Computers*. Evolutionary Design by Computers. Morgan Kaufman Publishers, 1999.

[3] Biles, J.: GenJam: A Genetic Algorithm for Generating Jazz Solos, 1994, pp. 131–137.

[4] Dick, G., Whigham, P.: Spatially-Structured Evolutionary Algorithms and Sharing: Do They Mix? In Wang, T.D., Li, X., Chen, S.H., Wang, X., Abbass, H., Iba, H., Chen, G.L., Yao, X., eds.: *Simulated Evolution and Learning*. Volume 4247 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 457–464.

[5] Eigenfeldt, A., Pasquier, P.: A Sonic Eco-System of Self-Organising Musical Agents. In Chio, C., Brabazon, A., Caro, G., Drechsler, R., Farooq, M., Grahl, J., Greenfield, G., Prins, C., Romero, J., Squillero, G., Tarantino, E., Tettamanzi, A., Urquhart, N., Uyar, A., eds.: *Applications of Evolutionary Computation*. Volume 6625 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 283–292.

[6] Fornari, J.: A Computational Environment for the Evolutionary Sound Synthesis of Birdsongs. In Machado, P., Romero, J., Carballal, A., eds.: *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Volume 7247 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 96–107.

[7] Johnson, C.: Fitness in Evolutionary Art and Music: What Has Been Used and What Could Be Used? In Machado, P., Romero, J., Carballal, A., eds.: *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Volume 7247 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 129–140.

[8] Lewis, M.: Evolutionary Visual Art and Design. In Romero, J., Machado, P., eds.: *The Art of Artificial Evolution*. Natural Computing Series. Springer Berlin Heidelberg, 2008, pp. 3–37.

[9] McCormack, J.: Open Problems in Evolutionary Music and Art. In Rothlauf, F., Branke, J., Cagnoni, S., Corne, D., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G., Squillero, G., eds.: *Applications of Evolutionary Computing*. Volume 3449 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 428–436.

[10] McCormack, J.: Aesthetics, Art, Evolution. In Machado, P., McDermott, J., Carballal, A., eds.: *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Volume 7834 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 1–12.

[11] Todd, P.M., Werner, G.M.: Musical Networks. MIT Press, Cambridge, MA, USA, 1999, pp. 313–339.

[12] Yuan, B.: Deterministic crowding, recombination and self-similarity. In: *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*. Volume 2., 2002, pp. 1516–1521.

Evolutionary Solution of the Game Mastermind

Metod RYBÁR*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xrybar@stuba.sk`

Abstract. Mastermind is a well-know board game in which the player must find hidden combination of colors suggested by the opponent, using signs that the opponent must supply (the number of correct guesses after each turn and the number of correctly guessed colors on the wrong positions). It is a popular combinatorial problem with dynamic constraints. This problem has already been solved using various approaches, including evolutionary heuristics. This work focuses on the analysis of existing solutions and new implementation of an existing solution, which is based on genetic algorithm.

1 Introduction

The game Mastermind was invented by Mordecai Meirowitz, Israeli post office official and telecommunication expert, in the years 1970 and 1971. The game was rejected several times by big toy manufacturers, until the February of 1971, when he presented the game on the International Toy Fair in Nurmberg to a small English company Invicta Plastics Ltd. The company bought complete rights for the game and after several small changes under the guidance of the company founder Edward Jones-Fenlegh the game has been introduced to the market in 1972. The game received several awards and sold more than 55 millions copies in 80 countries until the year 2000 [17].

The game Bulls and Cows, also known as MOO, is considered as a predecessor of the game Mastermind. It was played with pen and paper for at least a century [8]. Bulls and Cows was played with numbers, while Mastermind uses colors.

In the game there are two players, in literature most commonly known as Code Setter and Code Breaker. Code Setters objective is to create a secret code of length n using p colors. In the original game the length of a code was $n = 4$ and the number of colors was $p = 6$. Code Breaker objective is to guess the code in at most k steps. Number of possible combinations in the original version of game is $6^4 = 1296$ [17].

The game Mastermind is a puzzle. It is also an NP-complete puzzle [10]. NP-complete problem is such a problem, that belongs to the set of NP problems and at the same time to the set of NP-hard problems. Informally we can say, that an NP-hard problem is at least so hard as the hardest problem in NP [6].

* Bachelor study programme in field: Informatics
Supervisor: Professor Jiří Pospíchal, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 251–256.

In the work of J. Stuckman, et al. [20] is proved that Mastermind is an NP-complete problem. The game is taken as Mastermind Satisfiability Problem (MSP) in this way: If there is a set of guesses $G \subseteq N_l^k$, where N represent set of natural numbers representing colors, k is number of colors and l represent guess, is there at least one correct solution?

In their work they proved that problem MSP is NP-complete with respect to l. Mastermind is therefore NP-complete puzzle.

Next we examine state of the art, then the genetic algorithm on which is this work focused, we compare our new implementation with original implementation using experimental results and some conclusions are drawn in the closing section.

2 State of the art

There are many algorithms for solving Mastermind, both with and without evolutionary approaches. First we are going to look on some selected non-evolutionary approaches, then on the evolutionary solutions and at the end we compare them.

2.1 Non-evolutionary solutions

Possibly the first algorithmic solution of the game Mastermind is from 1977. It was published 5 years after introduction of the game to the market in the work of D. Knuth [11]. In his solution for 4 positions and 6 numbers the algorithm can find the solution in 5 or less steps. It uses the worst case scenario approach to eliminate non-eligible solutions.

Different approaches using MaxMin strategy and Maximal Entropy are used in the work of A. Bestavros, et al. [4]. They also introduced “dynamic” version of the game Mastermind. In this version, the Code Setter can change the secret code, but only if this change will not be in conflict with already played combinations from Code Breaker. By this change the Code Setter is becoming an active player and also the change eliminates the possibility of “random” guess from the Code Breaker.

In the MaxMin approach the Code Breaker is trying to gain maximum possible amount of information from Code Setter, while assuming the worst case scenario of amount of received information.

The algorithm using Maximal Entropy is using maximum of average amount of possible received information.

Different solution was presented in the work of B. Kooi [12]. It uses properties of probability, when guessing one element from the set, which can be well divided into different parts, like pack of cards. This algorithm basically divides the set to smaller sets, so it can in the step $r + 1$ determine the solution with probability

$$\sum_{i=0}^{m} \frac{\#(V_i)}{\#(A)} . \frac{n_i}{\#(V_i)} = \sum_{i=0}^{m} \frac{n_i}{\#(A)}$$

where the set V can be in step i divided to n_i parts, so that the probability of correctly guessing the solution in $r + 1$ step is equal to the number of parts in which set A can be divided with $r + 1$ questions, divided by size of set A.

2.2 Evolutionary solutions

Although non-evolutionary solution presented in the work of D. Knuth [11] can solve the original problem with 4 positions and 6 numbers in at most 5 steps, because it searched through the whole space, when number of colors or number of positions is increased, this problem is becoming too complicated. Therefore there were introduced several evolutionary approaches to this problem.

Evolutionary algorithm driven by entropy presented in the work of C. Cotta, et al. [7] starts like non-evolutionary algorithms with fixed guess. During the game the next step is selected from set of possible guesses found in previous step. Entropy is used as indicator of quality so, that after playing a guess the possible solutions are "rewarded" so, that the worst case scenario is avoided. There are three different implementations of this algorithm with several differences.

Algorithm using entropy described in the work of B. Kooi [12] was upgraded with evolutionary approach in the work of J. Merelo, et al. [14]. It is basically a hybrid between evolutionary and non-evolutionary algorithm. Instead of searching the whole space for solution, evolutionary solution is used to create a set of possible solutions.

Genetic approach was used in the work of L. Berghman, et al. [3]. It uses various mutations, permutations, inversions and fitness values to determine the next step and is in more detail described in the next section, since it is the subject of this paper.

2.3 Comparison of existing solutions

This comparison is using data taken from works by B. Kooi [12], L. Berghman, et al. [3], J. Merelo, et al. [15], J. Merelo, et al. [13] and C. Cotta, et al. [7]. It uses the results for the length of code $n = 4$ and number of colors $p = 6$. Bear in mind, that these data come from different works and therefore the data were taken in different conditions. It serves only as an overview of actual state.

Algorithm	Average	Maximum number of steps
D. Knuth [11]	4.478	5
A. Bestavros, et al. [4]* MaxMin	3.860	-
A. Bestavros, et al. [4] Entropy	4.415	6
B. Kooi, [12]	4.373	6
C. Cotta, et al. [7] A	4.489	-
C. Cotta, et al. [7] B	4.448	-
C. Cotta, et al. [7] C	4.425	-
J. Merelo, et al. [14]	4.410	7
L. Berghman, et al. [3]	4.390	7

*Table 1. Table shows experimental results of different algorithms. *result was produced used only with subset of possible guesses.*

As we can see in the Table 1, the only algorithm that can guarantee that it finds the solution in most 5 steps is the solution from the work of D. Knuth [11], despite that in average it is the second worst solution, after the entropy driven algorithm of type A from the work of C. Cotta, et al. [7].

If we do not consider the solution from the work of A. Bestavros, et al. [4] using MaxMin strategy, which were taken in too different conditions, the most successful algorithm is genetic algorithm from the work of L. Berghman, et al. [3], despite that it can guarantee to find the solution in maximum of 7 steps.

3 Description of the method

Algorithm suggested in the work of L. Berghman, et al. [3] is, unlike typical genetic algorithm, not playing the first possible guess after it is found, but also is evaluating their predictive values, such as it was suggested in the work of J. Merelo, et al. [14].

The algorithm uses population of size 150, which is initialized randomly, so that each individual in the population is different. Next generations are created by one and two point crossing of two parents from previous generations with probability 0.5. With probability 0.03 mutation occurs. When it occurs, one random position is changed to random color. Also with probability 0.03 occurs a permutation. When it occurs, it switches colors on two random positions. With probability 0.02 occurs inversion. When it occurs, the order of colors is changed on two different positions.

If the new created individual is already in the population, it is replaced by new one, so that all individuals in the population are random.

The probability, that the tip is selected as a parent is set by fitness value

$$f(c;i) = \sum_{q=0}^{i} |X'_q(c) - X_q(c)| + \sum_{q=0}^{i} |Y'_q(c) - Y_q(c)| + bP(i-1)$$

where $X'_q(c)$ is number of correctly guessed positions, that the guess would gain if previously played guess was the correct guess. $Y'_q(c)$ is number of correctly guessed colors in the wrong positions, that the guess would gain if previous guess was the code. $X_q(c)$ and $Y_q(c)$ contains information about correctly guessed positions and colors gained from Code Setter in previous step. P is the number of positions in code, b is a weight and i is number of played steps.

If the difference for every previous guess is 0, the guess is eligible. These guesses are added to the set of possible guesses. To select one guess maximum of $maxgen$ generation is created. They used $maxgen = 100$ and the size of the set of eligible guesses was 60.

To select a guess to play a function is used

$$f(c) = \sum_{i=0}^{n} |white(c_i) + black(c_i)|$$

where $white(c_i)$ and $black(c_i)$ is the number of correctly guessed colors and positions, if the secret code was c_i and n is the number of eligible codes.

4 Experimental Results

The algorithm described above was implemented in Java programming language. This implementation was compared with the original implementation in Borland Delphi. The experimental results showed major differences, even though the implementation is basically the same.

Language	Min guesses	Avg guesses	Median	Max guesses	Variance	St. dev.
Java	4.569	4.599	4.593	4.644	0.000479	0.0219
Borland Delphi	4.376	4.407	4.410	4.443	0.000474	0.0217

Table 2. Table shows experimental results for different maxgen sizes.

As we can see in the Table 2, Borland Delphi implementation provides better results. These results were obtained by running both implementation on the same computer 10 times for all 1296 possible code combinations and with parameters from original implementation. We can also see, that the variance and standard deviations are almost the same.

This difference in the number of guesses is most likely caused by different implementation of random number generator in their libraries. Although both use Linear congruential generator (LCG), described e.g. in the work of S. Park, M. Keith [18], there are differences in their parameters as we can see in Table 3.

Language	m	multiplier	inc	output bits of seed in rand() / Random(L)
Borland Delphi [2]	2^{32}	134775813	1	bits 63..32 of (seed * L)
Java [1]	2^{48}	25214903917	11	bits 47...16

Table 3. Table shows differences in the LCG random number generator implementation.

Next we compared different $maxgen$ sizes for Java implementation, to see how these influence the results. All other parameters were preserved from original implementation. Each configuration was repeated 10 times for all possible 1296 code combinations. We can see the results in Table 4.

From results we can see, that increase or decrease of number of generations generated when producing new population from original 100 from the work of L. Berghman, et al. [3] has in both cases negative effect on the results. Both increase and decrease in $maxgen$ caused worse results in almost all measured parameters. The only improvement was caused by decreasing the $maxgen$ to 60, when runtime for all 1296 possible combinations dropped to 7.3 minutes in average compared to 10.1 minutes. Increasing $maxgen$ to 500 meant increase of this average time to 51.1 minutes for one run of all 1296 combinations. Although they described in their work the limitation of $maxgen$ to 100 as harmful to objective function, this is not the case.

maxgen	Min guesses	Avg guesses	Median	Max guesses	Variance	St. dev.
60	4.594	4.619	4.618	4.671	0.000505	0.0225
100	4.569	4.599	4.593	4.644	0.000479	0.0220
500	4.588	4.618	4.622	4.664	0.000601	0.0245

Table 4. Table shows experimental results for different maxgen sizes.

5 Conclusion and future work

From the experimental results we can deduce, that Borland Delphi implementation provides better results, possibly because of different random number generator implementation. In the future we can try to use different random number generator library for our Java implementation to see, if the results will be better.

Also we found out, that $maxgen$ limitation to 100 is not in fact harmful to the objective function and the algorithm provides better results with this limitation than with increased number of $maxgen$. In the future work we can try to find optimal number of $maxgen$.

Though experimental runs are time consuming, there could be much promise in trying out hill climbing for various parameters, described e.g. in the work of H. Muhlenbein [16], to see, if we can obtain different results with these implementation. Because of the long time consumption of the algorithm, we should explore parallelization to compute more games at once, which can quicken the hill climbing process.

Alternative to hill climbing approach can be use of adaptive probabilities, described in the work of M. Srinivas, L. Patnaik [19]. Instead of finding best parameters to use during all games, we can modify them during the runtime.

Additional to changing the parameters of generating new generations, we can also use different approaches to select individuals to be played or to be selected as parents. A few of them are suggested in the work of D. Goldberg, K. Deb [9], including proportionate reproduction, ranking selection, tournament selection or genitor. More of them are in detail described in the work of T. Blickle, L. Thiele [5].

References

[1] Class Random, 2014, http://docs.oracle.com/javase/7/docs/api/java/util/Random.html.

[2] Arkin, B., et al.: How we learned to cheat at online poker: A study in software security, 1999.

[3] Berghman, L., Goossens, D., Leus, R.: Efficient solutions for Mastermind using genetic algorithms. *Computers & operations research*, 2009, vol. 36, no. 6, pp. 1880–1885.

[4] Bestavros, A., Belal, A.: MasterMind a game of diagnosis strategies. In: *Alexandria University*, 1986.

[5] Blickle, T., Thiele, L.: A comparison of selection schemes used in genetic algorithms, 1995.

[6] Christos, P., et al.: Computational complexity. Encyclopedia of Computer Science, 2003.

[7] Cotta, C., Guervós, J.J.M., Garćia, A.M.M., Runarsson, T.P.: Entropy-driven evolutionary approaches to the mastermind problem. In: *Parallel Problem Solving from Nature, PPSN XI*. Springer, 2010, pp. 421–431.

[8] Francis, J.: Strategies for playing MOO, or "Bulls and Cows". online, http://www.jfwaf.com-/Bulls%20and%20Cows.pdf.

[9] Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. *Urbana*, 1991, vol. 51, pp. 61801–2996.

[10] Kendall, G., Parkes, A.J., Spoerer, K.: A Survey of NP-Complete Puzzles. *ICGA Journal*, 2008, vol. 31, no. 1, pp. 13–34.

[11] Knuth, D.E.: The computer as master mind. *Journal of Recreational Mathematics*, 1976, vol. 9, no. 1, pp. 1–6.

[12] Kooi, B.P.: Yet Another Mastermind Strategy. *ICGA Journal*, 2005, vol. 28, no. 1, pp. 13–20.

[13] Merelo, J.J., Mora, A.M., Runarsson, T.P., Cotta, C.: Assessing efficiency of different evolutionary strategies playing mastermind. In: *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, IEEE, 2010, pp. 38–45.

[14] Merelo-Guervós, J., Castillo, P., Rivas, V.: Finding a needle in a haystack using hints and evolutionary computation: the case of evolutionary MasterMind. *Applied Soft Computing*, 2006, vol. 6, no. 2, pp. 170–179.

[15] Merelo-Guervós, J.J., Runarsson, T.P.: Finding better solutions to the mastermind puzzle using evolutionary algorithms. In: *Applications of Evolutionary Computation*. Springer, 2010, pp. 121–130.

[16] Mühlenbein, H.: How Genetic Algorithms Really Work: Mutation and Hillclimbing. In: *PPSN*. Volume 92., 1992, pp. 15–25.

[17] Nelson, T.: A Brief History of the Master Mind Board Game, 2014.

[18] Park, S.K., Miller, K.W.: Random number generators: good ones are hard to find. *Communications of the ACM*, 1988, vol. 31, no. 10, pp. 1192–1201.

[19] Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 1994, vol. 24, no. 4, pp. 656–667.

[20] Stuckman, J., Zhang, G.Q.: Mastermind is NP-complete. *INFOCOMP Journal of Computer Science*, 2006, vol. 5, p. 25–28.

Use of Biologically Inspired Algorithms for DNA Assembly

Štefan KASALA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`stefan.kasala@gmail.com`

Abstract. In Informatics we often come across problems, whose solutions are not possible with common algorithms. Inspiration for algorithms may come from nature. To this group of algorithms belongs also these inspired in social animals, eg. ants, bees etc. They are often simple organisms, but precise in reaching their common goal. Today shotgun sequencing technique allows to gather random small reads of DNA very fast. Reverse process, DNA assembly, is one of these complex problems. We made survey of nowaday assembly techniques and also application of BIA in this field. We proposed possible solution using firefly algorithm. We researched and designed new operators for algorithm and adjusted it for discrete problems. Finally, we implemented algorithm and tested it on Asymmetric Traveling Salesman Problem instances, which are suitable abstraction of DNA assembly problem.

1 Introduction

Analysis and data processing is often used in fields between informatics and other scientific discipline. One of the most recent problems in the field of bioinformatics is a DNA assembly. Knowledge of a DNA structure is used in prevention from many illnesses and their cure. Thanks to progress in technique, it is possible to gain huge amount of very small DNA fragment, faster and faster, using "shotgun sequencing" [9]. The problem is, that we do not know the origin location of fragments in the DNA sequence. To solve this computationally hard problem, many algorithms were designed. Their main goal is to find out the origin structure of DNA.

One of the possible solutions can use biologically inspired algorithms (BIA). BIA are stochastic search and optimization techniques based on principles of collective behavior and self-organization. They are inspired by social behavior of live organisms. They were successfully applied in many fields of computationally hard problems, where solutions using conventional algorithms are not possible. DNA assembly is such a problem.

[*] Master study programme in field: Software Engineering
Supervisor: Dr. Anna Bou Ezzeddine, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 257–262.

2 Problem definition

The main problem of DNA assembly consist of huge amount of small fragments on input side and origin DNA sequence, which is permutation of inputs, on output side. It is a combinatorial NP-hard class problem [11].

The nowadays technology (sequencing of second generation, shotgun sequencing) is able to read very fast in parallel DNA sequences, but for price of short reads (100 base pairs and less) of random location. To read the whole DNA we need to reach proper coverage DNA sequence by reads [11].

Problem can be defined as shortest common super-string, euler super-path [10], approach using string graphs. All the problem definitions are proofed to be O (NP). Some technologies need special pair read data.

The algorithms have to deal with additional complications. DNA structure consist of two complementary strings and the reads came from either of them. In organism genome repeats of variable length and frequency are common. Read process is not error proof, its output can contain various errors of certain probability, that needs to be detected and repaired. We can sum the main problems [5]. (1) unknown direction - overlaps between two reads depends on their order, (2) read errors - current technology is not error proof, 1 to 10 % error rate is common, (3) insufficient coverage - reads are random, the genome is covered only in some probability, (4) repeats - repeats bigger than reads cannot be detected, (5) chimeras and contamination.

3 Related work

3.1 Standard approaches

Vast majority of assemblers are based on 2 standard approaches [8]. Overlap-layout-consensus method was used in Sanger project. It is based on principle that suffix of one fragment is prefix of another. After that fragments are ordered to layout. Phase consensus means that reads in layout are aligned, decisions are made in case of conflicts are found in aligned columns. Conflicts can be caused either by incorrect layout or error in sequencing phase. Tools like `Newbler` `Celera Assembler` are based on this approach.

The most modern assemblers are based on De Bruijn Graphs [9]. Each node represents unique string of k length, which can be found in some input sequence, or its complement. Oriented edge links two nodes “aA” and “Ab” in case that string “aAb” can be found in origin sequence. DNA assembly is the shortest path of graph that consists of all the nodes. For double strand character of DNA the bi-directed graph is used. The tools based on this principle are for example Euler, Velvet, ABySS, AllPaths, and SOAPdenovo. The main differences between tools are internal memory representation of data, dealing with pair reads and dealing with read errors.

3.2 BIA approaches

Optimization in DNA assembly, increasing the quality of assembly output leads to use of alternate methods such as BIA approaches [3]. One group of BIA are algorithms inspired by swarm intelligence.

In [2] authors combined 2 BIA swarm algorithms, artificial bee colony (ABC) and Queen Bee Evolution Based on Genetic Algorithm (QUEGA). They successfully applied algorithms for assembly of errorless data and data with some rate of artificial errors. The food in ABC and individual in QUEGA was one solution represented by sequence permutation. The set of DNA fragments on input, the optimization problem was to minimize number of contigs and maximize the overlap score of permutation. Algorithms did not need any data preprocessing. The authors used problem aware local search (PALS). The authors used GenFrag and MetaSim for data generation. They observed comparable results for errorless data with another BIA approaches. For data with some degree of errors they observed better performance with QUEGA.

Another popular BIA optimization is based on ants behavior. It is often demonstrated on traveling salesman problem, that is also abstraction of DNA assembly. Work [7] applied the ant colony optimization (ACO) on the DNA assembly problem. The cities are fragments and the city distances are analogy to the fragment similarity. The goal is to travel the shortest path over all the cities, in DNA assembly world find the shortest string of all the fragments. The algorithm was tested on the subparts of human genome, cut to fragments with no errors. It performed better if numerous contigs were composed.

3.3 Firefly algorithm

For our problem solution we chose algorithm inspired by fireflies - Firefly algorithm (FA). Algorithm was first published in 2009 in [12] by his creator Xin-She Yang. Algorithm is inspired by attracting of fireflies by their lights. In real life males and females are attracted by lights for purpose of reproduction. The main principles are [12]:

- the fireflies are bisexual,
- attractiveness is proportional to firefly light intensity, the less lighter moves always to the more lighter firefly,
- the light intensity decreases proportional to distance between two flies, caused by light absorption,
- light intensity is formed by character of objective function. Objective function is similar to fitness function, which is metric for solution quality in genetic algorithms.

Pseudocode for general algorithm [12]:

```
1. initial parameters are defined: fireflies count, number of moves $M$, iteration limit $N$, light absorption $gamma$
2. generate initial population $P$
3. repeat $N$ times or until desired solution achieved:
    a. for every firefly $F$ in population $P$:
        i. find the most attractive $F_A$,
        ii. if more attractive not visible for defined $gamma$, move random $M$ times,
        iii. otherwise move to the more attractive $M$ times.
    b. evaluate new flies based on their lightness, choose fireflies for next iteration.
```

Each firefly is one problem solution, in our case a permutation of the fragments. The distance between fireflies is measure of difference between fireflies. Light absorption constant, usually from 0.01 to 100 is characteristic of how far can firefly see in the solutions space. If set too low, firefly can see all the other fireflies and algorithms is similar to particle swarm optimization. If set to high value, fireflies are blind and algorithms become random search.

Algorithm main idea is to improve solution with similar better solution. Author claims that it finds extremes in effective way, also with global optima between them. It performs better and converges faster compared to genetic algorithm and PSO. The attraction in distance r is defined as [12]:

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \tag{1}$$

where r - distance, B_0 - lightness of firefly = fitness, γ - light absorption constant, optional number (mostly between $0, 01$ and 100). Algorithm is primary designed for continues optimization problems, its performance was illustrated on Michalewitz function for 2 independent variables. It needs to be modify to meet combinatorial problem requirements. The possible solution was illustrated in [4]. The movement is the abstraction of solution change. In our solution we designed components of algorithm as follows.

3.4 Algorithm modifications

Firefly
Firefly is permutation of input DNA fragments, each fragment has flag for strand orientation.

Movement
Movement should express various length and direction. We designed two algorithms. First is pure random movement. We take random number N from uniform distribution of distance between 2 fireflies. Then the moving firefly is N times randomly changed. Second movement, better expressing problem characteristic is sequence constructed movement (SCM). It is inspired by SCX operator used in genetic algorithm for combinatorial problems [1]. It tries to compose new, better permutation by adding the subsequences of the firefly we are moving to. Repeatedly application of this movement function expresses the various length of the movement.

Distance
For distance between the firefly A and the firefly B we designed 2 algorithms. First is one to one compare of permutations, distance is percentage of object on different positions to all the positions in permutation. Second uses relative position of fragments in permutation. It looks at the predecessor of compared permutation objects. If they are not the same, the counter is incremented. The result is also in percentage.

Attractiveness
We are completely satisfied with original proposed mathematic model in Equation 1. The firefly quality, or better its light intensity in distance zero, is the sum of overlaps over all the fragments in a permutation. Another interesting model from [6] is based also on overlaps but penalizes permutations where two good overlaps are far from each other.

Selection
We use two simple methods, one chooses best solutions only from new generated fireflies, another chooses best solutions from union of old and new fireflies. For quality of solution we use the same function as for attractiveness.

4 Experiments and Results

The experiments were performed on freely available Asymmetric Traveling Salesman (ATSP) instances, instances of TSP where distance from city A to B are not equal to distance from city B to A. The solution of the problem is to find shortest path through all the cities[1]. Experiment on Picture 1

[1] `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`

Table 1. Parameters values for executed experiments.

Parameter	**Value**
GAMMA	0.15
Number of moves	5
Population size	15
Iterations	200
Maximum random movement	20

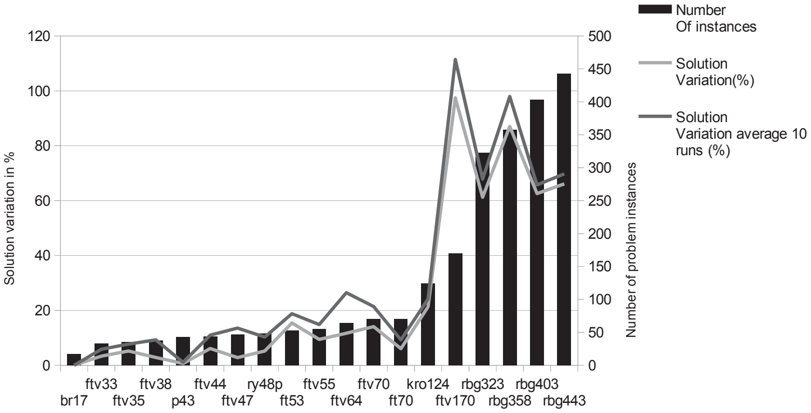


Figure 1. Results for ATSP instances.

shows results for all available ATSP instances. Experiment were run with attributes in Table 1, only new fireflies selection and SCM movement. Black columns show the difficulty of problem in number of cities. Darker gray line shows our results (best result from 10 runs), lighter gray average, both in percentage differ from best known solution.

As we can see algorithm performs reasonably well for smaller problems in this experiment till 124 instances. For bigger instances the results get significantly worse. Interesting is that further increasing of the problem did not change the variation of our results.

Experiment on Picture 2 shows best solution evolution dependent on iteration. The solution is mostly improved in first iterations, but it did not converge and is continuously improved until limit for max. iterations is reached.

5 Conclusion and Future Work

In this paper we described new approach for DNA assembly based on firefly algorithm. The algorithm, metaheuristic, shows interesting optimization possibilities. We designed new algorithm operators and evaluated them on asymmetric TSP instances. ATSP is suitable abstraction of DNA assembly and can be used for algorithm benchmarks.

Further fine tuning and improving can be done. The effectivity can be improved by further research and combination of new operators. In future work, it is important to evaluate algorithm and experiment on real DNA sequences. For test data we plan to use DNAgen and GenFrag instances from DNA, which are common used and can be compared with other published results.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0752/14.

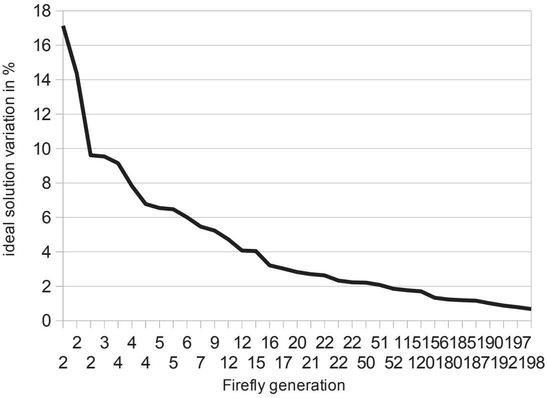


Figure 2. Convergence of algorithm for "kro124" instance.

References

[1] Ahmed, Z.: Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. *International Journal of Biometrics . . .* , 2010, vol. 3, no. 6, pp. 96–105.

[2] Firoz, J.S., Rahman, M.S., Saha, T.K.: Bee Algorithms for Solving DNA Fragment Assembly Problem with Noisy and Noiseless Data. In: *Proceedings of the Fourteenth International Conference on Genetic....* GECCO '12, New York, NY, USA, ACM, 2012, pp. 201–208.

[3] Indumathy, R., Maheswari, S.U.: Nature inspired algorithms to solve DNA fragment assembly problem: A Survey. *International Journal on Bioinformatics . . .* , 2012, vol. 2, no. 2, pp. 45–50.

[4] Jati, G., Suyanto: Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem. In Bouchachia, A., ed.: *Adaptive and Intelligent Systems*. Volume 6943 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 393–403.

[5] Li, L., Khuri, S.: A comparison of DNA fragment assembly algorithms. In: *Proc. of the Int'l Conf. on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, CSREA Press, 2004, pp. 329–335.

[6] Luque, G., Alba, E.: Metaheuristics for the DNA Fragment Assembly Problem. *International Journal of Computational Intelligence Research*, 2005, vol. 1, no. 2, pp. 98–108.

[7] Meksangsouy, P., Chaiyaratana, N.: DNA fragment assembly using an ant colony system algorithm. *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, 2003, vol. 3, no. C, pp. 1756–1763.

[8] Miller, J.R., Koren, S., Sutton, G.: Assembly algorithms for next-generation sequencing data. *Genomics*, 2010, vol. 95, no. 6, pp. 315–27.

[9] Pevzner, P.a., Tang, H.: Fragment assembly with double-barreled data. *Bioinformatics (Oxford, England)*, 2001, vol. 17 Suppl 1, pp. S225–33.

[10] Pevzner, P.a., Tang, H., Waterman, M.S.: An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 2001, vol. 98, no. 17, pp. 9748–53.

[11] Schatz, M.C., Delcher, A.L., Salzberg, S.L.: Assembly of large genomes using second-generation sequencing. *Genome research*, 2010, vol. 20, no. 9, pp. 1165–73.

[12] Yang, X.S.: Firefly Algorithms for Multimodal Optimization. In: *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications*. SAGA'09, Berlin, Heidelberg, Springer-Verlag, 2009, pp. 169–178.

A Comparison of Traditional and Swarm Based Clustering

Adrián KOLLÁR*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xkollara2@fiit.stuba.sk`

Abstract. An increase in both the volume and the variety of data brings more possibilities to study data, their meaning, relationships and structure. Clustering represents one method, which can be used for data analysis. It can organize data objects into groups of similar objects, called clusters. We carried out an experiment with two groups of clustering algorithms. The first group consists of two traditional and well known clustering algorithms, K-means and Hierarchical Agglomerative Clustering (HAC). The second group consists of algorithms inspired by meta-heuristic optimization techniques, ACO („Ant Colony Optimization") and PSO („Particle Swarm Optimization"). These algorithms are both population based and can be easily parallelized. The quality of the resulting clusters was compared by means of internal and external indices.

1 Introduction

Clustering, also known as unsupervised classification is a method of creating groups of objects, or clusters, in such a way that objects from one cluster are very similar and objects from different clusters are distinct. It is the main task of exploratory data mining useful in various fields, including pattern recognition, machine learning, image analysis, information retrieval, document retrieval and image segmentation [6].

Since clustering is used in a variety of areas and for many different types of data, large number of different clustering methods exists. Recent research made possible to view clustering as an optimization problem, due to which many new algorithms emerged based on meta-heuristic techniques [1]. Meta-heuristic is a generic method, which can be used on many optimization problems. Many of these methods are also known as biologically or nature inspired, or swarm based (or swarm intelligence) methods. These methods include Ant Colony Optimization Algorithm [5], Artificial Bee Colony Algorithm, Particle Swarm Optimization [1], Bird Flocking Algorithm, and Frog Leaping Algorithm which were proven to solve complex optimization problems like the Travelling Salesman Problem, Quadratic Assignment Problem and Graph Coloring Problem [9].

[*] Master degree study programme in field: Software Engineering
Supervisor: Assoc. Professor Mária Lucká, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

In this paper, we provide comparison of two traditional (K-means and HAC) and two swarm based (ACA and PSO) clustering algorithms. Experimental results were gathered after clustering data from three different datasets. Quality of the resulting clusters was evaluated by means of internal and external clustering indices.

2 Clustering algorithms

Hard clustering algorithms can be divided into two main groups of clustering algorithms: partitional and hierarchical. A partitional algorithm divides a dataset into a single partition, whereas a hierarchical algorithm divides a dataset into a sequence of nested partitions [6].

2.1 K-means

K-means was first published in 1955. In spite of the fact that K-means was proposed over 50 years ago and thousands of clustering algorithms have been published since then, K-means is still widely used [8]. It is a well known algorithm, so details of the algorithm are not presented in this paper.

The advantage of K-means is the relative simplicity of algorithm. The main disadvantages are: a) it can create only ellipsoid clusters, b) it heavily depends on initialization, c) necessity to specify number of clusters - parameter K.

2.2 Hierarchical Agglomerative Clustering (HAC)

There are two basic approaches to generating a hierarchical clustering [11]:

1. Agglomerative: Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.
2. Divisive: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

In this paper we chose the agglomerative clustering since it is a more frequently used approach than the divisive one.

An advantage of HAC is the ability of creating complex-shaped clusters. A disadvantage is worse scaling with large datasets. It is also very dependent on cluster distance measure. When the cluster distance is changed, clustering can produce very different results.

2.3 Ant Colony Algorithm (ACA)

Ant Colony Algorithm (ACA) [1] is a clustering algorithm inspired by the behaviour of ants during cleaning their nests. Ants are modelled as simple agents which are randomly moving in their environment. The environment is in a low dimensional space, usually it is a two-dimensional (toroidal) grid.

The first step of the algorithm is a random distribution of data objects in the two-dimensional grid; each data object represents an object from dataset. Ants will then start to move randomly within the grid. Ants can pick up objects from the grid, move them and eventually drop them with a certain probability. Probability of picking an object increases, if dissimilar objects are nearby in the environment. Probability of dropping object increases, if similar objects are located in the neighbourhood. Eventually, groups of similar objects are created in the grid environment.

Advantage of ACA is the ability to create clusters of complex shapes. In contrast to other clustering algorithms, ACA does not need a number of clusters as input parameter.

2.4 PSO Clustering (PSO)

In PSO, a population of conceptual particles is initialized. Each particle represents a solution of a clustering problem; more specifically it is a set of K centroids. Each particle is initialized with a random position and velocity. Positions and velocities are adjusted iteratively, and solutions represented by particles are evaluated in each iteration. Objective of PSO clustering is to find an optimal set of centroids.

The deterministic search used by K-means, always converges to the nearest local optimum depending on the initialization. In contrast, PSO has an important advantage over K-means; it is the ability to overcome a local optimum, due to the combination and comparison of several possible solutions simultaneously. This is an advantage over other algorithms which use local heuristics as well, such as simulated annealing [1].

3 Datasets

We used 3 datasets in the experiment. The first two datasets, Iris and Breast Cancer Wisconsin, are frequently used for classification and clustering tasks. They are available at UCI Machine Learning repository [2].

The third dataset, Reuters R8, consists of large number of documents. Text pre-processing and vector creation were necessary steps in order to perform clustering. Reuters R8 is available online [3].

3.1 Iris

The Iris dataset contains 150 instances of three kinds of Iris plants, each instance contains four attributes. Distribution of instances is uniform - each type of plant is represented by 50 instances. One type (Iris Setosa) is linearly separable from the other two, but the other two types are not mutually linearly separable (Iris Versicolor and Iris Virginica). Visualization of Iris dataset is in Figure 1.

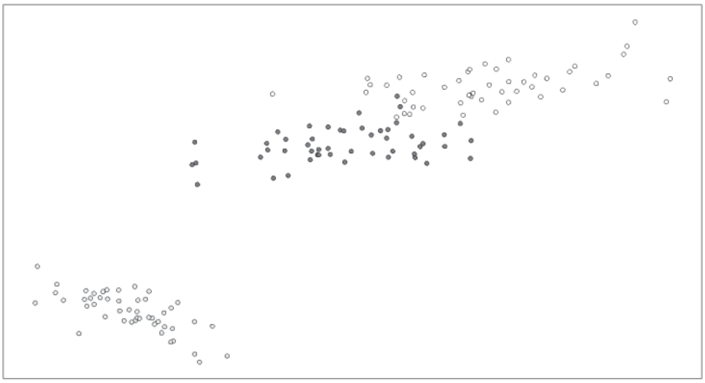


Figure 1. Iris dataset visualization.

3.2 Breast Cancer Wisconsin

The Breast Cancer Wisconsin dataset contains 569 instances, each instance has 32 attributes. They can be classified into 2 classes. One class represents malignant and the other one benign tumours. Visualization of the dataset is in Figure 2.

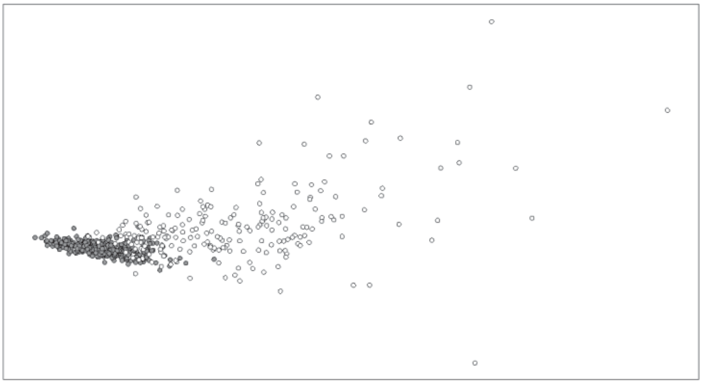


Figure 2. Breast dataset visualization.

3.3 Reuters R8

Reuters R8 is a subset of Reuters-21578, which is a collection of documents for text categorization. Reuters R8 removed documents which did not belong to any topic, or had more than one topic. Reuters R8 contains 7674 documents which can be classified into 8 topics. All documents from R8 dataset can be assigned exclusively to one topic. It is therefore a suitable dataset for hard clustering algorithms.

Text pre-processing was required in order to perform the experiment. Pre-processing steps include stop words removal, stemming, unique words identification and vector creation. TF-IDF method was used for creating vectors [11].

4 Evaluation of cluster quality

Two different measures exist for evaluation of cluster quality (sometimes also referred to as cluster validity).

One type allows us to compare different sets of clusters without any prior knowledge about the data [4]. This type of evaluation is known as internal validation. In this paper, we used Davis-Bouldin [10] and Silhoutte [10] indices to evaluate clustering quality.

The other type of evaluation uses external information about data therefore it is also known as external validation. It enables evaluation of clustering quality by comparing the produced groups to known classes. In this paper Rand index [10] was used to perform cluster validation.

5 Results

We used Euclidian distance [7] as a distance measure for all algorithms. HAC used average link [12] as a cluster distance measure. The number of clusters was specified for K-means, HAC and PSO algorithms. All algorithms were executed several times and mean values are summarized in tables 1-3. Values in bold represent the best result, underlined values represent the second best result.

Table 1. Experiment results for Iris dataset.

	K-means	HAC	ACA	PSO
DB Index	0.662	<u>0.659</u>	0.670	**0.654**
Silhoutte Index	0.553	<u>0.554</u>	0.551	**0.555**
Rand Index	0.929	**0.938**	0.913	<u>0.933</u>

Table 2. Experiment results for Breast dataset.

	K-means	HAC	ACA	PSO
DB Index	0.504	**0.429**	0.584	0.615
Silhoutte Index	**0.697**	0.691	0.602	0.603
Rand Index	0.854	0.663	0.709	**0.910**

Table 3. Experiment results for R8 dataset.

	K-means	HAC	ACA	PSO
DB Index	**3.729**	3.799	4.189	4.639
Silhoutte Index	**0.069**	0.046	0.056	0.031
Rand Index	0.602	**0.821**	0.587	0.689

The results proved good clustering of the PSO algorithm. Despite the fact that the values of internal indices on Wisconsin Breast Cancer dataset were slightly worse, the external index showed significantly better results.

The results of ACA clustering were very dependent on parameter settings. Fine tuning the parameters, especially parameter α, may improve the results of ACA clustering.

6 Conclusions

In terms of clustering quality, swarm based clustering algorithms proved that they can be equally good as traditional clustering algorithms. Moreover, swarm based clustering techniques have several significant advantages. All of the swarm based algorithms may be parallelized easily. Furthermore, their convergence can be faster with optimal parameter values. They have also potential for further improvements and they can be used for different types of data.

Future work will focus on the improvement of convergence rates on large datasets. Parallelization of swarm based clustering algorithms will be conducted. Vector operations (e.g. similarity and dissimilarity measures) will be parallelized too. Suitability of both CPU and GPU parallelization for individual algorithms will be analyzed and eventually implemented.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0752/14.

References

[1] Abraham, A., Das, S., Roy, S.: Swarm intelligence algorithms for data clustering. In: Soft Computing for Knowledge Discovery and Data Mining. Springer US, (2008), pp. 279-313.

[2] Bache, K.; Lichman, M. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. [online, accessed February 20, 2014]. Available at: http://archive.ics.uci.edu/ml.

[3] Datasets for single-label text categorization [online, accessed February 20, 2014]. Available at: http://web.ist.utl.pt/acardoso/datasets/.

[4] Deborah L. J., Baskaran R., Kannan A.: A survey on internal validity measure for cluster validation. In: International Journal of Computer Science & Engineering Survey (IJCSES), (2010), vol. 1, no. 2, pp. 85-102.

[5] Dorigo M., Stützle T.: Ant Colony Optimization. Cambridge: MIT Press, (2004), ISBN: 0-262-04219-3.

[6] Gan G., Chaoqun M., Jianhong W.: Data Clustering: Theory, Algorithms, and Applications. Philadelphia: ASA-SIAM, (2007), ISBN: 978-0-898716-23-8.

[7] Huang, A.: Similarity measures for text document clustering. In: Proceedings of NZCSRSC, (2008), pp. 49-56.

[8] Jain A. K.: Data clustering: 50 years beyond K-means. In: Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, (2008), pp. 3-4.

[9] Priya V., Natarajan A. M., Raja M.: Ants for Document Clustering. In: International Journal of Computer Science Issues, vol. 9. Interscience Open Access Journals, (2012), no. 2, p. 493-499.

[10] Rendón E., et. al.: Internal versus External cluster validation indexes. In: International Journal of computers and communications, (2011), vol. 5, no. 1, pp. 27-34.

[11] Steinbach M., Karypis G., Kumar V.: A comparison of document clustering techniques. In: KDD workshop on text mining, (2000), p. 525-526.

[12] Xu R., Wunsch D. C.: Clustering. New Jersey: John Wiley & Sons, Inc., (2009), ISBN: 978-0-470-27680-8.

Characteristics of Small World Networks

Šimon KOMPAS*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
kompas.simon@gmail.com

Abstract. A small world network refers to a family of networks in which the mean distance between vertices increases sufficiently slowly as a function of the number of vertices in the network. This property has many real world networks such as road maps, electrical power grid, Internet, neural networks, diseases spreading networks and social networks. In this paper we study methods of analysing small world specific characteristics, particularly average path length (APL) and clustering coefficient (CC). In addition, small world networks and their related network types, namely random graphs and scale-free networks were studied. While examining the characteristics of several real world networks, the values of average path length were inaccurate, since the studied graphs were disconnected. Thus we generalized the computation formula of the characteristics for disconnected graphs. Small world specific characteristics were applied to analyse chosen real graphs.

1 Introduction

Scientific community has started to address the small world phenomenon in the 20th century, when similar network properties were observed across different types of networks, such as social networks, biology networks, neurons or diseases spreading networks, power grids networks, WWW network, or Internet network.

For the first time, the small world phenomenon was analysed in 1998 [9]. Until then, the networks and dynamical systems were modeled using random graphs or regular lattices, which represent two opposite approaches in network modeling. Nevertheless, a variety of real world networks have the properties and characteristics that are somewhere between those two extremes. Watts and Strogatz tried to interpolate between these kinds of networks. This conjunction created the first network model that respects the well-known phenomenon. This network model shared with real networks their specific characteristics, namely low APL comparable to the value of the same characteristic in random graph with same size and high CC. These two characteristics define the small world networks.

[*] Master degree study programme in field Information Systems
Supervisor: Dr. Gabriela Kosková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 269–274.

2 Network characteristics

Graphs and networks are defined by their distinctive properties. These properties are expressed by characteristics. In this section we introduce them briefly.

2.1 Average path length

The APL is a measure of the information transfer effectiveness in the network. This characteristic is defined as the average of all shortest possible paths between all possible pairs of vertices:

$$l = \frac{1}{n*(n-1)} \sum_{i,j} d(v_i, v_j) \tag{1}$$

where n is count of all vertices and $d(v_i, v_j)$ is function of the shortest path length between vertices v_i and v_j.This method of measurement is only applicable to connected kind of networks. For disconnected networks was established other computation formula.

For disconnected networks a slightly different type of computation of shortest path length between two vertices in graph was used. In graph theory the shortest path length between two unreachable vertices v_i and v_j equals infinity, $d(v_i, v_j) = \infty$. This setting makes computation formula (1) inapplicable to disconnected types of networks. Therefore in real applications there was used following formula of computing the shortest path length between two vertices v_i and v_j:

$$d(v_i, v_j) = \begin{cases} 0 & if\ path\ between\ nodes\ v_i, v_j\ does\ not\ exist \\ |v_i, v_j| & if\ path\ between\ nodes\ v_i, v_j\ does\ exist \end{cases} \tag{2}$$

where $|v_i, v_j|$ is path with the lowest count of edges between vertices v_i and v_j. With this slightly modification of shortest path length function is computational formula of APL (1) applicable to disconnected networks.

The most widely used method of computation is based on the other principle. Despite of that, this method uses also basic formula of computation (1). The principle is based on idea that the best approximation of this characteristics in disconnected networks can be simply made by computing the APL only for the largest component[1] of the graph. Every component is connected at every time. Because of this fact (1) can be applied to disconnected networks.

2.2 Clustering coefficient

In graph theory, the CC is a characteristic that measures clustering of vertices at local significance. High values of CC have been observed in real world networks, especially in social networks. The origin of this characteristic is associated to small world phenomenon.

The first attempt to create a method measuring the CC is described in [7]. This method can be applied to unweighted graphs and determines the characteristics of the network in the global context. Firstly, it is necessary to define an open triplet as a triplet of vertices connected by two edges and closed triplet as triplet of vertices connected by exactly three edges. Then CC is:

$$C = \frac{t_c}{t_c + t_o} \tag{3}$$

where t_c is count of all closed triplets and t_o is count of all open triplets in the graph.

In [9], the authors define new formula for calculating CC, which is applicable to unweighted undirected connected graph G. This formula has been generalized also to directed graphs. The algorithm of computation consists of two main parts. Firstly, local CC for each vertex of the graph is computed. Secondly, global CC for the entire graph is computed. To calculate the local CC the neighbourhood of vertex v_i is defined as a subgraph G_i whose elements are all vertices that have a

[1] Largest component of graph is component with the largest number of vertices.

common edge with vertex v_i. Proportion of the real number of edges in the subgraph to the maximum possible determines the value of local CC as following:

$$C_i = \frac{|E_i|}{M_i} \tag{4}$$

where $|E_i|$ is a count of all edges in subgraph G_i and M_i is defined as count of all possible edges between every pair of vertices in subgraph G_i.

Then global CC is a mean of all local CCs in graph:

$$C = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{5}$$

3 Network types

In this section we define the random graphs that were used as a basis for the creation of small world networks. Subsequently, we define the small world networks. Afterwards we introduce the scale-free networks because of the similarity of characteristics that defines small world networks.

3.1 Random graphs

Random graph is a graph created by random processes. Random could be either total number of vertices, edges, or method of vertex connection. In contrast to other types of networks, they are not defined as a set of vertices and edges, but as a number of vertices, edges or probability of edge connection. The first random graph was defined by Paul Erdös and Alfred Renyi in joint works [4, 5]. In the same year E.N. Gilbert defined another random graph model [6]. Both models mentioned above are defined for unweighted undirected graphs.

In models of random graphs, a specific common property was observed – low APL between pairs of vertices. The APL in random networks was derived to following formula [1]:

$$l_{random} \sim \frac{ln(n)}{ln(\langle k \rangle)} \tag{6}$$

where n is a number of vertices and $\langle k \rangle$ is mean vertex degree[2] of graph. Low value of APL was leading to the assumption that the real world networks are showing signs of a small world and thus should be modeled by random graphs. But this assumption was not correct, because random graphs do not show another important property of small world networks which is clustering. For random networks, simplified formula for approximate calculation of CC was proposed [1]:

$$C_{random} = \frac{\langle k \rangle}{n} \tag{7}$$

Even though the random networks cannot be used to model small world networks, they are useful to study, since they share the specific property of low APL [1] with small world networks.

3.2 Small world networks

Small world networks represent a network type which is characterized by low APL according to small world phenomenon and high CC. Also it can be defined as a network that has the APL comparable to random graph of the same size [10].

$$l_{swn} \approx l_{random} \propto ln(n) \tag{8}$$

Unlike random graphs, small world networks capture the clustering property. CC of the small world networks is in comparison to random graphs in order of magnitude greater [10], i.e.

$$C_{swn} \gg C_{random} \tag{9}$$

[2] Vertex degree is defined as count of all edges connected to vertex

Models of the small world networks also use dynamic processes to create resulting graph. Advanced models use in their processes a technique of vertex preferential attachment. This technique is based on the principle in which the newly added vertex is most likely attached to the vertex with the highest count of edges. Preferential attachment is corresponding to behaviour of real world networks and therefore the technique was taken over to advanced network types such as scale-free networks.

3.3 Scale-free networks

The network is scale-free when the vertex degree distribution asymptotically follows power-law probability distribution.

$$P(k) \sim k^{-\gamma} \tag{10}$$

where γ is scale coefficient which is typically bounded as $2 < \gamma < 3$. Scale-free property in graph theory expresses the network which maintains the value of characteristics independent of size.

Derived formula of APL shows an ultra-small[3] character of scale-free networks [3]:

$$l_{sfn} \sim \frac{ln(n)}{ln(ln(n))} \tag{11}$$

CC formula for large scale-free networks has been approximated as following:

$$C_{sfn} \sim n^{-0.75} \tag{12}$$

Because of similarities to small world networks are scale-free networks often regarded as ultra-small world networks.

Models of scale-free networks are based on two principles: preferential attachment and vertex addition during modeling. Case studies [2] demonstrated that these two principles cannot independently produce scale-free network. Therefore, models of scale-free networks are parameterised by time and not size like in previously described network types.

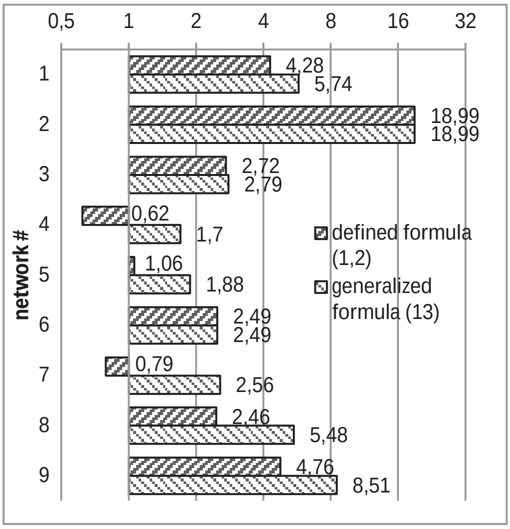
4 Characteristics in real world networks

We selected and studied several real world networks. Typical properties of small world networks – APL and CC, were observed in the studies. According to the measured values of the characteristics we have tried to associate chosen real world network from Table 1 with a proper network type.

Table 1. Basic characteristics of analysed real world networks. Datasets were obtained from online sources. Source 1 (`http://toreopsahl.com/datasets`), source 2 (`http://snap.stanford.edu/data`).

#	Network name	Source	Edge orientation	Graph connectivity	Number of vertices	Number of edges
1	*C. Elegans neural net.*	1	*Directed*	*Disconnected*	*306*	*2,345*
2	*US power grid*	1	*Undirected*	*Connected*	*4,941*	*6,594*
3	*FB-like forum net.*	1	*Directed*	*Disconnected*	*1,899*	*20,296*
4	*US airports net.*	1	*Directed*	*Disconnected*	*7,976*	*30,501*
5	*Scientific collaboration net.*	1	*Undirected*	*Disconnected*	*16,726*	*47,594*
6	*FB-like social net.*	1	*Undirected*	*Connected*	*899*	*71,380*
7	*Wiki vote net.*	2	*Directed*	*Disconnected*	*7,115*	*103,689*
8	*Physical citation net.*	2	*Directed*	*Disconnected*	*27,770*	*352,807*
9	*Physical theory citation net.*	2	*Directed*	*Disconnected*	*34,546*	*421,578*

[3] Ultra-small network is considered as a network with higher information throughput according to APL.



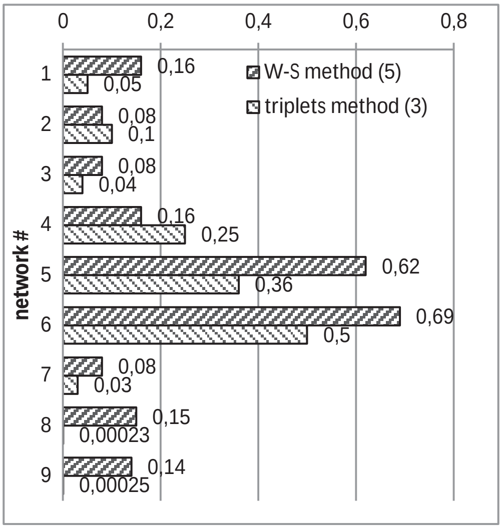


Figure 1. Analysed APL in selected networks.

Figure 2. Analysed CC in selected networks.

4.1 Average path length in disconnected networks

As is shown in Figure 1, defined formula of APL (1) along with modified formula of shortest path length (2) has produced incorrect output values in 2 out of 9 analysed networks, since the domain of an APL function l is $D(l) = \langle 1, \infty)$. Minimum of this function is set by the value of complete graph. Incorrect output values were measured for networks (#4, #7) from Table 1, which can be seen in Figure 1.

Cause of inaccuracies in the calculation can only be zero values obtained from modified shortest path length formula (2). Thus we generalized APL formula by combining the definition formula and the used formula of computation in disconnected networks described in section 2.2 based on computation only for largest component. In the proposed calculation is APL calculated in two steps. Firstly, an APL is calculated for each component of a graph separately. Secondly, a weighted average of all APLs is computed. A weight equals the number of component vertices and normalization constant equals the count of graph vertices.

$$l = \frac{\sum_i \left(l_{C_i} * n_{C_i} \right)}{n_G} \tag{13}$$

where $C_i \in G$ represents the component of graph G, l_{C_i} represents the APL of component C_i calculated in step 1, n_{C_i} represents the weight and its value equals or vertices count of component C_i, n_G represents the vertices count of graph G.

The generalised formula (13) is designed to generate output for connected networks equal to defined formula (1) with modification of shortest path length (2). For connected networks (#2, #6) from Table 1 is value of APL equal between defined and generalised formula, which demonstrate correctness of calculation.

The analysis shows that all networks except US power grid (#2) have sufficiently low value of APL. Accordingly to [8], C. Elegans neural network (#1) and both physics citation networks (#8, #9) best fit to six degrees of separation phenomenon which is characteristic for small world networks. Other networks show signs of ultra-small world property which is characteristic to scale-free networks. Values of measured CC can be seen in Figure 2. Most of networks show average value of CC, except the scientific collaboration network (#5) and Facebook-like social network (#6) which reached very high values. Whereas, high clustering property makes the greatest difference to small world networks and scale-free networks. Therefore we can most likely assign these two networks to small world networks.

5 Conclusions and related issues

Few computational methods of APL and CC as basic characteristics of small world networks have been studied and described. Also small world networks, random graphs and scale-free networks have been described along with their specific values of small world specific characteristics. Afterwards, we have analysed APL and CC in nine selected real world networks. We have generalized the formula of APL to disconnected networks since inappropriate values of this characteristic were observed.

When analysing CC, we have obtained some differences between values calculated by different methods. In related work we would like to study this inaccuracy. We believe that the inaccuracy is based on imperfect measurement of clustering property in large networks. Because clustering property can be seen at all levels of network clusters and not only at level of vertices. Therefor it is necessary to study this domain.

References

[1] Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Reviews of modern physics*, (2002), vol. 74, no. 1, pp. 47–97.

[2] Barabási, A.-L., Albert, R., Jeong, H.: Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications*, (1999), vol. 272, no. 1, pp. 173–187.

[3] Cohen, R., Havlin, S.: Scale-Free Networks Are Ultrasmall. *Physical Review Letters*, (2003), vol. 90, no. 5, 058701

[4] Erdős, P., Rényi, A.: On Random Graphs I. *Publicationes Mathematicae*, (1959), vol. 6, pp. 290–297.

[5] Erdős, P., Rényi, A.: On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, (1960), vol. 5, pp. 17–61.

[6] Gilbert, E.N.: Random Graphs. *The Annals of Mathematical Statistics*, (1959), vol. 30, no. 4, pp. 1141-1144.

[7] Luce, R.D., Perry, A.D.: A method of matrix analysis of group structure. *Psychometrika*, (1949), vol. 14, no. 1, pp. 95–116.

[8] Milgram, S.: The small world problem. *Psychology Today*, (1967), vol. 1, no. 1, pp. 61–67.

[9] Watts, D.J., Strogatz, S.H.: Collective dynamics of "small-world" networks. *Nature*, (1998), vol. 393, pp. 440–442.

[10] Watts, D.J.: Networks, dynamics, and the small-world phenomenon. *American Journal of Sociology*, (1999), vol. 105, no. 2, pp. 493-527.

Crowd Evacuation Simulation in Interior Areas

Michal KYŽŇANSKÝ *

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
michal.kyznansky@gmail.com

Abstract. The main aim of this paper is to analyse available solutions and models in the area of crowd simulation and choose proper model and software resources to create reliable simulation. Implementation of selected simulation should reach the level where it could fulfil advisory purposes such as getting answers for real life questions e.g. design of emergency exits in buildings. Created simulation is based on psychological model PECS and results of various test scenarios indicate that great level of resemblance with the real world was reached. Resulting models and visualizations can also serve to study human behaviour in general and improve its understanding.

1 Introduction

Human behaviour is being studied extensively by modern psychology in last decades but we are no closer to identify all variables that guide human behaviour. Despite the fact that it is determined by many unknown principles simulation of human behaviour has profound consequences for various areas such as design of emergency exists, police training (riots simulation [1]) or even military purposes.

In this paper we propose simulation of crowd based on PECS [2] psychological model with extensive coverage of many known behavioural patterns. Implementation contains collision avoidance system and agents interacting in environment were given additional attributes (e.g. fear, heart rate or stamina) to resemble reality as close as possible.

Proposed and implemented simulation is based on multi-agent toolkit MASON [3, 4] written in Java which offers new visual perspectives and provides complete control of simulation. Few scenarios are available and thanks to Drag & Drop system there are no limits for extending and customizing scenario to accommodate all sorts of requirements. We believe that such a tool can serve as advisory system for managing everyday life situations.

2 PECS psychological model

PECS [2] psychological model is acronym and stands for its four main components – *Physical conditions*, *Emotional state*, *Cognitive capabilities* and *Social status*. It was proposed as new referential model by Professor Schmidt, author of the book The Modelling of Human Behaviour. It

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 275–280.

is based on older BDI [5] (belief-desire-intention) model but it brings some crucial improvements. BDI model is still being improved and enriched with emotions [6].

Psychological researches showed that human behaviour follows (when calm or disturbed) one simple principle – resulting behaviour is not an activity which satisfies many motives at once but a result of competing motives which leads to execution of one or few actions to satisfy only one, the most prevailing motive.

PECS model has 3 layers: perception, body and actor. *Perception* gets information about outer environment, *body* consists of 4 modules that define the behaviour of agent and *actor* selects and executes activity based on winning motive generated by body.

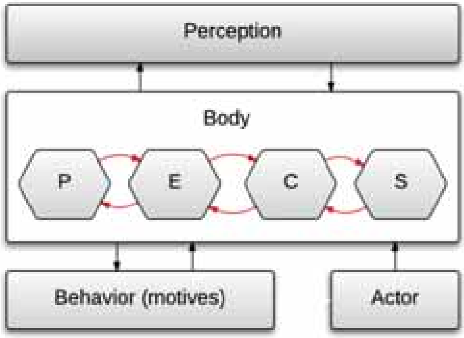


Figure 1. PECS psychological model - layers.

2.1 Collision avoidance

In proposed solution collision avoidance is implemented by two different methods for each type of collision (agent/walls). People in crowd try to avoid dense areas where they can get injured and avoid walls. *Negative gravitational pull* has proved to be a good approximation of avoiding the walls and *intimate zone factor* as a decent approximation of this motive. When multiple obstacles are close to the agent their repulsion forces fold into resulting vector.

$$\boldsymbol{F} = \mathcal{H} * \frac{m_{obstacle} * m_{agent}}{distance^2} \tag{1}$$

2.2 Emotions – Fear

Fear is a crucial human emotion which guides activities and amplifies reactions in tense situations. It is modelled by various techniques whose combination model fear transition for simulated agents (Figure 2). Fear is an emotion which is characterized by quick increase by discrete values when agent gets frightened and slow decreasing when calming down.

Calming down is described by equation 2 and is visible on Figure 2in time <0, 80>.

$$\boldsymbol{f_{fear}} = max_{fear} * e^{\left(-0.04*\ln\left(\frac{fear}{10}\right)*(-25)+\,0{,}5\right)} \tag{2}$$

In out model agent gets frightened when too many other agents are in his intimate zone (Figure 2 time <225, 235>). This behavior is essential for our simulation, because we are using it for emergence of panic condition when the crowd gets dense.

$$\boldsymbol{f_{fear}} = \boldsymbol{f_{fear}} + \ tooClose * 0.010 \tag{3}$$

Emotions, like fear, are spread between agents [7]. When there are agents with higher fear around, fear transition is in action and out agent gets frightened too. In this situation fear is modified by eq. 4.

$$\boldsymbol{f}_{\boldsymbol{fear}} = \boldsymbol{f}_{\boldsymbol{fear}} + e^{\left(\frac{agentsFear\,-\,fear}{3}\right)} - 1 \tag{4}$$

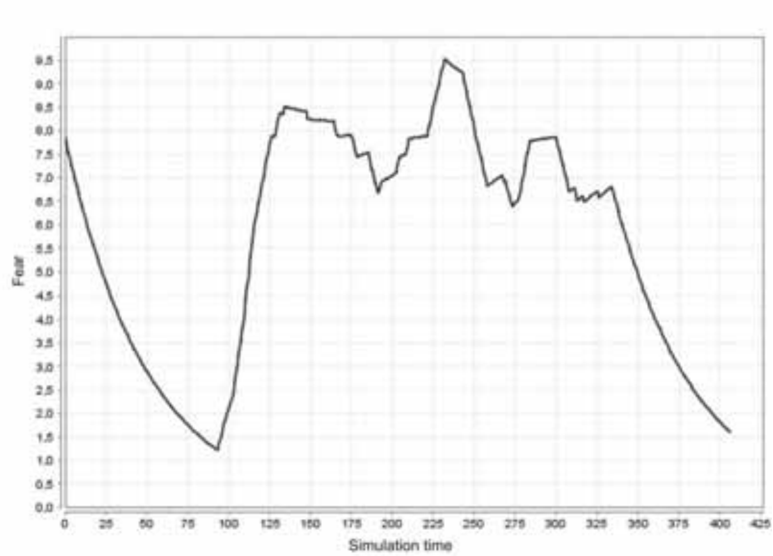


Figure 2. Example of fear transition for an agent during 400 simulation steps.

2.3 Physical conditions – Heart rate

Heart rate was chosen as a state variable of PECS physical module. Heart rate is in our simulation influenced by multiple factors such as age, speed of movement or fear. To simulate such uneasy variable we designed 3 functions that represent each factor. It has also upper limit which can be defined by generally accepted equation 220 – age.

Heart rate is slowly decreasing from higher values to value around 60 which is normal heart rate at calm state (eq. 5).

$$\boldsymbol{f}_{\boldsymbol{heart\_rate}} = (max_{heartRate} - 60) * e^{-0.004*\left(-250*\ln(heart\_rate - 60)*\frac{1}{max_{heart\_rate}-60}\right)} + 60 \tag{5}$$

Fear has a great impact on rising the heart rate and it usually causes quick discrete growth in heart rate (eq. 6).

$$\boldsymbol{f}_{\boldsymbol{heart\_rate}} = heart\_rate + e^{0.4*fear} \tag{6}$$

Speed of movement has significant impact on heart rate and in our simulation it was modelled by travelled distance per simulation step (eq. 7).

$$\boldsymbol{f}_{\boldsymbol{heart\_rate}} = hear\_rate + 5\ln(0.5 * travelled_{distance} + 1) \tag{7}$$

2.4 Physical conditions – Stamina/calories

Stamina as another state variable was introduced to model heterogeneous agent population in physical capabilities and predispositions. Agent’s stamina is decreasing while performing physical actions and is renewed while agent is resting. According to Journal of Sports Sciences [8] energy expenditure depends on gender, age, weight, and current heart rate. Eq. 8 describes energy expenditure for 80kg male agent.

$$\boldsymbol{f}_{\boldsymbol{calories/min}} = \frac{(-55{,}0969+0{,}6309*\,hear\_rate+0{,}1988 \text{ x } 80+0{,}2017*\,age)}{4{,}184} \tag{8}$$

2.5 Social satisfaction – Social status

We have integrated in our simulation social interaction in form of relationships between the agents. Every agent knows few other agents and when he/she gets one of those in sight he/she will try to get closer to them because there is a strong belief that when panic is around, familiar person will provide more secure environment than unknown crowd.

3 Motives and actions

PECS modes guides agents behaviour on multiple levels by state variables and transition between them that is occurring in each module. Agent's primary goal/motive is to get to his destination while it can be in certain stages of his journey supressed by other motives. Motives drive every action and create psychological justification for performing all actions. PECS model is flexible and does not require to fill all models. It generally consists of 3 layers (Figure 3).

Various state variables that describe specific details of human behaviour were implemented in this work. Motives are created from state variables (e.g. fear, heart rate, calories, age, weight, speed, etc.) and enter the competing process (with other motives) in each step of simulation.

Motives are generated inside the *Behaviour* component of PECS model. Information from *Perception* component is passed to body (modules) where state variables are transformed by complex functions into new state. This is a starting position for motive generating. After this phase motives are transformed to fixed scale (e.g. 0 to 10, 0 is not important, 10 is most serious) and can be numerically compared and winning motive for actual inner configuration of agent is picked.

The winning motive then enters the decision making process which is responsible for choosing action that will be performed in order to reduce actual weight of this motive (Figure 3). For several motives not only one actions can be chosen. An example is motive that results from state variable fear. When it gets sever there are several actions that can be executed depending on how serious this motive is. If it is less serious agent will accelerate which means he is getting nervous and starts to run. Otherwise it can end up in impaired judgment.

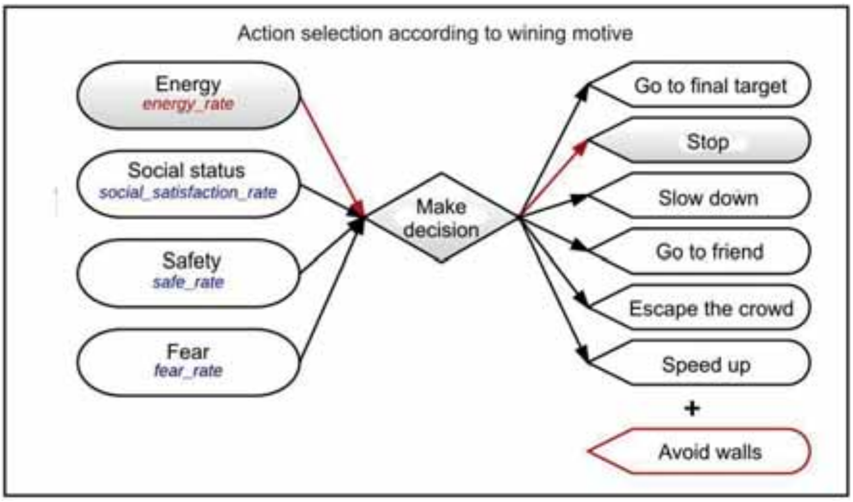


Figure 3. Wining motive is exhausted energy, so stopping and resting is winning action.

4 Example scenario

To test and verify our solution we constructed few test scenarios where agents tried to escape from building (see Figure 4) with different emergency exit locations. We used various number of agents and simulation parameters to get closer to reality and to find boundaries and limitations of our solution. Figure 4 shows visualisation of simulation in step 108, where 5000 agents were evacuated and 3 safe destinations (red spots on the right side) were provided. It is clear, that some agents were able to find alternative escape route, but the majority is pressed at the main exits, which very much resembles some critical crowd situations we are familiar with. Dense clustered patterns in front of emergency exists were formed and they are centres of panic and fear because these areas represent primary danger for agents (possibility of injury).

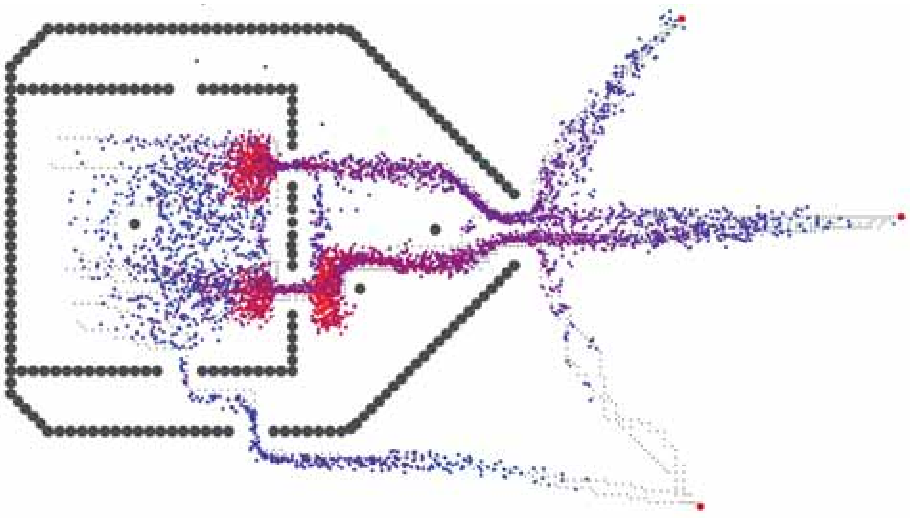


Figure 4. Simulation visualisation (step 108/1500).

5 Conclusions

Crowd simulation is still a big challenge for computer scientists as well as psychologists. Implemented simulation provided very real view of crowd evacuation in interior areas. There are still enhancements which will be added to our implementation e.g. pressure simulation for injury evaluation.

The most valuable contribution of this work is a powerful toolkit which can be further extended by adding more and more specific human behaviour patterns which can increase the level of resemblance with real word conditions. Our future work will include extension of social module for family simulation (agents will have very strong relations to 1-4 other agents). We are willing to simulate evacuation from big shopping centres with focus on parents with children.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0233-10.

References

[1] Lacko, P., et al.: Riot simulation in urban areas, *Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on*. (2014) pp.489-492

[2] Schmidt B.: Modelling of Human Behaviour The PECS Reference Model. *Proceedings 14th European Simulation Symposium*, 2002.

[3] Multiagent Simulation And the MASON Library, http://cs.gmu.edu/~eclab/projects/mason/manual.pdf. [2014-03-22]

[4] Paulovič, A Lacko, P., Návrat, P.: Agent-based Modelling and Simulation Tools. In: *WIKT 2011 Proceedings 6th Workshop on Intelligent and Knowledge oriented Technologies*, (2011) pp 97-102

[5] Schmidt, B.: Human Factors in Simulation Models, *19th European conference on modelling and simulation*, ECMS 2005, (2005)

[6] Korecko, S., Herich, T.:On some concepts of emotional engine for BDI agent system. *Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on*, (2013) pp 527-532

[7] Lacko, P: Modelovanie šírenia emócií v dave. In: *WIKT 2012 : 7th Workshop on Intelligent and Knowledge Oriented Technologies Proceedings*, STU Press, (2012) pp. 101-104

[8] Keytel, L. R., et al.: Prediction of energy expenditure from heart rate monitoring during submaximal exercise. *Journal of Sports Sciences* 23, no. 3 (2005), pp 289-297.

Biologically Inspired Approaches Used in Clustering

Michal KYŽŇANSKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`michal.kyznansky@gmail.com`

Abstract. In this article we propose an experiment performed on various data sets verifying the robustness and efficiency of bio-inspired algorithms, based on the so-called swarm intelligence (SI). We focused on specific behaviour of insect, particularly ants (ACA) and compared it to traditional clustering algorithms. During implementation ACA undergone a process of tuning and various enhancements were introduced to improve the results of clustering. The implementation proved that meta-heuristics based on the social behaviour of living organisms (insects) achieved excellent results in clustering. These results only confirm the latest approach in information technology, which is based on closer cooperation with other research fields, particularly biological ones.

1 Introduction

Human activities generate immense amounts of data which is getting increasingly difficult to interpret, classify and extract relevant information from it. Clustering as a subtask of data mining focuses on organizing objects into similar groups.

In this paper we propose experiment that examine new methods of clustering based on behaviour of insect (ants) and compare the results on various data sets with traditional algorithms such as K-Means and Hierarchical Agglomerative Clustering (HAC).

We also propose several modifications of Lumier-Faiera algorithm based on ants and present preliminary results. We propose new design of simulation application that is based on multi-agent toolkit MASON written in Java. We believe that complete control of simulation and visual observation will help to better understand the significance of individual features used in tested ant based algorithm.

2 Swarm intelligence

Term Swarm Intelligence (SI) was coined by Beny and Wang in 1989 who studied social behaviour of insect and its versatility while solving problems connected to their lives and

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Gabriela Kosková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

environment. They tried to imitate this behaviour in algorithmic ways by means of cellular automaton. The results of their studies were set of meta-heuristics (e.g. ants, bees [8], fish [10] etc.) that can be applied to vast variety of problems.

Typical system based on SI consists of individuals with low competence and only a limited set of activities. There is no central coordination entity or system and yet all individuals in colony work together to achieve one goal unaware of it. This is called emergent behaviour and it emerges from the activities of individuals in swarm.

3 Lumer-Faieta and Ant colony Algorithm

Meta-heuristic Ant colony optimization (ACO) was firstly described and published by Marco Dorigo in 1992 [3]. ACO is based on ants foraging behaviour. They explore space around the nest by random movement and they deposit pheromone on the ground which evaporates after certain time. Ants prefer to follow paths where pheromone is stronger to those without it. These two assumptions lead to conclusion that pheromone trail will inevitably last longer time period on shorter paths than on longer. When ant discovers food source more and more ants get to it by following evaporating pheromone trail and while they follow it they also deposit more and more pheromone which eventually leads to state where all ants follow the same shortest path to food source.

The fact that ants move randomly and pheromone evaporates gives them ability to explore vast space of solution for problem how to get to the food source by shortest path. Their movement is guided by positive feedback which is pheromone trail. This model was transformed into mathematical equations and successfully verified on famous NP-hard problems such as TSP (travelling salesman problem) and others [5].

The initial version of algorithm based on ACO meta- heuristic adapted for clustering problems was published in 1994 by Lumier, Faieta. It was also influenced by observing ants while cleaning the nest and clustering corpses of dead ants. The positive feedback in this algorithm does not depend on pheromone trail but was transformed info *neighbourhood function* $f(X_i)$. It has several parameters where *s* is the size of close neighbourhood that ant takes into consideration, *α* which scales the dissimilarity of documents [4].

$$f(X_i) = \max\left\{0, \frac{1}{s^2} \sum_{X_j \in N_{s \times s}(r)} \left[1 - \frac{d(x_i, x_j)}{\alpha}\right]\right\} \tag{1}$$

Ants move in 2D space where they pick and drop documents. Dropping is based on the neighbourhood function and the idea is to drop documents near similar ones and pick different ones and carry them elsewhere. Two functions control this process ($P_{drop}(X_i)$ and $P_{pick}(X_i)$) with pick and drop constants.

$$P_{drop}(X_i) = \begin{cases} 1.0 & if\ f(X_i) \geq k_d \\ 2 * f(X_i) & if\ f(X_i) < k_d \end{cases} \tag{2}$$

$$P_{pick}(X_i) = \left(\frac{k_p}{k_p + f(X_i)}\right)^2 \tag{3}$$

Standard ACA/Lumer-Faieta algorithm can be described in these steps [4].

```
Place each object $X_i$ of dataset on 2D grid on random cell
Place no_of_ants ants on 2D grid on random empty cells
while iteration_count < maximum_iteration do
    for i = 1 to no_of_ants do

        if ant doesn’t carry object and cell is occupied by $X_i$
```

```
                  calculate $f(X_i)$ and $P_{pick}(X_i)$
            else
                  if ant carries $X_i$ and cell not occupied
                        calculate $f(X_i)$ and $P_{drop}(X_i)$
                        drop object with probability $P_{drop}(X_i)$
            move to random unoccupied cell in neighbourhood
  iteration_count = iteration_count + 1
```

4 Proposed experiment

The main focus of the proposed experiment is to analyse and compare biologically inspired algorithms used for clustering (ants, fish, bees etc.) and as preliminary results are available, only ACA will be taken into consideration. In this paper we propose experiment where traditional algorithms will compete with slightly modified ACA on three datasets. Constitution of datasets have been chosen deliberately to prove versatility especially in textual documents (dataset R8). Experiment used both cosine and Euclidian distance to measure similarity/dissimilarity of document pairs. More algorithms will be added in the future to make the comparison thorough.

The experiment used two traditional algorithms K-means and Hierarchical Agglomerative Clustering (HAC). Empty clusters created while running K-means were reinitialized with outlier and in HAC the distance matrix was represented as heap.

4.1 Data sets

The first chosen dataset for the experiment is *IRIS* which can be considered as most popular and standard dataset in clustering and classification [11]. *IRIS* consists of 150 samples (each has 4 attributes/dimensions) divided into 3 classes evenly. Second one is *Breast Cancer Wisconsin (Diagnostic)* which consists of 569 samples (each has 30 attributes/dimensions) divided into 2 groups – malign (212) and benign (357). The last one is textual dataset R8 which consists of 7674 samples divided into 8 groups. In this paper R8 dataset was used in form of TF-IDF vectors.

4.2 Results verification

For experiment described in this paper two techniques were used to verify assumptions.
First one was using Davies-Bouldin and Silhouette indices to determine numeric quality of clustering results. Second method was visual observation of results.

5 Implementation

JAVA was chosen as implementation language thanks to its portability, maturity and mainly because we wanted to integrate robust simulation toolkit MASON.

5.1 MASON (Multi-Agent Simulator Of Neighborhoods)

MASON is robust, minimalistic toolkit for simulation of multi-agent systems. It is highly optimized for performance and it provides various useful data types (e.g. sparse matrix) and replaces some original JAVA implementations (e.g. random number generator *java.lang.Math.random()* or *java.util.ArrayList*) with optimized and tuned ones (e.g. *Mersenne twister* or *Bag*).

MASON is based on MVC architecture and enhances simulations with visual observation. Simulations can be run with or without GUI and simulations parameters can be altered while running. These particular features along with Drag & Drop options, start/stop were used to tune and debug individual features of ACA algorithm. MASON was chosen for this project because of extensive experience with simulating multi-agent systems. No special modifications to ACA algorithm were necessary to accommodate this toolkit. K-means and HAC were run in MASON as 1 step of pseudo-simulation process.

5.2 Algorithm modifications

Ant based algorithm implementation started in pure form which is Lumer-Faieta. Then various modifications described in many sources were added and parameters were tweaked to explore its robustness and verify assumptions [1][2][6] of other authors about ACA. In this paper we summarize parameter tweaking process and specify details about slightly modified way of adapting parameter α.

Grid – was used bounded although experiments with toroidal were performed.

Ant movement – was implemented by means of random walking on the grid. Ant chooses one column in its neighbourhood (defined by parameter s) where no ant is present and goes there. Alternative to this implementation is skipping completely this phase and assigning document to ant as soon as it is not holding one.

Adaptive changing of α – Parameter α scales dissimilarity and it heavily depends on the used dataset. This means that its value can't be easily guessed. More studies [4][12] propose adaptive changing of this crucial parameter and introducing heterogeneous population where each ant maintains its own value [12] while there are several techniques how to adapt α. In our experiment each ant will start with initialized value of parameter α (0.1) and after 250 steps it will increase by 0.01.

- Parameters [4]
  - Ants count $\frac{1}{3}N$
  - Grid size $\sqrt{10N} * \sqrt{10N}$
  - $\alpha = 0.35$
  - $k_p = 0.1, k_d = 0.1, s = 2$

6 Results

Preliminary results of proposed experiment are based on visualisation of clustering space after chosen number of steps and marking documents by different colours which correspond to different classes according the gold standard (Figure 1.). This can be used to see convergence state of clustering solution. Quality indices were the second method used to compare results of clustering and validate high expectations from biologically inspired algorithm – ACA.

Table 1. Results of quality indices on various datasets.

	K-means			**HAC**			**ACA**		
	IRIS	*Breast*	*R8*	*IRIS*	*Breast*	*R8*	*IRIS*	*Breast*	*R8*
Silhouette index	0.552	0.697	0.079	0.551	0.690	0.056	0.522	0.642	0.105
DB index	0.662	0.504	3.806	0.658	0.429	3.677	0.595	0.621	3.182

Table 1. shows clustering configurations for three datasets (*IRIS*, *Breast* and *R8*) created by three algorithms (K-means, HAC and ACA) expressed by values of Silhouette and Davies-Bouldin index. Best result for combination quality index and dataset is underlined. These results show that

ACA can be considered as more robust solution and performs better on multidimensional datasets. Table 1. also shows good performance of ACA in comparison with traditional algorithms such as K-means and HAC.

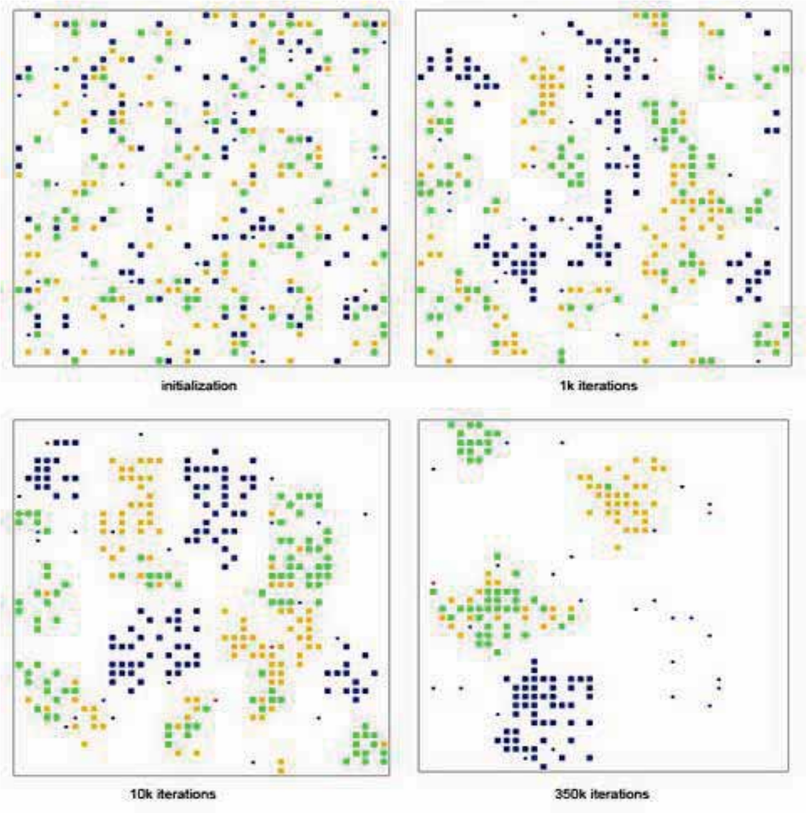


Figure 1. ACA applied on IRIS dataset, stages.

7 Conclusions

Preliminary experiment performed in this paper proved that biologically inspired algorithms have a great potential in solving hard clustering problems. Tweaking process of these algorithms represents true challenge for obtaining good results and was examined for ACA algorithm by tweaking and modifications.

Another conclusion that this paper supports is that more and more technologies and methods are being based on solutions that exist in nature for millions of years and we are increasingly unlocking these techniques observed in various living organisms by mimicking them by means of algorithms or prototypes.

Implementation proved that traditional algorithms such as K-Means and Hierarchical Agglomerative Clustering (HAC) still provide decent results in respect to ease of implementation and their simplicity. They can be used as primordial tool for preparing non-textual datasets because they perform sufficiently for datasets with low dimensions.

Another contribution is solving K-Means empty cluster anomaly by reinitialising it by furthest outlier.

The most valuable achievement of the experiment was tool based on popular multi-agent simulation environment MASON that can run ACA algorithm in new fashion, able to change even minute details of simulation in-runtime and obtain large quantities of visual data.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0752/14.

References

[1] Ajith A., Swagatam D., Sandip R.: Swarm Intelligence Algorithms for Data Clustering. In: *Soft Computing for Knowledge Discovery and Data Mining*, Springer, Berlin, (2007), pp. 279-313.

[2] Bharne, P.K. et. al.: Data clustering algorithms based on Swarm Intelligence. In: *Electronics Computer Technology (ICECT)*, 2011 3rd International Conference on, vol. 4, (2011), pp. 407-411.

[3] Dorigo M., Stützle T.: *Ant Colony Optimization*. 1. edition. London: MIT Press, (2004). 319 pp. ISBN 978-0-262-04219-2 .

[4] Handl J., Knowles J., Dorigo M.: Ant-Based Clustering: A Comparative Study of its relative performance with respect to k-means, average link and 1D-SOM (2003). In: *Design and Application of Hybrid Intelligent Systems*.

[5] Bharne, P.K.: Data clustering algorithms based on Swarm Intelligence, Electronics Computer Technology (ICECT), 3rd International Conference, (2011), vol.4, pp. 407-411.

[6] Santos, D.S.: A biologically-inspired distributed clustering algorithm, Swarm Intelligence Symposium, (2009). SIS '09. IEEE, pp. 160-167.

[7] Krishnamoorthi, M., Natarajan, A.M.: A comparative analysis of enhanced Artificial Bee Colony algorithms for data clustering, Computer Communication and Informatics (ICCCI), (2013), International Conference on, pp. 1-6.

[8] Zhang C., Ouyang D., Ning J.: An artificial bee colony approach for clustering, Expert Systems with Applications, Volume 37, vol. 7, (2010), pp. 4761-4767, ISSN 0957-4174.

[9] Zhou Y., Liu B.: Two Novel Swarm Intelligence Clustering Analysis Methods, Natural Computation, (2009). ICNC '09. Fifth International Conference on, vol.4, pp.497-501.

[10] Xiao L.: A clustering algorithm based on artificial fish school, Computer Engineering and Technology (ICCET), (2010), 2nd International Conference on, vol.7, pp. V7-766, V7-769.

[11] *UCI Machine Learning Repository*. [Online]. Available at: http://archive.ics.uci.edu/ml/index.html.

[12] Handl J., Meyer B.: Improved Ant-Based Clustering and Sorting in a Document Retrieval Interface. In: *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)*, volume 2439 of LNCS, (2002).

[13] Handl Pedersen M. E. H.: Ant Colony Clustering & Sorting. Aarhus: Daimi, University of Aarhus, (2003), 9 pages.

DNA Fragment Assembly by Using of Mosquito Host Seeking Algorithm

Milan MARTINKOVIČ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
martinkovic@fiit.stuba.sk

Abstract. DNA fragment assembly problem is one of the most complex problems faced by computational biologists. In this problem we have to build up a complete DNA sequence from a set of small DNA fragments. Whole process of determining of complete DNA sequence is a NP-hard problem and what is more this process can be complicated by issues such as reading errors in fragments or relative fragment orientation and so on. Therefore bio-inspired algorithms, which are based on meta-heuristic principle, are suitable candidates for solving of such a problem as DNA fragment assembly. One of these algorithms is mosquito host seeking algorithm, which application on fragment assembly problem is described in this paper.

1 Introduction

Deoxyribonucleic Acid (DNA), consists of four kinds of nucleobases (guanine, adenine, thymine and cytosine). In DNA fragment assembly problem, we have to determine structure of original DNA, which is represented by combination of these four nucleobases, where length of this combination can vary from ten thousand to hundreds of millions of bases depending on studied organism. Complexity of DNA fragment assembly problem arises from the fact that nowadays technologies are not able to read whole genome at once but only fragments of a short length.

Reading of structure of DNA is called sequencing. This process produces huge amount of short fragments with information about their structure but without knowing of exact location and ordering of these fragments. In order to be able to determine ordering of fragments, original DNA is cloned and process of sequencing is repeated on these copies as well. In every iteration of this process, genome is split into short fragments in different parts of genome. Because of this fact we can use overlapping of fragments in order to retrieve exact ordering of fragments (see Figure 1).

There are many known solutions for DNA fragment assembly problem. The most popular approach comes from using of benefits of graph theory. Problem is represented as some kind of graph and finding of path through this graph, which satisfies a set of rules that are specific for a particular approach, will produce solution for the problem. Classic solutions using this kind of approach are for example: YAGA [1], that uses bidirected string graphs or PASHA [3], which uses

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Anna Bou Ezzeddine, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

another kind of graph, called de Bruijn graph. Another solutions that can be mentioned are Velvet [9] or Abbys [5].

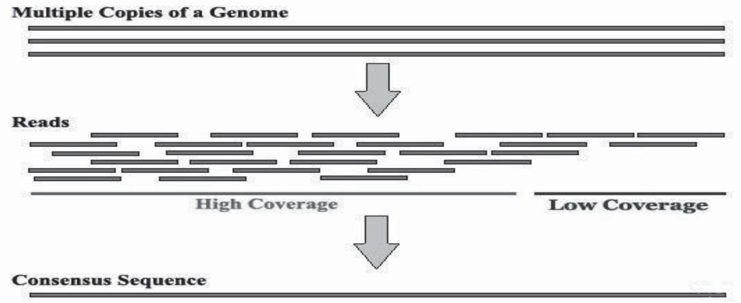


Figure 1. Assembling based on overlapping of fragments.

2 Biologically inspired algorithms

Nature is unfailing source of inspiration. Majority of organisms are able to cope with difficult problems without presence of some complex directional logic and therefore can algorithms, which are inspired by these organisms, achieve very good results when applied to complex optimization problems such as DNA fragment assembly problem.

Biologically inspired algorithms have become very popular in last few years. They were used in a lot of scientific fields and were proven to be very effective in many difficult tasks. One of these tasks is also solving of DNA fragment assembly problem. Some of the bio-inspired algorithms were successfully applied on this problem.

For example in work by Verma, Singh and Kumar [7], authors are using algorithm, which is called particle swarm optimization to solve the fragment assembly problem. This algorithm is inspired by movement of bird flocks and fish schools. It is agent based algorithm, where agents are represented as particles. Every particle contains possible solution of the problem and by specific moving of particles it is enabled to perform communication between them. Algorithm runs through iterations and because of communication between particles it is enabled to find best solutions in each iteration.

Another source of inspiration from nature that was successfully applied on fragment assembly problem is behavior of honey bees, specifically inspiration by process of finding of pollen [2] and inspiration based on process of reproduction of bees [6]. First of the mentioned works describes also agent based iterative algorithm and the second one represents genetic algorithm.

On the basis of mentioned works, it is clear that bio-inspired algorithms are suitable candidates for solving of fragment assembly problem. In the next chapter we describe usage of bio-inspired algorithm that has not been used on problematic of fragment assembly problem so far. Algorithm is inspired by behavior of mosquitoes, specifically by seeking of their hosts.

3 Mosquito host seeking algorithm

Mosquito host seeking algorithm [8] was developed by Xiang Feng, Francis Lau and Huiqun Yu. Biological model of the algorithm describes behavior of mosquitoes in the process of searching for host that is source of their food, blood. Mosquito is attracted towards its victim by carbon dioxide, which is being exhaled by most of the living organisms. Following this track of carbon dioxide, mosquito comes to its victim so close that its thermo receptors can capture radiated heat from this possible host. When mosquito recognizes this radiated heat, it starts to directly follow this trail, until it finally reaches the host (see Figure 2).

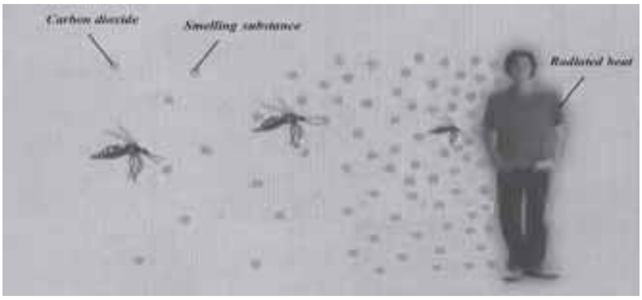


Figure 2. Biological model of algorithm [8].

Mathematical model is derived from the biological one and authors of the work successfully applied this algorithm on known optimization problem, called traveling salesman problem. All of the formulas are stated in the original work [8], however we mention the most important ones in equations 1, 2 and 3. Let us recall that in the traveling salesman problem, we are given a set of cities and routes with specific lengths between them and aim of a solution to this problem consists of finding of route, which visits every city exactly once but with fulfillment of condition that this route have to be the shortest one. This problem is in graph theory also called as finding of Hamiltonian path.

In mathematical model of the algorithm, there is exactly one mosquito assigned for every edge between the cities. When mosquito "attacks" its edge, it means that the shortest path will lead through this edge. Another important attribute of the algorithm is sex of mosquitoes. In real life only female mosquitoes suck blood. By applying this property into algorithm we are able to represent existent and nonexistent edges between cities. When the edge exists, female mosquito will be assigned to this edge and in opposite situation, where the edge does not exist, it will be male mosquito. This fact is represented in formulas of the algorithm and should ensure separation of nonexistent edges from the final solution. Let $u_{ij}(t)$ be the distance between mosquito m_{ij} and the edge at the time t, where i and j are indexes of neighboring cities. We define $u_{ij}(t)$ in equation 1.

$$u_{ij}(t) = \exp(-c_{ij}(t) r_{ij}(t) x_{ij}(t)) \tag{1}$$

The smaller $u_{ij}(t)$ is, the closer the mosquito is to the host. Variable c_{ij} represents weight of edge, r_{ij} stands for solution variable which represents whether mosquito attacks its edge or not and variable x_{ij} represents sex of mosquito. In equation 2 and 3 we define computation of change of variables r and c in time t respectively. J represents utility sum, P stands for attraction function and Q stands for interaction behavior function.

$$dr_{ij}(t)/dt = -\lambda_1 \frac{\partial u_{ij}(t)}{\partial r_{ij}(t)} - \lambda_2 \frac{\partial J(t)}{\partial r_{ij}(t)} - \lambda_3 \frac{\partial P(t)}{\partial r_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial r_{ij}(t)} \tag{2}$$

$$dc_{ij}(t)/dt = -\lambda_1 \frac{\partial u_{ij}(t)}{\partial c_{ij}(t)} - \lambda_2 \frac{\partial J(t)}{\partial c_{ij}(t)} - \lambda_3 \frac{\partial P(t)}{\partial c_{ij}(t)} - \lambda_4 \frac{\partial Q(t)}{\partial c_{ij}(t)} \tag{3}$$

3.1 Application on fragment assembly problem

As mentioned above, mosquito host seeking algorithm was developed in order to solve traveling salesman problem, however fragment assembly problem can be also converted into traveling salesman problem. Conversion consists of the fact that fragments can be considered as cities and length of an overlap between fragments can be taken as distance between cities. Finally by passing through this converted graph, with maximization of traveled distance (difference against original TSP), we can acquire final string that will be consisting of joined fragments, based on quality of their overlapping, what is in fact aim of solving the fragment assembly problem. It is important to mention that in DNA fragment assembly problem, version of the TSP is asymmetric one. It means that for example distance from city (fragment) A to B can be 5 but in reverse direction from B to A it can be 8. It is caused by the fact that we need to represent direction of reading of fragments. Basic principle of mentioned procedure is described in Figure 3. For simplification we show symmetric version of the graph.

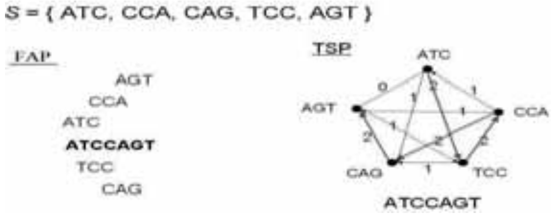


Figure 3. Principle of conversion to TSP.

3.2 Implementation

We have successfully implemented mosquito host seeking algorithm. Algorithm is written in C language with combination of popular technology for parallel processing, called OpenMP[1]. Mosquito algorithm has great potential in parallelism because of its core logic, which supports inherent parallelism and has all of the prerequisites to reduce total running time of the algorithm.

During implementation of the algorithm, we had to overcome few problems. As we have already mentioned, version of the TSP that we need to use for DNA fragment assembly problem is the asymmetric one. However mosquito host seeking algorithm is able to cope with symmetric version only. Because of this fact we decided to convert asymmetric distance matrix of cities, which is main source of information for TSP, into its symmetric version. This conversion is done by opposite placing of transposed matrix to the original matrix but this operation leaves two other corners of newly created matrix empty so the next step is filling of these corners by values which will not affect the algorithm and will be ignored. In paper [4], it is mentioned that suitable value for these corners is constant 2n*d_{max}, where n is number of cities and d_{max} stands for maximal value of original matrix. Described conversion can be seen in equation 4, where M is newly created symmetric matrix, O is original asymmetric matrix, O^T stands for transposed original matrix and D represents corners filled with dummy values.

$$M' = \begin{matrix} D & O^T \\ O & D \end{matrix} \tag{4}$$

Another problem that we had to solve was with assigning of sex to the mosquitoes. Formulas described in [8] involve cooperating with values that describe sex of the mosquito (0 for male and 1 for female) but when we were using male mosquitoes for problems that contained nonexistent edges between cities, the algorithm was not able to solve this problems. However in the case of problems, where all of the edges were available, algorithm worked without problems. We solved this problem by disadvantaging of edges which were nonexistent by assigning of values that will cause the algorithm to ignore them. Because of the characteristics of the algorithm, assigning of extreme values caused it to stuck so we had to develop a little bit more sophisticated method to find suitable values. Finally we created the method which searches nearest edges to the one which

[1] http://openmp.org/wp/

we want to declare as nonexistent and on the basis of these values we compute new value that is the most undesirable of all of the neighboring edges but not too extreme so the algorithm can be fully functional. This value is computed as rand(max) + max + max/10. Where rand is function for retrieval of random number which is not higher than max and max represents the highest value among neighboring edges.

4 Results

We have successfully applied mosquito host seeking algorithm on fragment assembly problem. However due to need of further scaling of algorithm we are currently able to present results for smaller problems only.

Firstly, in the table 1, we compare optimality of solution received by mosquito host seeking algorithm and known optimal results for TSP benchmark problems acquired from project called TSPLIB[2] to prove optimality of algorithm. Values in the table stand for length of the shortest route found by mosquito algorithm (MOSQ) and known optimal value (OPT).

Table 1. Optimality of solution.

Dataset(number of cities)	OPT	MOSQ
ftv44(45)	1613	~~1604~~
ft53(53)	6905	6964
ftv70(71)	1950	1965
kro124p(100)	36230	39255

As seen in table 1, results of mosquito algorithm are comparable with best known solutions. In the first case we can see that path received by mosquito algorithm is shorter than the shortest one. This is caused by situation where algorithm gets stuck on local optimum and the route is not formed of one circle but with two or more. However result will be still very close to optimal solution so it is still interesting result for solving of DNA fragment assembly problem.

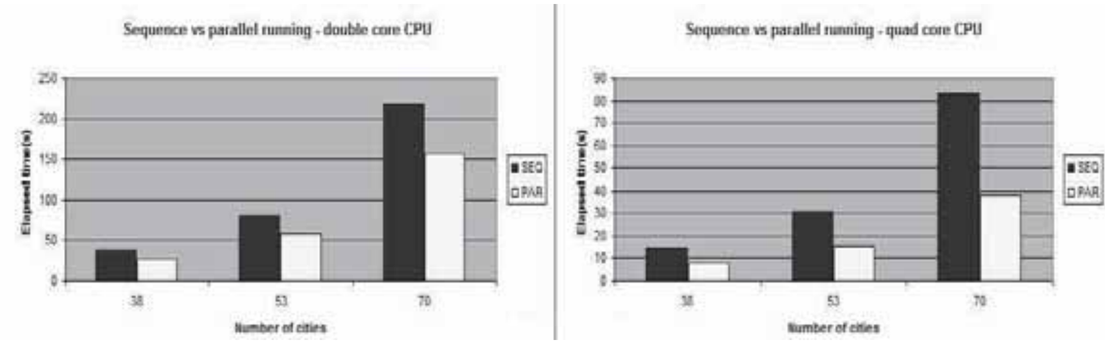


Figure 4. Reduction of time needed for execution.

Secondly, we applied mosquito algorithm on DNA fragment assembly problem. Data were received from DNA FAP repository[3]. We have used all of the available instances of benchmark x60189 consisting of 39, 48, 66 and 68 fragments and by using of our proposed algorithm we have successfully assembled original sequence of total length of 3835 bases for each instance of this benchmark. Result of algorithm is set of longer sequences called contigs. Contig consists of

[2] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[3] http://www.mallen.mx/fragbench/genfrag.php

fragments of DNA that have high overlap score and its structure can be declared to represent real piece of DNA. After retrieval of these contigs we were able to reconstruct original sequence from which the fragments were originated by repeated exploitation of overlapping.

As we have already mentioned, mosquito host seeking algorithm has great potential in parallelism. We have successfully implemented parallelism based on shared memory. Improvements of total running time are shown in figure 4. We achieve approximately 30% reduction of total time needed for execution of algorithm when used on CPU with two cores. However when executed on four cores, improvement exceeds 50%. These numbers are very promising since fragment assembly problem is very time consuming procedure.

5 Conclusions and future work

In this paper we have proposed new solution for DNA fragment assembly problem, based on biologically inspired algorithm, specifically mosquito host seeking algorithm. We have proven that this algorithm is capable of solving of fragment assembly problem and we have also shown great potential of the algorithm in the way of parallel processing.

In the future work we want to apply our algorithm on more complex dna fragment assembly problems since results from smaller problems are promising. In order to fulfill that we also want to implement parallelism based on distributed memory, what can be very effective when combined with parallelism based on shared memory that has been already implemented.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0752/14.

References

[1] Jackson, B. G.; Regennitter, M.; Yang, X.; Schnable, P.S.; Aluru, S., "Parallel de novo assembly of large genomes from high-throughput short reads," *Parallel & Distributed Processing (IPDPS), IEEE International Symposium*, (2010), pp.1,10.

[2] Jesun Sahariar Firoz; M. Sohel Rahman; and Tanay Kumar Saha., "Bee algorithms for solving DNA fragment assembly problem with noisy and noiseless data." *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference* (GECCO '12), Terence Soule (Ed.). ACM, New York, NY, USA, pp. 201-208.

[3] Liu Y.; Schmidt B.; Maskell D., "Parallelized short read assembly of large genomes using de Bruijn graphs" *BMC Bioinformatics*, (2011).

[4] Ratnesh K.; Haomin L., "On Asymmetric TSP: Transformation from Symmetric TSP and Performance Bound" *Journal of Operations Research*, (1994).

[5] Simpson T.; Wong K.; Jackman S.D.; Schein J.E.; Jones S.J.; and Birol I., "ABySS: a parallel assembler for short read sequence data.“ *Genome Research*, Preprint, (2009).

[6] Sung Hoon Jung, "Queen-bee evolution for genetic algorithms," *Electronics Letters* , vol.39, no.6, (2003), pp.575,576.

[7] Verma, R. S.; Singh, V.; Kumar, S., "DNA Sequence Assembly using Particle Swarm Optimization“ *International Journal of Computer Applications*; Vol. 28, (2011), p 33.

[8] Xiang Feng; Francis C. M. Lau; and Huiqun Yu., "A novel bio-inspired approach based on the behavior of mosquitoes." *Inf. Sci.* 233, 87-108, (2013).

[9] Zerbino D. and Birney. E., "Velvet: Algorithms for de novo short read assembly using de Bruijn graphs.“ *Genome Re-search*, 18:821-829, (2008).

Evolutionary Algorithm to Solve Rubik's Cube

Miroslav Ort[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xort@is.stuba.sk`

Abstract. Since 1974, the year of its invention by Ernö Rubik, Rubik's Cube has become very popular. Thus for computer scientists there was a challenge to ascertain if computers could help to solve the puzzle. In the last 40 years software engineers have discovered the ways how to solve the Rubik's Cube using computers. Besides using complete search algorithms with heuristics, evolutionary algorithms are used. In this paper we first briefly describe complete search algorithms to solve the Rubik's Cube. Our research is focused on evolutionary algorithms. In the paper we define and describe the design of the proposed evolutionary algorithm in detail. We conduct some experiments with our proposed prototype and compare it with a chosen already existing evolutionary algorithm to solve the Rubik's Cube. In conclusion we discuss future work.

1 Introduction

In 1974, the year of the invention of the Rubik's Cube, a new puzzle was invented by Hungarian architect Ernö Rubik. Rubik's Cube became popular quickly. There are many reasons why this puzzle is still so modern and many people are interested in solving it, but the most important one is that puzzle is 3-dimensional with diameters n x n x n [7]. The standard version of the Rubik's Cube consists of a 3 x 3 x 3 cube, with different colored stickers on each of the exposed squares of the subcubes called also *cubies*.

The standard version of the Rubik's Cube, which is the subject of this paper, thus consists of 26 visible cubies:

- 8 corner cubies,
- 12 edge cubies,
- 6 center cubies.

Each side of the Cube is called *face* and each visible colored side of a cubie is *facelet*. Each face of the Cube rotates separately in 90, 180, or 270 degrees clockwise or counterclockwise.

[*] Master study programme in field: Software Engineering
Supervisor: Professor Jiří Pospíchal, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

There are several known notations for applying single moves on the Rubik's Cube. We use *de-facto* standard notations *F*, *R*, *U*, *B*, *L*, *D* to denote a clockwise 90 degrees turn of the front, right, up, back, left, down face and F_i, R_i, U_i, B_i, L_i, D_i for a corresponding counterclockwise quarter-turn. We do not take into consideration other possible notations, such as notations for double moves.

The puzzle is scrambled by making a number of random rotations of its faces. The goal is to reach the solved configuration, in which all the squares on each side of the Cube are the same color. Notice that finding the solution of a given Rubik's Cube is to determine the sequence of moves whose execution lead to the solved configuration. This task is not trivial because the number of all reachable configurations of the Cube is $8!x3^8x12!x2^{12} = 43\ 252\ 003\ 274\ 489\ 856\ 000$ [5].

2 Related work

Two primary approaches how to solve this puzzle exists. There are non-evolutionary and evolutionary methods.

2.1 Non-evolutionary algorithms

The first published method of solving the Rubik's Cube was discovered by Thistlethwaite [6]. This algorithm was devised in 1981 [2]. It is based on solving the Cube by breaking the whole complex optimization problem into smaller parts which are being solved separately. The solution of each subproblem is combined with other solutions to gain a solution of the original complex problem. Finding the solution for each subproblem is based on finding a solution in precalculated lookup tables.

Thistlethwaite's method was improved by Kociemba in 1992 [6]. Kociemba showed the Cube could be solved by using only two-phased algorithm. In each phase only a few possible rotations are applied on the Cube. To find a solution of each subproblem IDA* algorithm is used with a heuristic function which is a memory based lookup table and allows pruning up to 12 moves in advance [4]. Korf devised in 1997 the algorithm which could find an optimal solution. His method uses IDA* algorithm and lookup tables used as pattern databases [5].

2.2 Evolutionary algorithms

In 1994 Herdy devised a method [2] which solves the Cube. This algorithm uses predefined sequences as mutation operators that only alter a few cubies. Another approach is Thistlethwaite's evolutionary strategy [3] which improves its non-evolutionary sibling by using fitness evaluation to solve each subproblem instead of precalculated lookup tables. In 2009 Borschbach and Grelle presented their evolutionary approach which incorporated human strategies of solving the puzzle [1] called *HuGO!*. This algorithm is divided into three phases that are similar to a human strategy where firstly subcube 2x2x3 is solved and finally the rest of the Cube.

3 Design Of our solution

The aim of the solution is to solve the Rubik's Cube from any scrambled configuration. We do not want to set any restrictions or constraints to any inceptive state of the Cube. However, due to the number of all reachable configurations we assume the most prospective evolutionary algorithm is a genetic algorithm. This evolutionary algorithm is able to evaluate many individuals at once more independently within a population. Thus we can simulate biological evolution more precisely and gain better solutions.

The improvement of the genetic algorithm we see in defining sequences of rotations that could twist or move a few cubies at once. The idea is derived from tutorials and guides for solving Rubik's Cube which help humans. We believe this proposal will lead to better solutions.

The enormous number of the Cube's states may lead to a lot of local optima. We need to prevent being trapped in a local optima and therefore we suggest pausing the currently unpromising run of evolution and starting it from the very beginning. The newly created evolution process we will call *helping* evolution and the paused evolution *main* evolution. After the helping evolution is done, the whole population is joined with the population of the main evolution. After the merge the main evolution continues as usually. This leads to double number the individuals in population for one generation; after the generation the population size is back at the value before the merge. We assume helping population may bring new rotations and cube's moves and prevent the main evolution being stuck in a local optima.

3.1 Representation

The representation of the Rubik's Cube may not be so important to find a solution, but it could improve the calculation of the whole algorithm. We demand the representation to contain all necessary information, but, on the other hand, it must be simple and memory efficient. Therefore, we represent the Cube as a 3-dimensional array of strings. This depiction illustrates the original puzzle.

Each string presents one cubie. The string contains of cubie's color and its rotation in a virtual 3D grid. The example of such a representation is shown in the figure (Figure 1).

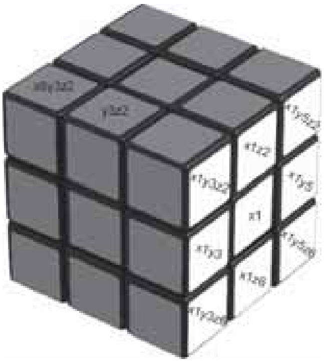


Figure 1. Representation of the Cube.

As it may be seen in the figure color of each facelet is represented by a number. Numbers, which represent colors, are chosen from 1 to 6 and related colors in ascending order are: white, orange, green, red, blue, yellow.

3.2 Fitness function

Every good fitness function should return the quantified distance to the solution, in this case the number of moves needed to reach the goal state. When solving the complex puzzle like Rubik's Cube, it is not possible to have such a function. Thus we simulate the imaginary best fitness function by the function which returns the various differences between the current state of the Cube and the goal state - the solved Cube.

To evaluate the fitness of an individual we define four quality parameters:

- colorDiffersFromCenter
  - ○ a facelet color differs from the center facelet on the same face
  - ○ parameter is increased by $+1$ for each wrong facelet

- wronglyPositionedEdge
  - ○ edge cubie is wrongly positioned
  - ○ parameter is increased by $+2$ for each wrongly positioned edge cubie, orientation is not considered
- wronglyPositionedCorner
  - ○ corner cubie is wrongly positioned
  - ○ parameter is increased by $+3$ for each wrongly positioned corner cubie, orientation is not considered
- wronglyOrientedCubies
  - ○ edge or corner cubie is wrongly oriented
  - ○ parameter is increased by $+4$ for each wrongly oriented corner or edge cubie

The first three parameters we borrowed from Herdy's approach [2] and adapted their weights. The maximum value of the objective function calculated via previously defined parameters is 180 (See Equation 1). The Cube is solved when objective function is 0.

$$objectivefunction_{max} = 48 * 1 + 12 * 2 + 8 * 3 + 21 * 4 = 180 \tag{1}$$

3.3 Mutation

Mutation is performed for the most individuals in population. We propose to mutate individuals using mutation operators or using a single random rotation. Mutation operators are defined as sequences of rotations. The sequences are adopted from tutorials and guides for solving Rubik's Cube[1][2][3]. An individual is mutated by a random mutation operator with higher probability then by a single random rotation. However, we believe a single rotation may bring the required variation to the current Cube's state.

Mutation operators are not defined for every corner or edge. This may cause that one or a few faces are never rotated by a mutation operator. Thus we propose random rotations of the whole Cube before applying any mutation operators.

3.4 Selection and crossover

As a selection method we have chosen tournament. According to [2] simple selection method is effective. After the selection, two parents are randomly chosen and two offspring are created by a crossover, which then replace the parents.

The crossover operator is Cartesian product which is realized over parents' moves and whose results are evaluated by fitness function. Each move is considered to be atomic. In terms of mutation operators the whole mutation operator is a part of the Cartesian product without splitting it into single quarter moves. The offspring with the highest fitness value is selected to be in the next generation with high a probability.

[1] `http://cubemania.cz/rubikova-kostka/rubikova-kostka-3x3x3/rubikova-kostka-navod`; last accessed January 15th, 2014

[2] `http://www.hlavolam.maweb.eu/rubikova-kostka-3x3x3`; last accessed December 06th, 2013

[3] `http://www.hlavolam.maweb.eu/rubikova-kostka-poslepu`; last accessed December 06th, 2013

Table 1. Three turns scrambles results.

	HuGO!		Our Proposed Solution	
Test No.	**Min generations**	**Total turns**	**Avg generations**	**Total turns**
1	50	3	16	3
2	150	3	29	4
3	50	3	14	3

Table 2. Four turns scrambles results.

	HuGO!		Our Proposed Solution	
Test No.	**Min generations**	**Avg turns**	**Avg Generations**	**Avg turns**
1	50	5	38	4

4 Experiments

With the proposed evolutionary algorithm we conducted some experiments. We are going to compare our approach to the reference algorithm HuGO!. HuGO! algorithm also includes human strategies in solving the Cube. The population size is set to 50. To scramble the Cube we use the online generator[4] which is used for generating scrambled Rubik's Cubes for WCA[5].

The first experiment involves three-turns scrambles. In the Table 1 there is a comparison between our proposed algorithm and algorithm HuGO!. The second experiment is conducted with four-turns scrambles. The results are shown in the Table 2. All results are rounded and results of algorithm HuGO! are adopted from [1]. Due to the lack of information we are not able to determine which moves were used in experiments with HuGO! to scramble the Cube. To scramble the Cube we used moves:

- 3-turns - R' F' R'
- 4-turns - U' F R U

In the third experiment we also used four-turns scrambles. In this performance test 15 different scrambles were used. For each scramble our proposed algorithm generated solution 10 times. The resulting average solution length is 8.45 turns. 74% of the scrambles were solved in maximum of 4 turns. The rest 26% had an average solution length 25.10 turns. From these results we gained less precision in comparison to HuGO! algorithm with its 88% of the scrambles solved in a maximum of four turns. And also our average solution length is longer in comparison with 5 turns. However, average HuGO!'s minimal number of generations to find solution, 50 generations, which is more our average number of generation that equals 30.

[4] `http://ruwix.com/puzzle-scramble-generator/?type=rubiks-cube`; last accessed February 10th, 2014

[5] `https://www.worldcubeassociation.org/about`; last accessed February 10th, 2014

5 Discussion

In previous sections we introduced our evolutionary algorithm that is able to solve a given Rubik's Cube. According to the experiments we consider our approach as a promising solution. Our approach gives better results in number of generations needed to solve a scrambled Cube. On the other hand we can clearly see that sometimes the length of the sequence of moves exceeds the optimal or reference solution. The cause of this weakness we see in the relatively huge number of moves required to solve the Cube from nearly solved state (one or two cubies are wrongly positioned or oriented).

We would like to eliminate the weakness in future work. We believe that using pattern databases with nearly solved configurations and moves to the goal state will improve the algorithm in a desired manner.

References

[1] Borschbach, M., Grelle, C.: Empirical Benchmarks of a Genetic Algorithm Incorporating Human Strategies. Technical report, Technical Report, University of Applied Sciences, Bergisch Gladbach, 2009.

[2] El-Sourani, N., Borschbach, M.: Design and comparison of two evolutionary approaches for solving the Rubik's cube. In: *Proceedings of the 11th international conference on Parallel problem solving from nature: Part II*. PPSN'10, Berlin, Heidelberg, Springer-Verlag, 2010, pp. 442–451.

[3] El-Sourani, N., Hauke, S., Borschbach, M.: An evolutionary approach for solving the rubik's cube incorporating exact methods. In: *Proceedings of the 2010 international conference on Applications of Evolutionary Computation - Volume Part I*. EvoApplicatons'10, Berlin, Heidelberg, Springer-Verlag, 2010, pp. 80–89.

[4] Kociemba, H.: The Two-Phase-Algorithm. `http://kociemba.org/cube.htm`, 2011, [Online; accessed October 06th, 2013].

[5] Korf, R.E.: Finding optimal solutions to Rubik's cube using pattern databases. In: *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*. AAAI'97/IAAI'97, AAAI Press, 1997, pp. 700–705.

[6] Kunkle, D., Cooperman, G.: Twenty-six moves suffice for Rubik's cube. In: *Proceedings of the 2007 international symposium on Symbolic and algebraic computation*. ISSAC '07, New York, NY, USA, ACM, 2007, pp. 235–242.

[7] Rubik, E.: *Rubik's cubic compendium*. Recreations in mathematics. Oxford University Press, 1987.

Evolutionary Generation of Error-correcting Codes

Filip PAKAN*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xpakanf@stuba.sk

Abstract. In today's interconnected world we experience a huge amount of data being transmitted over the network. However, sometimes errors may occur in transmitted data due to electromagnetic interference and thermal noise. Encoding the message in a redundant way by using an error-correcting code gives receivers the ability to recover corrupted data. Larger error-correcting codes allow more messages to be encoded and hence improve communication efficiency. Unfortunately, the generation of error-correcting codes is equivalent to NP-complete problem of finding maximum clique in a graph. An exhaustive search for a solution is not feasible due to the exponential complexity of the problem. One possibility is to employ stochastic optimization algorithms inspired by biological evolution. Evolution is a process where over many generations the optimal solution may be achieved. In this work we analyze recent evolutionary approaches to the generation of optimal error-correcting codes and finding the maximum clique in a graph. We propose a simple evolutionary heuristic for the generation of error-correcting codes. Our solution is based on greedy approach of generating lexicographic codes and the results after several experiments look very promising. We believe that our solution can outperform rather complicated genetic algorithms in terms of speed. Proposed algorithm is a very efficient approach to code discovery.

1 Introduction

When data is transmitted over the network or stored on data storage devices, errors may occur for a variety of reasons such as noise in the transmission or dirt on the storage media. For a binary data this has the effect of flipping 0 to 1 or vice-versa. It is not always possible to ask the sender for retransmission, because some services (namely VoIP, IPTV) have to operate in real time. These services require low latency, high throughput and huge amount of multimedia data to be transferred. Therefore these services mostly rely on UDP protocol which does not slow down the communication by retransmitting corrupted packets but on the other hand reliable data transfer is not guaranteed. This is a perfect scenario for employment of error-correcting codes.

[*] Master degree study programme in field: Software Engineering
Supervisor: Professor Jiří Pospíchal, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

Larger error-correcting codes increase the maximum sizes of transmittable messages and hence improve communication efficiency. Discovering optimal error-correcting codes for different code specifications is a hard problem that we are addressing in this work. We try to apply evolutionary algorithms to this problem and find some suitable evolutionary heuristic for generating error-correcting codes. It is not goal of this paper to find new larger error-correcting code for given parameters since this usually requires months of CPU time. Our goal is to find suitable technique for generating error-correcting codes.

2 Error-correcting codes

Error-correcting codes are mathematical constructs from the field of coding theory that offer communication error detection and correction. They were first pioneered by Richard Hamming in 1950 [2]. Prior to transmission, a message is encoded by adding redundant information to form a code word. The redundant information allows for a distance to be maintained between code words, which allows for error correction.

An error-correcting code is a set of code words. Every error-correcting code is specified by 3 parameters:

1. *n* – length of the code words (number of bits)
2. *M* – number of code words
3. *d* – minimum distance between each pair of code words

An error-correcting code is usually denoted as *(n, M, d)* code of *M* code words, each of length *n* and any pair of code words differs in at least *d* positions. The minimum distance of the code determines the number of errors that may be detected and corrected. In this paper, the distance measure used is Hamming distance – the number of positions at which corresponding bits are different in two binary strings of the same length.

If a corrupted code word that does not match a known code word is received, it is corrected by replacing it with the closest known code word (Figure 1).

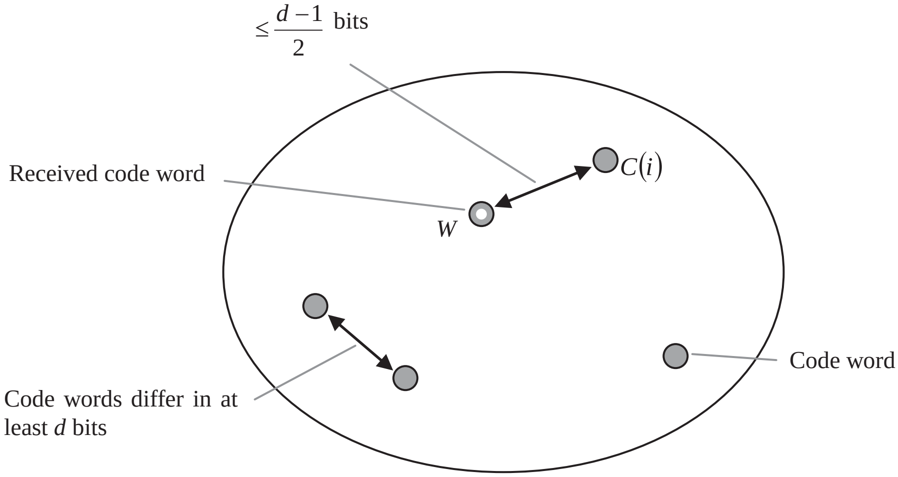

$\leq \frac{d-1}{2}$ bits
Received code word
$C(i)$
W
Code word
Code words differ in at least *d* bits

Figure 1. Correcting of corrupted code word by replacing it with the closest known code word.

3 Generation of error-correcting codes

There are several desirable properties for a code. A good *(n, M, d)* code should have large *M* as this determines the number of different pieces of information that may be represented by the code. It should have small *n*, so that the code words are not too long which would affect communication efficiency. Then it should have large *d*, so that we can detect and correct as many errors as possible. As we can see, these properties are clearly conflicting. The main problem of coding theory is to optimize one of these parameters for given values of the other two. In this work we try to maximize number of code words *M* for given length of the code *n* and minimum distance *d*. The maximum possible number of code words in a code of length *n* and minimum distance *d* is denoted by $A_2(n, d)$.

3.1 Reduction to maximum clique problem

Every time the program looks at a possible solution, it needs to calculate distance between each pair of code words in order to meet minimum distance requirement. The distances between each pair of code words can be pre-computed and stored in compatibility matrix.

We can always assume presence of all-zero code word in the code since any code not containing the all-zero code word is equivalent to one that contains it. So we can generate all possible candidate code words of length *n* and distance at least *d* from all-zero code word. We store results in compatibility matrix *A* in which $a_{ij} = 1$ if code word *i* and code word *j* are compatible (i.e. meet minimum distance requirement), and 0 otherwise. It follows from the definition of compatibility matrix that this matrix is symmetric.

Suppose we wish to find maximum number of code words in a binary code of length 4, minimum distance 2 and the following words are the only possible candidate code words: {0000, 0011, 1010, 1011, 1110, 1111}. The compatibility matrix for this problem is shown in Table 1.

Table 1. Example compatibility matrix.

	0000	0011	1010	1011	1110	1111
0000	0	1	1	1	1	1
0011	1	0	1	0	1	1
1010	1	1	0	0	0	1
1011	1	0	0	0	1	0
1110	1	1	0	1	0	0
1111	1	1	1	0	0	0

Turning compatibility matrix shown in Table 1 into adjacency matrix of a graph is straightforward. Vertices of a graph correspond to code words and two vertices are connected by an edge if two corresponding code words meet minimum distance requirement. A particular graph for a given adjacency matrix is depicted in Figure 2.

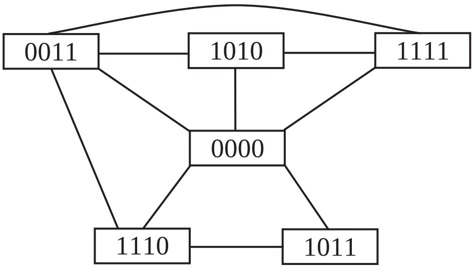


Figure 2. Graph for a given compatibility matrix [1].

Since the resulting code must contain only compatible code words, there must be an edge between each pair of corresponding vertices. So basically we have to find maximum complete sub-graph of a given graph also known as maximum clique. The set of vertices in maximum clique is equivalent to optimal error-correcting code. It is worth noting that maximum clique problem is NP-complete.

3.2 Search space

Let's take a closer look at the size of search space. By assuming the existence of all-zero code word in a code, we may generate all possible candidate code words of length *n* and distance at least *d* from the all-zero word. The minimum distance *d* from the all-zero code word means that every candidate code word must have at least *d* bits set to 1. Number of code words of length *n* and minimum distance *d* from all-zero code word equals to:

$$\sum_{i=d}^{n} \binom{n}{i} \tag{1}$$

Every candidate code word may belong to the code or not, therefore the complexity of this problem is exponential by the number of candidate code words. This exponential complexity prohibits naïve exhaustive search.

3.3 Relationship between parameters

A certain relationship exists between code parameters. By making use of it we may significantly reduce the search space.

Consider $(n, M, 2r\text{-}1)$ code of M code words, each of length n and with minimum distance $2r\text{-}1$. By adding a parity bit we get $(n+1, M, 2r)$ code, thus $A_2(n, 2r\text{-}1) \leq A_2(n+1, 2r)$. Let's note that a parity check adds a bit at the end of every code word. This bit is 1 if the remaining part of the code word has an odd number of 1's and 0 otherwise. As a consequence every code word has now even number of 1's.

Now consider $(n+1, M, 2r)$ code. By removing one bit we get (n, M, d) code where $d \geq 2r\text{-}1$. Therefore we have $A_2(n, 2r\text{-}1) \geq A_2(n+1, 2r)$. Since $A_2(n, 2r\text{-}1) \leq A_2(n+1, 2r)$ and $A_2(n, 2r\text{-}1) \geq A_2(n+1, 2r)$, they must be equal. In conclusion a code of length n and minimum distance $2r\text{-}1$ has fewer possible code words than a code of length $n+1$ and minimum distance $2r$. This reduces the number of candidate code words by half [3].

4 Related work

Several recent papers address the problem of generating optimal error-correcting codes. Considering the fact that this problem is equivalent to a well known maximum clique problem, the research of this problem has a long history.

Haas and Houghten [1] made a comparison of stochastic optimization algorithms for generating of error-correcting codes. Namely they compared hill climbing, beam search, simulated annealing, greedy algorithm for generating lexicographic codes, randomized greedy and several variants of genetic algorithm with two different solution representations. Genetic algorithm with indirect chromosome representation combined with greedy lexicographic algorithm outperformed all other tested algorithms. However only 6 – 9 code words were chosen by genetic algorithm and thousands of code words in resulting code were actually selected by greedy algorithm.

McCarney, Houghten and Ross [5] examined genetic algorithms and genetic programming for generating optimal error-correcting code. The use of genetic programming is novel in this domain. They compared genetic programming to the same type of genetic algorithm as described in previous article which was considered as superior approach among other approaches at that time. In this more recent work they have shown that genetic programming is very competitive approach and there was no clear winner for all codes examined. Both GA and GP provided similar results but running GP took 3 times longer than GA.

Marchiory [4] proposed simple heuristic based genetic algorithm for maximum clique problem. In previous research it had been shown that pure genetic algorithm is not suitable for maximum clique problem. Therefore author introduced a simple heuristic to support optimization of genetic algorithm. The idea is to build a maximal clique by starting with a random subgraph of given graph. This graph is first enlarged by adding some randomly selected nodes, next it is reduced to a clique and finally it is extend using naïve sequential greedy heuristic. Despite its simplicity, this heuristic can outperform all other approaches based on genetic algorithm for finding maximum clique.

5 Proposed algorithm

Our proposed algorithm is based on greedy algorithm for generating lexicographic codes. Lexicographic codes are greedily generated binary error-correcting codes with remarkably good properties. They are generated in lexicographic (dictionary) order. A lexicographic code of minimum distance *d* and length *n* is generated by starting with the all-zero vector and iteratively adding the next vector in lexicographic order of minimum Hamming distance *d* from the vectors added so far. Since this algorithm is deterministic it always gives the same result.

Our algorithm is stochastic variant of lexicographic algorithm. Since the genetic algorithm described in [1] is not very beneficial in generating code words, we try to replace it with something faster. We randomly generate several compatible code words and afterwards we add them to the resulting code. Then we run greedy lexicographic algorithm which will try to optimize the code including those code words that had been inserted at the beginning. By doing this we will slightly modify the standard behaviour of this greedy algorithm because now the greedy must cope with code words already present in result set. This stochastic version will provide various results. Therefore it is wise to introduce multi-start as an integral part of the algorithm and return the best solution found over multiple runs.

Randomized greedy in [1] was heavily modified by adding code words in arbitrary order what in our opinion destroyed the original greedy idea. We still use the main idea of greedy lexicographic algorithm that provides good results and we expect that our modifications can produce better results.

The only parameter of the algorithm that needs to be set is number of randomly selected and inserted code words at the beginning. The optimal value depends on code parameters and thus cannot be once optimized and hardcoded in a program. If the value is too small, we will not influence the run of original algorithm enough and resulting code will not be very different. If the value is too big we will limit greedy algorithm too much and the original greedy idea cannot optimize so well.

6 Implementation

Being able to efficiently compute Hamming weight of binary string is crucial for this application. Fortunately latest processors have implemented the new instruction set SSE4.2 which includes specific instruction for counting number of bits set to 1. This instruction is called POPCNT (population count) and requires 3 clock cycles for execution. The presence of this instruction in processor is indicated by 23rd bit in ECX register after executing CPUID instruction.

According to our measurements, computing Hamming weight with POPCNT instruction is more than 3 times faster than a well known HAKMEM algorithm which is still 10 times faster than a trivial loop implementation. So we gain 30 times speedup comparing to loop implementation. The downside is that only the newest processor's architectures have support for this instruction. Nevertheless, this is by far the fastest way how to compute Hamming weight.

7 Results

Our stochastic lexicographic algorithm clearly outperformed randomized greedy algorithm from [1]. In each test case it produced code with larger number of code words. Moreover our algorithm provides comparable results as the best algorithm from mentioned article.

In addition it is significantly faster than complicated genetic algorithm with indirect chromosome representation combined with lexicographic algorithm. Results in more detail are shown in Table 2.

Table 2. Summary of results.

	$A_2(12, 6)$	$A_2(13, 6)$	$A_2(17, 6)$	$A_2(17, 4)$
Best known	24	32	256	2720
Lexicographic algorithm	16	16	256	2048
Randomized greedy	13	20	129	1693
Stochastic lexicographic algorithm	24	32	256	2218
Genetic algorithm + Lexicographic finish	24	32	256	2238

8 Conclusion and further work

Finding the subset of code words that are compatible among hundreds of thousands candidates is not an easy task. Corresponding graph has almost 2 millions edges. Therefore we are convinced that greedy methods are the only feasible ones for such a big problem. We proposed an algorithm that outperforms some of the previously discovered methods.

The next challenge is improving our algorithm to achieve even better optimization. We also have to propose a way how to automatically optimize parameter of the algorithm – number of code words inserted to the result set.

We have been examining specific algorithms from the domain of coding theory so far. However, in the near future we would like to try out generic algorithms for finding maximum clique in a graph and compare provided results.

Acknowledgements: This work was supported by Grant Agency VEGA SR under the grants 1/0553/12 and 1/0458/13.

References

[1] Haas, W., Houghten, S.: A Comparison of Evolutionary Algorithms for Finding Optimal Error-Correcting Codes. *In Proceedings of the 3rd IASTED International Conference on Computational Intelligence*, ACTA Press, (2007), pp. 64-70.

[2] Hamming, R.W.: Error Detecting and Error Correcting Codes. *The Bell System Technical Journal*, (1950), pp. 147-160.

[3] MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. *North Holland Publishing Company*, (1977).

[4] Marchiori, E.: A Simple Heuristic Based Genetic Algorithm for the Maximum Clique Problem. *In Proceedings of ACM Symposium on Applied Computing*, ACM Press, (1998), pp. 366-373.

[5] McCarney, E.D., Houghten, S., Ross, J.B.: Evolutionary Approaches to the Generation of Optimal Error Correcting Codes. *In Proceedings of the 14th International Conference on Genetic and Evolutionary Computation*, ACM Press, (2012), pp. 1135-1142.

Active Learning Support Vector Machines to Detect Dysfluencies of Non-Fluent Speech

Pavol RUŽIČKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xruzicka@fiit.stuba.sk

Abstract. In this paper we deal with detection of non-fluent speech. We implemented four prototypes for detection of prolongations. Recordings 800 ms long were used as input data, which were processed by using Mel Frequency Cepstral Coefficients (MFCC) and Dynamic Time Warping (DTW). For classification we used Support Vector Machines (SVM) with Radial Basis Function (RBF) kernel. For active learning we used Pool-Based scenario and three strategies: Nearest to the decision boundary, Kernel Farthest First and Balancing Exploration and Exploitation. For evaluation of implemented prototypes we used three metrics: accuracy, precision and recall. We proved that our prototypes with active learning approach reached 90 % of accuracy in comparison to SVM using classical supervised learning approach with 82 % of accuracy.

1 Introduction

Speech is the most effective way of communication between people. Non-fluent speech contains dysfluencies. Commercial automatic speech recognition (ASR) applications reached great performance, but commonly fail in case of non-fluent speech. The first step to improve the ASR's robustness is to develop classifiers able to detect dysfluent events in non-fluent speech. Using speech recognition we can reach more user friendly human machine interaction. Detection of dysfluencies beside of speech recognition in Speech Language Pathology is very in need.

One of most known dysfluency is prolongation. Prolongations in speech are characterized by sound/syllable elongation in speech segments, mainly at the beginning of the words. At first, we must pre-process speech to detect prolongation. The Mel Frequency Cepstral Coefficients (MFCC) at present, are standardly in use for the task of speech parameter extraction. By using MFCC we obtain input parameters, which can be used in classifier. At present, the supervised learning approach is the leading-edge for detection of dysfluencies. In present works for dysfluency detection the Kohonen network [1], the Multilayer Perceptron network [2], the Radial Basis Function (RBF) neural network [2], the Hidden Markov model (HMM) [3] and the Support Vector Machines (SVM) are used [4],[5],[6],[7], [8]. In contrary to supervised learning, the active learning approach, was not studied in recent works for detection dysfluencies in speech. Applying

[*] Master degree study programme in field: Information Systems
Supervisor: Juraj Pálfy, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

active learning for detection dysfluencies may help in Speech Language Pathology to minimalise the time needed to annotate dysfluencies in non-fluent speech.

Active learning approach has many different scenarios to ask queries. The three main settings that have been considered in the literature are Membership Query Synthesis [9], Stream-Based Selective sampling [10], and Pool-Based sampling [11].

In our work we have chosen SVM with Radial Basis Kernel with supervised learning approach and for active learning approach we have studied the Pool-Based sampling.

2 Support Vector Machines

Support Vector Machine (SVM) is a classifier, which classifies m-dimensional vectors into two classes. Training of SVM is based on supervised learning algorithm. In higher dimension in Hilbert's vector space, the hyperplane from the training dataset is constructed. The constructed hyperplane then separates vectors into two classes. Classification of testing dataset is based on decision function. In linearly separable problem the decision function is given by:

$$g(x) = w^T x + b \tag{1}$$

where b is a bias term and w is m-dimensional vector. Vector x is classified as class one if $g(x) > 0$. If $g(x) < 0$, then x is classified as class two.

For non-linearly separable problem we use several kernels. By using kernel we can map input vectors into feature space. Mapped vectors can be linearly separable in feature space. We use Radial Basis Function (RBF) which is given by:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \tag{2}$$

where γ is parameter using to control radius. In Figure 1 we can see utilizing RBF kernel for non-separable problem.

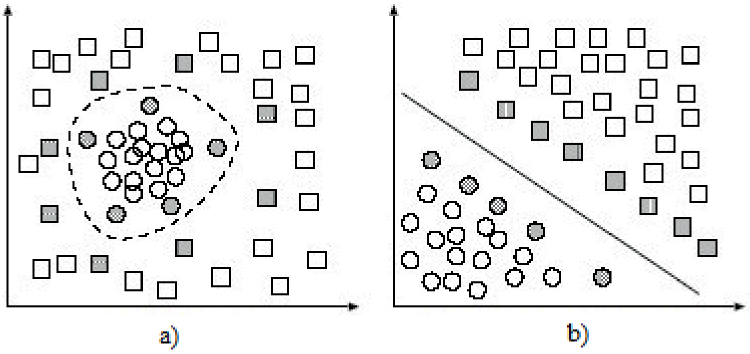


Figure 1. a) Example of RBF. b) RBF mapping on non-linearly separable problem into feature space.

2.1 Active Learning

In this paper we choose Pool-Based Scenario. This scenario can be used with SVM [12]. In this scenario we have small training dataset and we used this set to create decision function. The query can request labels of few instances from the whole dataset (i.e. pool). The main issue with active learning in this setting is finding a way to choose good queries from the pool [12].

Queries strategies can be differ, but in this paper we deal with three of them: Nearest to the decision boundary [13], Kernel Farthest First (KFF) [14] and Balancing Exploration and Exploitation [14].

In *Nearest to the decision boundary strategy* we have built classifier C_i by using all labelled instances. Now we try to find the nearest unlabelled instance to decision boundary of C_i. We then test all unlabelled instances in the pool. The distance we measure is in feature space and is given by:

$$|w_i \cdot \Phi(x)| \tag{3}$$

where w_i is a unit vector of hyperplane and $\Phi(x)$ is instance mapped into feature space (i.e. feature vector). We try to query the nearest instance, because we supposed that it has the highest uncertainty for classifier C_i.

KFF strategy tries to query the most different instance from all labelled instances. Due to this reason it will query the farthest instance from labelled instances. Let L is a set of labelled instances and U is a set of unlabelled instances. Now we find $x \in U$, which is farthest from all $y \in L$. The distance we compute is the Euclidean distance given by:

$$d(x,y)^2 = \|\Phi(x) - \Phi(y)\|^2 = K(x,x) + K(y,y) - 2K(x,y) \tag{4}$$

where $K(x,x)$ is kernel and $\Phi(x)$ is feature vector. Using this strategy we query x computed by:

$$\underset{x \in U}{\operatorname{argmax}} \left(\min_{y \in L} \|\Phi_K(x), \Phi_K(y)\| \right). \tag{5}$$

Balancing Exploration and Exploitation strategy is based on probability of exploration and exploitation. Exploration is executing by KFF strategy and exploitation is executing by Nearest to the decision boundary strategy. After each step strategy is considering “success” to adjust probability of step. Let us choose KFF strategy to query instance for the first time. Before querying the instance we have for instance hypothesis h. After querying we reach hypothesis h'. The achieved success is then given by function:

$$d(h,h') = \frac{\langle H, H' \rangle}{\|H\|\|H'\|} \tag{6}$$

where H is a set of hypothesis for instances before querying and H' after querying. If KFF strategy was successful, then we do not change probability, but if was not then we decrease probability. Computing probability is given by:

$$p' = max(min(p\lambda \exp d(h,h'), 1 - \epsilon), \epsilon) \tag{7}$$

where p is probability before querying and p' is a new probability. ϵ is a parameter for upper- and lower-bounds and also λ is a learning rate for updating probability.

3 Results

Like authors [5], we used audio recording of stuttered speech from The Database University College London Archive of Stuttered Speech (UCLASS)[1]. We selected 70 (35 fluent, 35 non-fluent) recordings by 800 ms length [2]. All the speakers of 70 recordings were patients with stuttering of male gender. Speakers were in age range from 8 to 17 years. Speech recordings were in a Waveform Audio File Format (WAV). At first, we converted all recordings to have a

[1] Available at: http://www.uclass.psychol.ucl.ac.uk/.

consistent 16 000 Hz. sampling frequency. Then we used MFCC to extract characteristic features of speech. To obtain these coefficients we used Hamming window of length 32 ms along with the shift between adjacent frames was 16 ms. Then we obtain 20-dimensional vector of coefficients per every Hamming window. Like authors [4] we used Dynamic Time Warping (DTW) to score matching of MFCC vectors. We obtained 48 dimensional feature vector for every 800 ms recording. These features were used as input for classifier SVM with RBF kernel. We used for supervised learning a library LibSVM for SVM training and for active learning we found SVM implementation Active SVM of authors [14].

In this paper we change size of training set and compare reached results for SVM and Active SVM. Training set has always same number of fluent and non-fluent recordings. In testing set were the remaining 70 recordings. For active learning the training set was always smaller, because we query for six instances.

For comparison of result we used metrics: accuracy, precision and recall. Accuracy is given by:

$$accuracy = \frac{number\ of\ true\ positives + number\ of\ true\ negatives}{number\ of\ all\ instances} \tag{8}$$

Precision is given by:

$$precision = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + false\ positives} \tag{9}$$

Recall is given by:

$$recall = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + false\ negatives} \tag{10}$$

In Figure 2 we can see accuracy which we have achieved. Active learning approach was better for training sets, which have 14 and more instances. SVM with supervised learning has achieved approximately 82% for all sets. The best approaches were Active SVM with Nearest to decision boundary and Balanced exploration and exploitation strategy. We have achieved 90% accuracy using these strategies.

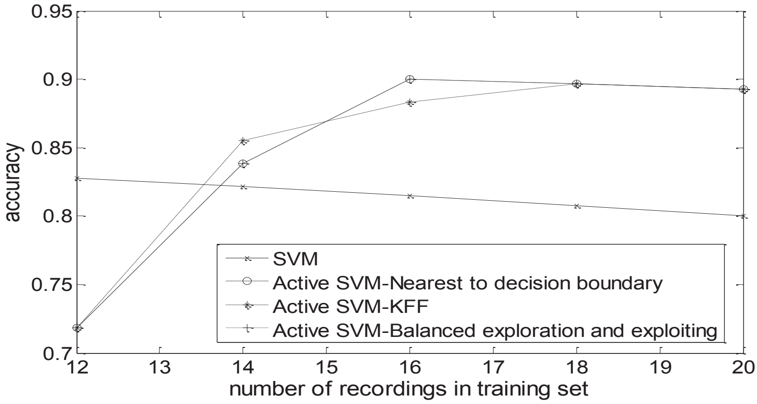


Figure 2. Achieved accuracy for classifiers.

In Figure 3 we can see precision which we have achieved. For Active SVM precision has decreased for training sets where accuracy was high. The highest precision was 93.7% for all strategies. In Figure 4 we can see recall which we have achieved. Active learning approach was better for training sets, which have 14 and more instances like accuracy. The best recall was 96.2% for Nearest to decision boundary and Balanced exploration and exploitation.

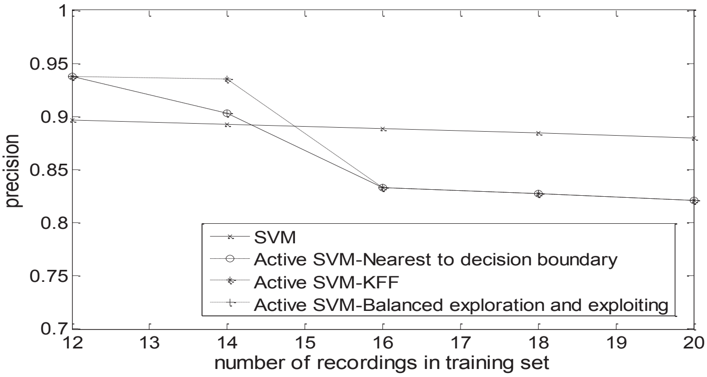


Figure 3. Achieved precision for classifiers.

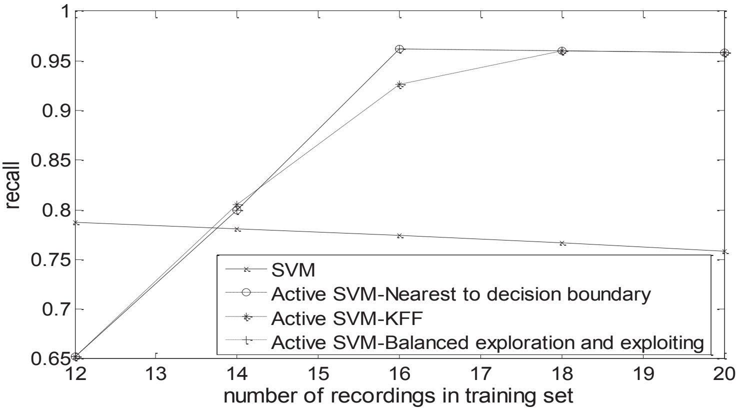


Figure 4. Achieved recall for classifiers.

4 Conclusion

In this paper we have implemented four prototypes of SVM to detect of dysfluencies. For three of them was used active learning approach. The best results we have achieved by using Nearest to the decision boundary and Balanced exploration and exploitation. These two strategies have achieved same results, because Balanced exploration and exploitation decide only for first query for KFF strategy. Then querying instances were the same as for Nearest to the decision boundary strategy.

We have compared prototypes based on SVM by three metrics: accuracy, precision and recall. The best strategy for this theme was Nearest to the decision boundary.

We proved that by using active learning strategies we can achieve superior results compared with supervised SVM learning. We have reached 90% accuracy when the training set contains 16 speech recordings of length 800 ms. This result we have achieved by using SVM as well RBF kernel. We extracted from input audio data set MFCC speech parameters and applied DTW template matching to found similar MFCC.

In future work we try to obtain more labelled recordings and extend our prototypes for longer speech recordings and, we plan to design and implement our new querying strategy.

Acknowledgments: This contribution was supported by Grant Agency VEGA SR under the grants 1/0553/12 and 1/0458/13.

References

[1] Szczurowska, I., Kuniszyk-Jóźkowiak, W., Smołka. E.: *Speech nonfluency detection using Kohonen networks*. Neural Computing and Applications Vol. 18, (2009), pp. 677-687. ISSN: 1433-3058.

[2] Swietlicka, I., Kuniszyk-Jóźkowiak, W., Smołka. E.: *Artificial Neural Networks in the Disabled Speech Analysis*. Advances in Intelligent and Soft Computing Vol. 57, Springer, Berlin, (2009), pp. 347–354. ISSN: 1867-5662.

[3] Wiśniewski, M., Kuniszyk-Jóźkowiak, W., Smołka. E., Suszyńsky, W.: *Automatic detection of prolonged fricative phonemes with the hidden Markov models approach*. Journal of medical informatics & technologies , (2007), Vol. 11. ISSN: 1642-6037.

[4] Ravikumar, K.M., Rajagopal, R., Nagaraj, H. C.: *An Approach for Objective Assessment of Stuttered Speech Using MFCC Features*. DSP Journal Vol. 9, no. 1, (2009) pp. 19–24. ISSN: 1753-2358.

[5] Pálfy, J., Pospíchal, J. *Recognition of repetitions using Support Vector Machines*. Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA). IEEE Poland Section, Poznan (2011). ISBN: 978-1-4577-1486-3

[6] Che-Chuang, L. Lin-Shan, L. *Improved Features and Models for Detecting Edit Disfluencies in Transcribing Spontaneous Mandarin Speech*. IEEE Transaction on audio, speech, and language processing, (2009) Vol. 17, No. 7. ISSN: 1063-6676.

[7] Ganapathiraju, A. *Applications of Support Vector Machines to Speech Reco.gnition*. IEEE Signal Processing, (2004), Vol. 52, pp. 2348-2355. ISSN: 1053-587X

[8] Hasegawa-Johnson, M., Gunderson, J., Perlman, A., Huang, T.: HMM-based and SVM-based recognition of the speech of talkers with spastic dysarthria. Proc. IEEE, (2006), Vol. 3. ISBN 1-4244-0469-X.

[9] Angluin, D.: *Queries and concept learning*. Machine Learning, (1988), Vol. 2, pp. 319–342. ISSN: 0885-6125.

[10] Altlas, L., Cohn, D., Ladner, R., El-Sharkawi, M. A., Marks, R. J.: *Training connectionist networks with queries and selective sampling*. Advances in Neural Information Processing Systems (NIPS) Vol. 2, pp. 566-573. Morgan Kaufmann Publishers Inc. San Francisco, (1990). ISBN: 1-55860-100-7.

[11] Lewis, D., Gale, W.: *A sequential algorithm for training text classifiers*. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, (1994) pp. 3–12. ISBN: 0-387-19889-X.

[12] Tong, S., Koller D.: *Active Learning: Theory and application.* University Stanford, (2001), pp. 24-31. ISBN: 0-493-40474-0.

[13] Tong, S., Koller, D.: *Support vector machine active learning with applications to text classification.* Journal of Machine Learning Research, (2001), Vol. 2, pp. 45–66. ISSN: 1532-4435.

[14] Osugi, T., Deng, K., Scott, S.: *Balancing Exploration and Exploitation: A New Algorithm for Active Machine Learning*. Proceedings of ICDM, (2005). ISSN: 1550-4786.

Random DNA Read Generator

Juraj Šimek*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
simekjuraj@gmail.com

Abstract. Research of DNA is an important part of the medicine because many diseases are caused by DNA mutations. To study DNA, is necessary to sequence the genome using DNA sequencers, which produce a set of short overlapping reads. These reads have to be assembled to original DNA molecules, which were sequenced using assemblers. New algorithms for DNA assembly are proposed to improve quality of results or time and space requirements. Newly developed assemblers have to be tested using a set of reads where assembled sequence is known. Parameterised random read generator introduced in this work is able to generate data suitable for testing assembly algorithms. The advantage of the generator is its ability to create reads from the genome with selected characteristics.

1 Introduction

Deoxyribonucleic acid (DNA) encodes genetic information of all organisms. DNA consists of small biopolymers called *nucleotides*. Four types of nucleotides, which are building blocks of DNA, are distinguished: *adenine* (A), *guanine* (G), *cytosine* (C) and *thymine* (T). DNA usually occurs in form of a double stranded molecule (called *DNA molecule*). Individual strands have an opposite direction and they are paired according to the *Watson-Crick rule*, which says, that the adenine pairs with the thymine and the cytosine with the guanine [1]. Forward direction of DNA molecule is from 5' to 3' (the Figure 1) [2]. Number 5 and 3 signifies carbon atoms in nucleotides. It is possible to see DNA molecule as a tuple *(m,n)*, where *m*, *n* are words from alphabet ∑,, ∑ = *{A, T, G, C}* and *m = reverse(n)*, where *reverse(n)* is a reverse complement operation of strand n, which can be created by changing direction of strand n and replacing each nucleotide for complement nucleotide according to Watson-Crick rule (for example reverse complement of a strand *ATGCCTGC* is *GCAGGCAT*). DNA determines structural characteristics of every cell and controls chemical processes which are running in their bodies. Mutations of DNA lead to various genetic diseases. Correlation between some special kinds of DNA mutations and a cancer was also reported [4]. And this is why research in this field is very important. To study structure and characteristics of DNA, it is necessary to sequence the genome of selected organism using DNA sequencer. *DNA sequencer* is a machine which is able to read a certain segment of DNA. Genetic information in cells doesn't occur as a single long DNA molecule, but this information is separated into several shorter DNA molecules. These DNA molecules are called *chromosomes* [1]. Number

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Gabriela Kosková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 311–316.

of chromosomes is differs among species. In case of a human, the number of chromosomes is 46. These chromosomes are paired, so there are 23 kinds of chromosomes in a human cell [6]. Differences between two chromosomes of the same type are very small.

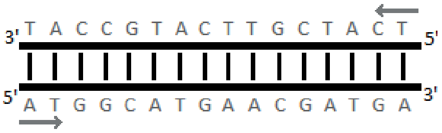

3' T A C C G T A C T T G C T A C T 5'
5' A T G G C A T G A A C G A T G A 3'

Figure 1. DNA structure is determined by order of nucleotides. Forward direction of each strand is marked by the arrow. Opposite strands are complementary according to Watson-Crick rule.

Unfortunately, DNA sequencers are not able to sequence the whole chromosome from its beginning to the end [2]. Output of sequencer is just a set of short overlapping pieces of DNA sequences (called reads), whose location and orientation in chromosome is unknown. When chromosome's order of nucleotide is unknown, it is not possible to predict from which strand and from which part of this strand the read was sequenced. It is important to order reads into the correct order and find overlaps between them. Various bioinformatics algorithms were developed for this reason. It is still important to develop modern algorithms with a higher precision and performance of assembly. It is important to have a parameterized generator of reads for the reason of testing these algorithms. The aim of this article is to introduce a simple random read generator, which will generate a selected number of chromosomes (an order of nucleotides in chromosomes will be random) and generate reads belonging to them. Next aim of this article is to introduce a simple algorithm for DNA short read assembly, which assembles short reads that are sequenced from an organism or generated by a generator, for a purpose of testing the generator.

1.1 DNA Sequencing

DNA sequencing is a process of obtaining short reads from chromosomes of some organism [2]. Input of sequencer is a set of chromosomes and output is a set of short reads. Reads are always sequenced from a one strand of a fragment (from some chromosome) in a forward direction (from 5' to 3') [2]. It is possible consider the reads to be some substrings of chromosomes. Several techniques for DNA sequencing have been developed in recent years. The first and most famous techniques for DNA sequencing is *Sanger's sequencing* [5] based on the chain termination sequencing. Due to a high price of this technique and an amount of time needed, it is not used for the whole genome sequencing. Advantage of Sanger's sequencing technique is generation of longer reads (to 1000 nucleotides). *Next-generation* sequencing methods, which are used currently (like *pyrosequencing, reversible-terminator sequencing, or sequencing by ligation*) generate shorter length reads (35 – 400 nucleotides) [2], but their advantage is much lower price in comparison with Sanger's sequencing. On the other hand the shorter length of reads makes the read assembly problem more complicated.

It is also possible to generate paired reads using some modern sequencers. These reads are read from the single fragment of DNA molecule. Techniques which can generate this kind of reads are able to read both ends of these fragments (end of each strand), which means that every paired read consists of two parts, where each part of the paired read is from another of fragment's strands. Distance between these parts is called an *insertion length* and this length is approximately known (see the Figure 2) [2].

1.2 Short read assembly problem

The short read assembly problem has, as its input, a large set of short reads. The output is a set of DNA molecules, from which reads were probably sequenced. Actually, it is inverse action of DNA sequencing. Short read assembly problem is complicated by many issues. *Repeated sections* of

DNA within chromosomes (or among chromosomes) cause significant problems [2]. When length of a repeated section is much longer than the read length, it is almost impossible to predict the number of repeats. Various algorithms for short-read assembly differ in technique of repeat identification and their correct incorporation into output assembly. It is important to use paired reads, which can help determine the correct number and location of repeats in genome, for correct manipulating with a data containing repeats. Another problem which complicates assembly is caused by *sequencing errors* [2]. It is also probable, that the sequencing machine will incorporate errors into data during sequencing. Some nucleotides can be misread, some of them can be accidentally skipped or some of them can be mistakenly inserted into the sequence, which is read. Currently used algorithms have to find sequencing errors and correct them. Important parameter of sequencing for short read assembly problem is its coverage. *The coverage of the genome is the ratio of the total length of all reads to the total length of the genome. Assembly projects using chain termination methods of sequencing have typically generated 6-fold to 20-fold coverage* [2]. Coverage of projects using next-generation sequencing is much higher.

Several approaches to solve sequence assembly problem were introduced. Assemblers for de novo short read assembly are mostly based on the *de Bruijn graphs*. Benjamin Jackson described de Bruijn graph with following words: *Each node in the graph corresponds to a unique k-mer (length k string) present in some input sequence or its reverse complement. A directed edge connects two nodes labelled a and b, where is a string of length k − 1, if and only if a b is present in some read. This graph is a subgraph of a de Bruijn graph of k-mers, and each input sequence a path* [2]. Reads usually don't represent nodes of graph, but they are usually cut to the k-mers of smaller length. The sequence assembly problem is then transformed to the problem of finding the shortest tour of the graph that includes each edge (this property is satisfied by an *Eulerian path*) [3]. De Bruijn graph helps us find the shortest common superstring where all reads are present. Another approaches use *bidirected de Bruijn graphs*, which are modified to consider a double stranded nature of DNA. Last approach mentioned is based on *k-sting graphs* which are derived from de Bruijn graphs. De Bruijn graphs, bidirected de Bruijn graphs and k-string graphs are closely described in Benjamin Jackson's thesis [2].

2 Random read generator

Random read generator introduced in this work generates a required number of chromosomes. Order of nucleotides within these chromosomes is generated randomly according to the *GC-content*. GC-content is a percentage occurrence of nucleotides G and C to occurrence of nucleotides A and T. It is possible to choose between *single stranded* or *double stranded* molecule generation. Program counts distribution of nucleotides and randomly generates their order within all chromosomes after its launching. Random read generator is written in C language as a console application and for random generation of data it uses C's `rand()` function. Program always generates only one strand. If the double stranded molecule generation is selected, program computes the reverse complement of the generated strand for each chromosome and saves it into the data structure storing data of a corresponding chromosome. Then a next phase is launched, where the reads are generated. Input for this phase is a set of chromosomes generated in previous step; output is a set of reads stored in the output file selected in the program settings. Program randomly chooses chromosome and its strand in case of non-paired reads generation. Then the start position of read is chosen and the strand is read. It is possible to generate reads only from one single strand or from both strands of DNA molecule. The length of reads can be constant or variable and is passed by a parameter. Number of the reads is counted according to the coverage parameter. Program allows generation of paired reads, too. Insert length can be constant or variable as well as the length of reads. Paired reads are generated in different way in contrast to single reads. Every paired read consists of two parts. Each part is sequenced from an opposing strand. At first, chromosome is chosen like in the case of single strands. Then start positions of both parts are chosen, firstly start position within first strand is chosen randomly and after addition of first part length, insert length and simple conversion of counted position to the second strand, start position of second part within second strand is found. For this reason, generating of paired reads requires double stranded molecules. Parts of paired reads can be inward or outwardly directed (as the Figure 2 shows). Next important property of this random read generator is a simulation of read errors. It is possible to set the probability of an error occurrence and distribution of error types. Three types of errors are distinguished, insertion errors (wrong nucleotide is mistakenly incorporated into the result), deletion errors (some nucleotide is not read) and mismatch errors (nucleotide was read incorrectly). All settings of DNA generator are stored in a settings file, which is provided to the program. This file contains various fields which have to be set before generating reads. Description of these fields is stored in settings file as a comment (comment line starts with symbol #). File with reads is generated and a reference sequence (order of nucleotides of all generated chromosomes in first step) is stored for the purpose of comparison assembler's results at the end.

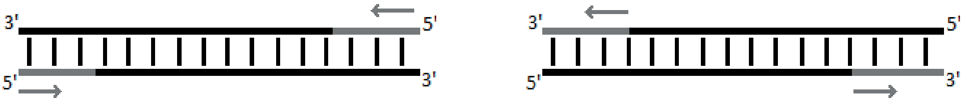


Figure 2. Picture shows different directions (inward or outward) of two parts of paired reads. Parts of paired read are grey. They are always read in a forward direction from opposing strands. Insertion length is a space between two parts of the paired read.

3 Simple short read assembler

Simple genome assembler is based on de Bruijn graphs. At first a *k-mer length* and a *read length* are inserted as input of the program. Also name of the input file with reads and name of the output file with assembled contigs (*contig* is term for a set of reads which are concatenated to one superstring) are inserted. Input file with reads is read and every read is split into k-mers. These k-mers are transformed into nodes. Each node stores two strings of length k-1. First string is a prefix (first k-1 letters from k-mer) and second string is a suffix (last k-1 letters from k-mer). At first, list of existing nodes is searched to find out whether the same node exists or not. If the same node exists, program continues with next k-mer to node transformation. If a newly created node is unique, it is added to the list of nodes and program will generate edges between this node and all other nodes that fulfil the condition for an edge creation described in following sentences: All edges are directed. Every node has a list of outcoming edges. Every edge has a pointer to the node, where it is pointing to and a flag whether it was traversed or not. Edge from node A to node B is created if and only if suffix of the node A is the same as prefix of the node B. When input file with reads is processed and all the reads are transformed into edges, the graph is ready for a graph traversal. Finding the path in graph is simplified. Assembler does not search Eulerian path in graph. It finds a node with no incoming edges instead of it. Program continues from this node to another node (which belongs to the first edge in the list of outcoming edges of current node, which has not been traversed yet). Edge connecting them is marked as traversed. During traversing of the graph a first letter of each node's prefix is stored to the output file. When the program cannot continue to another node, the k-mer of the last node is stored into the output file and it tries to find the other node with no incoming edges. If node with this property does not exist, program finishes. The number of contigs is the same as the number of nodes with no incoming edges. Assembler does not work with paired reads. It is designed to work properly only with reads of the same length read from one single stranded DNA molecule with no sequencing errors and repeats. Coverage of the sequenced genome should be sufficiently high. This simple assembler is written in a C language as a console application to reach a higher performance of assembly.

4 Experiments

Simple random read generator was tested using simple short read assembler described in the chapter 3 of this article. Selection of assembly test result is viewed in the Table 1.

Table 1. Selected results of test.

Sample	Length of sequenced chromosome	Read length	Coverage	k-mer length	Number of contigs	Length of assembled sequence	Assembly time (ms)
1	100	30	40	5	0	0	2
2	100	30	40	6	1	43	3
3	100	30	40	19	1	100	2
4	1000	30	40	7	0	0	265
5	1000	30	40	10	1	40	304
6	1000	30	40	36	1	1000	228
7	1000	70	40	41	1	999	161
8	1000	70	40	67	41	X	33
9	1000	60	100	11	1	147	652
10	1000	60	100	21	1	1000	490
11	1000	60	100	31	1	1000	369
12	1000	60	100	57	2	1000	59
13	10000	100	100	33	1	10000	45390
14	10000	100	100	83	1	10000	13438
15	500	70	50	36	1	351	50
16	400	40	50	21	1	200	69
17	400	40	50	37	2	400	50
18	200	40	50	21	2	199	421
19	300	40	5	11	1	216	4
20	300	40	5	21	2	299	8
21	300	40	5	31	13	X	2

Samples 1 – 14 were generated as the assembler requires for a correct assembly. Sample 15 was generated with sequencing (misread) errors of rate 0.0001%. Samples 16, 17 contained two chromosomes sequenced. Sample 18 was sequenced from a double stranded molecule and samples 19 – 21 was sequenced with a low coverage. X in the table means, that genome was not assembled as expected due to high number of contigs.

Assembler was able to assembly about 100% of length of sequenced data, when a single stranded data sequenced from one chromosome with no errors and repeats were used. Results confirmed, that the assembler is able to work properly only with correctly chosen k-mer length. If the length is too high (samples 8, 21), a lot of contigs are present in a result, which means, that it is not possible to recover the whole genome as was expected (in table this fact is marked with X). Too short lengths mean that assembled contig does not match reality (samples 1, 4). A good selection of k-mer length is very important for a speed of assembly, too. Longer length produces fewer nodes in the graph, which means shorter time needed for assembly.

Data sets which contain more than one chromosome sequenced (16, 17), reads with sequencing errors (15), reads from a double stranded single chromosome (18), or reads of small coverage (19, 20, 21), would cause problems to DNA assembly using simple assembler, as was confirmed by the experiments. Impact of these problematic data sets can be decreased by a manipulation with a k-mer length. Sometimes it was possible to recover almost 100% of the sequenced genome with a good selection of k-mer length.

Result of the tests confirmed, that the generator is working correctly. Generated reads were properly assembled to the one contig matching sequenced genome, when these reads were generated according to conditions mentioned above. Paired read generation is based on the similar principles like generation of single reads. The only difference is that two parts of paired read are separated by an insertion length. When paired reads with the constant insertion length were generated, separate tests confirmed constant distance between parts of every paired read.

The sequence generation is fast enough. Chromosome of length 1 million nucleotides was generated in 48 milliseconds. Generation of chromosome of length 1 billion nucleotides took about 55 seconds. Operation of the reverse complement is approximately 10 times faster than generation. Program was tested on the machine with Intel Core i5-4200U processor and 4 GB RAM. Speed of sequencing depends on coverage, read length, whether paired reads are generated or not, count of chromosomes and their lengths, but sequencing is incomparably faster than real sequencing using sequencers.

5 Conclusions and future work

Random read generator is able to generate required number of chromosomes. Its nucleotide order is generated randomly according to selected GC-content. Simulation of DNA sequencer is launched with these chromosomes and reads are generated. Generator is able to generate paired reads, too. Generated data are suitable for testing of newly designed assembly algorithms. Actually, DNA has lots of characteristics which have not been covered by this random read generator. DNA of organisms contains various kinds of repeat subsequences, which are problematic for assembly. Due to the random character of nucleotide order in generated reads, occurrence of long repeats is highly improbable. Next version of DNA generator will not generate random DNA, but these DNA molecules will have structure determined by settings of repeats. Next possible improvement is to make a graphical environment for building a settings file. Generator will remain in the console mode because of the generation speed enhancement.

Simple assembler presented in this article is able to recover about 100% of DNA which was sequenced. Several limitations are present. Reads should be generated from the single stranded DNA; length of DNA strand should not be long. The reads should be sufficiently long and the coverage should be of higher value, too. The coverage higher than 30 and reads about 100 nucleotides long are recommended. If sequencing errors are present in the sample, probability of correct DNA molecule recovery is very low. Also the assembler is not able to work with reads from chromosomes with repeats. These repeats will cause occurrence of cycles in the graph and the graph traversing will stop after it passes through first cycle. Improvement of the assembler is also possible. Assembler could be faster using hash tables to find out whether currently processed node is in the list of nodes. Also approach using Eulerian path will provide better results.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0752/14.

References

[1] Alberts B., Johnson A., Lewis J., et al.: Molecular Biology of the Cell. 4th edition. Garland Science, New York, (2002). [Online; accessed January 23, 2014]. Available at: `http://www.ncbi.nlm.nih.gov/books/NBK21054/`

[2] Jackson B.: Parallel methods for short read assembly. PhD thesis, Iowa State University (2009).

[3] Pevzner A. P., Tang H., Waterman S. M.: An Eulerian path approach to DNA fragment assembly. In: PNAS, (2001), vol. 98, no. 17, pp. 9748-9753

[4] Rasnick D., Duesberg H. P.: How aneuploidy affects metabolic control and causes cancer. In: Biochemical Journal, (1999), vol. 340, pp. 621-630.

[5] Sanger F., Nicklen S., Coulson A. R.: DNA sequencing with chain-terminating inhibitors. In: Biochemistry, (1977), vol. 74, no. 12, pp. 5463-5467.

[6] Ensembl: Whole genome. [Online; accessed January 23, 2014]. Available at: `http://www.ensembl.org/Homo_sapiens/Location/Genome`

Task Scheduling in Distributed Computational System

Matúš UJHELYI*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
ujhelyi.m@gmail.com

Abstract. Using of distributed systems becomes more important, because of increasing of amount of data that has to be processed. Effective task scheduling across these systems is one of the biggest challenges for the creators. This paper focuses on the new characteristic of hosts in volunteer computing systems called assessment of host performance. It is based on the standard characteristics of hosts in Boinc system, the main representative of volunteer computing systems. The main goal of paper is to present the new class based algorithm which uses defined characteristic. It also refers about the result of simulation and the performance of this algorithm in compare with Boinc scheduling FCFS algorithm.

1 Introduction

There are many groups of computational tasks, which cannot be computed in reasonable amount of time on single host machine. Many of these tasks can be split into smaller parts, which could be computed separately. This is the idea of distributed computational systems. There are many types of distributed computational systems nowadays. One of main group is represented by the volunteer computing systems and main representative of this system is called BOINC [4].

Volunteer computing systems are based on hosts that voluntarily accommodate the computational capacity to the system. These hosts lately claim for the tasks and lately deliver the results. It is not a trivial problem which tasks should be sent to which hosts. There are many differences between tasks and also between hosts. Hosts differ in many aspects such as CPU performance, memory size, and also the amount of time that is used for computing. Tasks differ in size and amount of operations needed to correctly complete them. These differences strongly affect the scheduling process. Right scheduling plan should reduce the total amount of time needed for completion of all tasks.

The purpose of this document is to present the new characteristic of the host that can be used in scheduling algorithms and also to present the algorithm based on this characteristic.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 317–322.

2 Related work

There are many directions in task scheduling improvements. One of the primarily is based on genetic algorithm. The interesting representative is the special genetic algorithm based on IF-THEN-ELSE rules [1]. The sense of these rules is to decide whether it is possible to assign specific task to the specific host. These rules are created from the Boolean logic expressions and compiled in to the python code. Lately are connected to the Boinc scheduling mechanism. Performance of the rules was measured using simulation in environment called SimBA created by the authors of the algorithm.

The next approach is based on using different heuristics. Some of them try to define the reliability of host based on previous supplied results [2]. Other heuristic is based on relative cost of tasks defined as approximation of differences of counting times of tasks on hosts [3].

Scheduling problem could be also known as a standard assignment problem as it is discussed in linear programming optimization technique. It is based on static minimal optimization technique. There are many algorithms used for such problems e.g. Hungarian method [6].

3 Boinc scheduling mechanism

The main Boinc scheduling process is based on simple First-come-first-serve algorithm with some modifications.

Volunteer computing projects must deal with erroneous computational results. Scheduling mechanism has to be able to settle up with these results. These results arise from malfunctioning computers (typically induced by over clocking) and occasionally from malicious participants. Boinc provides support for redundant computing, a mechanism for identifying and rejecting erroneous results. A Boinc project can specify that N results should be created for each task. Once $M \leq N$ of these have been distributed and completed, an application-specific function is called to compare the results and possibly select a canonical result [4]. Such mechanism is one of the main differences between standard distributed computational systems and volunteer computing systems.

Other modifications are based on rules which operate over characteristics of hosts. One of them is locality scheduling, which sends same tasks to the hosts with near geo location. Another modification is homogeneous redundancy, when task is sent to hosts with the same CPU type or operational system [4].

3.1 Boinc task and host characteristics

Every Boinc project collects many characteristics about hosts and tasks. These characteristics could be used for scheduling improvements. The most important are:

- Estimate flops – amount of assumed floating point operations to finish the task
- Estimate time – amount of assumed time to finish the task on concrete host
- CPU time – amount of time spent by host counting the task on CPU
- Elapsed time – total amount of time between assigning the task and delivering the result [5].

4 New Class Based Algorithm

As we described in the previous section, Boinc uses the standard FIFO algorithm with modifications and also provide characteristics of hosts. To define the new algorithm we had to somehow characterize the hosts. We use the Boinc characteristics and based on them we designed the assessment of host performance factor for each host connected to the system. This characteristic is the main factor of the new class based scheduling algorithm.

4.1 Assessment of host performance

The characteristic assessment of host performance as the main characteristic of hosts is defined as follow:

$$host\_performance = \frac{\sum_{k=0}^{j} \frac{cputime(k) * peakflops(k)}{elapsed\_time(k)}}{j} \qquad (1)$$

The assessment of host performance (see Formula 1) is the factor defined for each host and represents the supplied amount of performance of host. It can hold the value from 0 to 1.0 in maximal range. After multiplying with CPU performance, the result is the average amount of floating point operations that the host delivers during the one quota of time. It is mathematically defined as ratio of CPU times and elapsed times of all tasks *k* computed on the host. It covers up the time when the hosts do not compute and also covers when the hosts deliver bad results. Both of these situations reduce the value of the assessment of host performance.

4.2 Algorithm

The assessment of the host performance is the main factor of our class based algorithm. It characterizes the hosts. For characterizing the tasks, we decided to use estimates flops of tasks form the Boinc system. The main idea of algorithm is to divide the hosts and tasks in to the separate groups or classes and map the classes of hosts to the classes of tasks. Dividing is based on the host performance characteristic. If host connects and ask for work, system determines his class, based on his assessment of his performance, and it gives him the task from the mapped class. If group of tasks is empty, it gets the task from the class with lower level of average estimate flops. If the lowest one is empty the rescheduling takes place. Choosing the concrete task to send is based on sorting the tasks in concrete class and taking the most difficult one.

Creating classes has to be done on sorted hosts (or tasks). When they are sorted, one of the two approaches is used:

- Statistic method - based on median characteristic of sorted items. Groups of tasks and hosts are created by median. The sorted items are recursively divided into groups by their median values until the requested groups count is reached.
- Binary groups – tasks are divided by median but hosts are divided by medium value, where the dividing is done only on the faster group.

Our algorithm also uses redundant computing mechanism. It sends the same task more times to deal with erroneous results. The important part of the algorithm is related to updating the values of the assessment of the host performance. It is recalculated if the host delivers the result. It is also recalculated if the timeout is reached or the error result is delivered.

The algorithm is inspired by the standard multilevel queue scheduling algorithm used in process scheduling on single CPU machine but it is extended on volunteer computing distributed systems with many hosts.

The main idea of the algorithm is to reduce the probability of the situation that long tasks are executed by slow hosts at the end of the computational process. It is mainly insured by grouping mechanism and updating process of the host characteristic based on previous delivered results.

5 Evaluation

It is not possible to evaluate the algorithm by using it in fully operational system. It should take a lot of time to get results. Because of this, we decided for the software simulation. We created a simulator with the implementation of the new algorithm and the old Boinc scheduling algorithm.

It allows creating hosts and tasks using standard statistics distributions. It also allows visualizing the statistic distributions of connected hosts and remained tasks to compute. We also had to define the values for the assessment of the host performance for the simulation. It had to be defined rationally so that the simulation should represent the real computational process.

5.1 Assessment of host performance – statistic distribution

We defined that the main simulation factor, the assessment of host performance, is generated from the follow statistic distribution:

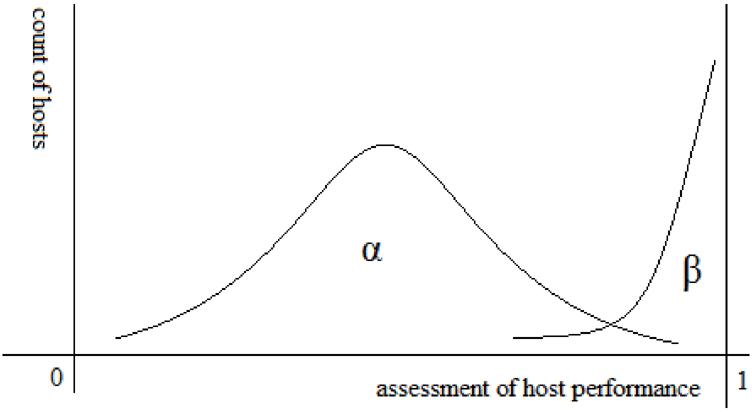


Figure 1. Distribution of the assessment of host performance.

The distribution (see Figure 1) is divided in to two areas. The alpha are represents the casual hosts and the values are from normal distribution. Their machines compute the tasks during the users normal PC usage e.g. during work, or gaming or web browsing. The machines are also not used for 100% for computing tasks. The value of assessment of host performance varies from 0 to 0.8. It means that the computer is used for counting tasks for 0 to 80% of total time. The beta area represents the special hosts that are connected to the system at almost 100% of time. They are used mainly for Boinc computing at the value is nearby 1.0. It means theses hosts deliver almost full of their performance to the volunteer computing system.

5.2 Experiments

We realized 4 experiments with follow parameters:

Table 1. Experiment specification.

	Experiment 1.	Experiment 2.	Experiment 3.	Experiment 4.
Task count	15000	60000	15000	60000
Host count	1000	1000	1000	1000
Task sizes – distribution type	150 - 2000 normal	150 - 2000 normal	1000 - 2000 balanced	1000 - 2000 balanced
Host performance – distribution	1 - 50 Normal (1057, 308.3)	1 - 50 Normal (1057, 308.3)	1 - 50 normal	1 - 50 normal
Redundant computing factor	5	5	5	5

The experiments parameters described in see Table 1 differ by task count and host count and also by distribution type of task sizes. Other parameters include the probability of the bad result delivery and the probability that the host does not deliver the result in time. Both of them were set to 0.05. The performance of each host changes during the simulation, which also influenced the task scheduling. The changes

5.3 Results

We measured the total time of simulation and normalized it to the time of Boinc algorithm. The second column shows our algorithm with statistic median method used for creating groups. The third shows the class based algorithm with binary groups' method. The first line of each experiment shows the normalized value, the second one presents the absolute values and the third one the value of standard deviation. As the result shows, our algorithm reaches better results in almost every experiment. Experiments also points to that with the growing count of tasks, the benefit of the algorithm reduces.

Table 2. Experiment results.

	BOINC FIFO	CBA - MEDIAN GROUPS	CBA - BINARY GROUPS
Experiment 1.	*1.0*	*0.940*	***0.880***
Average	8377	7231	6769
St. dev.	1666	680	302
Experiment 2.	*1.0*	***0.998***	*1.000*
Average	33536	33408	33543
St. dev.	856	695	871
Experiment 3.	*1.0*	*0.810*	***0.870***
Average	7691	6523	6875
St. dev.	530	430	446
Experiment 4.	*1.0*	*0.982*	***0.979***
Average	33693	33018	33194
St. dev.	767	246	1091

The results of the experiments are described in Table 2. Each simulation case was tested fifteen times. The table shows the average values of those cases and the standard deviation.

6 Conclusion and further work

In this paper we described the main aspects and characteristics of Boinc scheduling. Based on these characteristics we present the new characteristic of hosts called assessment of host performance. We have also designed the new class based algorithm that counts with this characteristic. The results of the algorithm show, that it can be used for reducing total time of computation in volunteer computing systems in small and middle sized tasks. It

The main goal for future work is to test the algorithm on the real data acquired from real Boinc project and to cover up a new methods in the algorithm based on prediction of connecting some specific known hosts to the system.

Acknowledgement: This work was partially supported by the Science and Technology Assistance Agency under the contract No. APVV-0233-10.

References

[1] Trilce, E., Fuentes, O., Taufer, M.: A distributed evolutionary method to design scheduling policies for volunteer computing. *SIGMETRICS Perform. Eval. Rev.*, (2008), vol. 36, no. 3, pp. 40-49.

[2] Sonnek, J., Nathan, M., Chandra, A., Weissman, J.: Reputation-Based Scheduling on Unreliable Distributed Infrastructures. In: *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference*, (2006), p. 30.

[3] Kim, J.K., Shivle, S., Siegel, H.J. et al.: Dynamic mapping in a heterogeneous environment with tasks having priorities and multiple deadlines. In: *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, (2003), p. 15.

[4] Anderson, D.P.: BOINC: a system for public-resource computing and storage. In: *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, (2004), pp. 4-10.

[5] *BOINC*. [Online; accessed March 28, 2013]. Available at: http://boinc.berkeley.edu

[6] Dúbravská, M., Rosinová, D.: *Optimalisation*. Press STU, Bratislava, 1 edition, (2007). (in Slovak)

DNA Assembly: Reducing K-mers Number, Unique and Erroneous K-mers Detection

Peter KUBÁN*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
kuban@fiit.stuba.sk

Abstract. Next Generation Sequencing (NGS) methods for massively parallel DNA sequencing enable to perform genome sequencing of organisms faster and at much lower cost. These improvements have some disadvantages, e.g. sequenced fragments (reads) are shorter and therefore its number is higher. Many current assemblers require to generate k-mers (substrings of length k) over all reads. Data produced by NGS contain also errors what motivated researchers to develop a number of short-read error correctors and implement error corrections into assemblers. For achieving computational and memory efficiency unique and low covered k-mers/short-reads are often discarded if they cannot be repaired. In our paper we present a method based on k-mers that determines whether these unique k-mers are actually unique or incorrect.

1 Introduction

Next Generation Sequencing technologies are not able to sequence whole genome at once, hence a large amount of short fragments (reads) of genome is generated. Further development of NGS methods is connected with producing and processing large amount of sequencing data. Nowadays it is still challenging task to store, use and process all these raw data. To obtain the DNA data represents just one part of the problem. Another challenging problem is to analyze and assemble them into the whole genome or into collections of long contigs or scaffolds. Two main methods of assembling genomes currently exist: (i) mapping reads to a reference genome and (ii) *de novo* assembly.

Fragment assembly (Sanger sequencing) following the classical "overlap-layout-consensus" paradigm is used in many assembly tools. With application of new NGS methods using these tools faces difficulties and is not efficient any more [1]. Research on DNA assembly with short-reads is currently focused on de Bruijn graph methods [1, 2]. Demand to assemble genomes from NGS data has led to growth of novel assembly software. Several assemblers for short-reads in past few years such as ABySS [3], PASHA [4], ALLPATHS [5], Velvet [6], etc. have been developed. Among them there are few that are able to work in parallel computational environment, e.g. ABySS, PASHA.

[*] Doctoral degree study programme in field Software Engineering
Supervisor: Assoc. Professor Mária Lucká, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

NGS methods require sequencing every base in a sample several times for two reasons: (i) need for multiple observations per base to come to a reliable base call and (ii) reads are not distributed evenly over the entire genome, because reads sample the genome in a random and independent way. That is why many bases are covered by fewer reads while other bases are covered by more reads than average. This fact is expressed by the coverage metrics, that mirrors the number of times a genome has been sequenced (the depth of sequencing). High coverage does not ensure errorless reads, on the other hand it brings higher requirements for storage space, computation time and memory usage. In case of the de Brujin graphs is the amount of reads crucial. The nodes in a *de Bruijn* graph are the *k*-mers of a specified length *k* that are contained within the sequencing reads. Two *k*-mers are connected in the graph if they are adjacent in at least one sequencing read. Although *de Bruijn* graphs provide a nice conceptual framework that cuts down on computational time, the size of the graph can be very large, typically including billions of *k*-mers for vertebrate-sized genomes [7].

In this paper we describe a method and present experiment that aim to minimize the number of k-mers by correcting them and determining if they are unique (using statistical probability). We are focusing on single-base substitution errors which are most common. Other errors, for example include insertions or deletions of bases. Even the effect of substitution errors on the final sequence is mostly not very significant, in the large genomes - with high sequencing coverage - it makes a difference. Processing k-mers may seem to be simple, but with increasing number of k-mers it is computationally very demanding, so we focus only on small genomes to check up proposed method without focusing on its performance and efficiency.

1.1 Errors and quality scores

Sequencing errors are caused by limitations in DNA sequencing techniques. DNA sequencer manufacturers use a number of different methods to detect which DNA bases are present. Quality scores measure probability that a base is selected incorrectly. Each base in a read is assigned a quality score by an algorithm. Different manufacturers use different algorithms, mostly *phred*-like [12] algorithms. Phred quality score of a given base, Q, is defined by the equation

$$Q = -10\log_{10}(e)$$

here e is the estimated probability of the base call being wrong. Thus, the higher quality score indicates smaller error probability. With currently most used methods the error rate for bases goes from 0,001% to 2%.

2 Relevant work and existing tools

Over past few years a lot of research has been done in this field. For example, usage of Bloom filter for efficient counting of k-mers is presented in [7]. Software, called Jellyfish [8], is designed for k-mer counting. It has much lower memory requirements than other available methods. Number of methods for efficient storing of k-mers were proposed, usually using hash tables. For example, the assembly software ABySS [3] uses the Google sparsehash library, which has minimal memory overhead. Some error detectors and correctors were created, e.g. Quake [9].

In 2009 the team that sequenced the giant Panda genome counted a total of 8.62 billion 27-mers. After removing or correcting low-coverage *k*-mers, they eliminated 68% of the observed *k*-mers, reducing the total number to just 2.69 billion. Their genome assembly was based on this reduced set [10, 7].

3 Idea and Experiments

3.1 Basic idea

For our experiments we have picked some already assembled small genomes from GenBank (https://www.ncbi.nlm.nih.gov/genbank) and used them for comparison of the amount and coverage of k-mers over the whole genome *before* and *after* running our algorithm. With the software MetaSim [11] we executed some simulations to create sequencing reads. After creating k-mers (from these reads) we ran our algorithm to find erroneous k-mers and to identify unique k-mers. Moreover, we have created a simple visualization tool for comparing achieved results.

3.2 Algorithms

We used C++ language with standard libraries without focusing on performance and efficiency.

Firstly, we have created algorithm for generating k-mers from reads, since mentioned software Jellyfish did not provide identifiers of reads from which the k-mer came. For storing produced reads we have used *Read* object that contained full nucleotides (bases) sequence (read itself) and read identifier. All reads (Read objects) were stored in an *unordered_map*. Reads with smaller length than k-mer size (k) were discarded. *Kmer* object stored information concerning k-mer count from all reads, identifiers of reads and nucleotides sequence (k-mer itself). All k-mers (Kmer objects) were stored in an *unordered_map*.

After all k-mers were created and counted, we have iterated through all of them and for each k-mer which count was equal or less than certain threshold we tried to find other *similar* k-mer with higher count. K-mers were considered similar when their Hamming distance was lower or equal as defined threshold (in our experiments with single-base substitution errors we have used the threshold equal to 1). If no similar k-mer with higher count was found, we have continued with the next k-mer otherwise we have assumed that an error in the k-mer with lower count was found. In the case that an error was found, we have compared prefixes and suffixes of reads from which similar k-mers were created to determine if they are really from same place in genome. If that was the case, the count of correct k-mer was increased and all k-mers containing incorrect base (found in that read) were removed from *unordered_map*. After removing these k-mers we have assumed that the rest of k-mers (from the same read) were *unique*, because the probability of another error in the same read was very small (in the *Figure* 1).

Finally, a fasta file containing resulting k-mers and their counts was created and the achieved results were visualized.

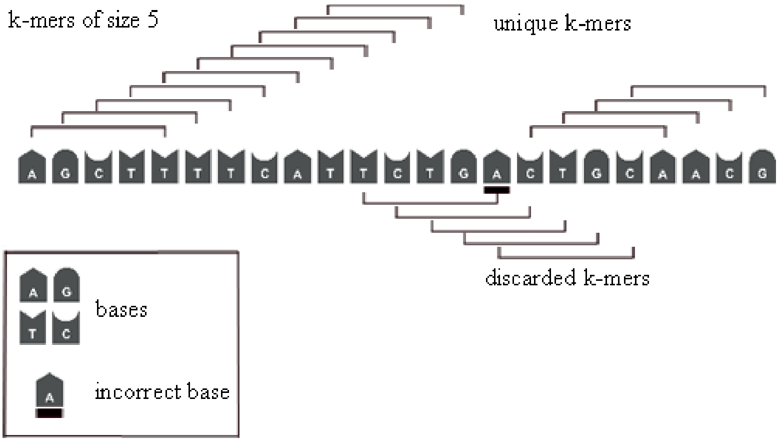


Figure 1. Unique and discarded k-mers by our algorithm (for k = 5).

3.3 Experiments

Genome selected for our experiments stemmed from bacteria organism Candidatus Carsonella ruddii (AF268064.1) downloaded from GenBank with the length of whole genome 2712 bp (base pairs). Sequencing reads were generated using MetaSim [11]. After generating reads we ran our experiments for k-mers of sizes 31, 27 and 23.

3.3.1 Dataset

Organism: Candidatus Carsonella ruddii(GenBank: AF268064.1)
Genome length: 2712 bp
MetaSim settings:

- Error Model: Sanger
- Number of Reads or Mate Pairs: 200
- Mean: 50
- Second Parameter: 10
- Insertion and Deletion Error Rate: 0
- all other settings as default

3.4 Results

For visual comparison we have created simple tool (using J*avascript* and HTML <*canvas*> element) runnable in web browser. Visualization shows whether the base is covered by k-mers and how many times. The darker vertical line means that base is more covered (more times contained in k-mers). For each experiment there are 4 visualizations: (*a*) shows coverage of genome if all short-reads and therefore k-mers were correct, (*b*) coverage without correction, (*c*) coverage after using our algorithm, (*d*) coverage of unique k-mers found by our algorithm.

Table 1. Counts of k-mers for k-mers of size 31.

k = 31	Different k-mers	Correct k-mers	Incorrect k-mers	Covered bases
a)	2021	2021	0	2588
b)	3805	1774	2031	2588
c)	2738	1735	1003	2588
d)	293	269	24	994

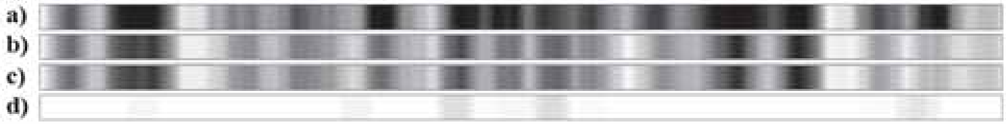


Figure 2. Visualization for k-mers of size 31.

Table 2. Counts of k-mers for k-mers of size 27.

k = 27	Different k-mers	Correct k-mers	Incorrect k-mers	Covered bases
a)	2201	2201	0	2599
b)	4307	2027	2280	2599
c)	2847	1980	867	2599
d)	498	450	48	1215

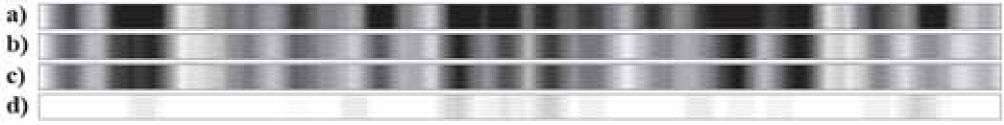


Figure 3. Visualization for k-mers of size 27.

Table 3. Counts of k-mers for k-mers of size 23.

k = 23	Different k-mers	Correct k-mers	Incorrect k-mers	Covered bases
a)	2326	2326	0	2599
b)	4634	2228	2406	2599
c)	2862	2190	672	2599
d)	829	682	147	1519

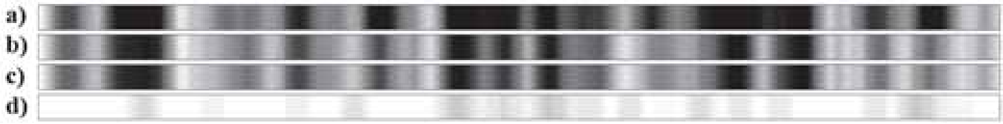


Figure 4. Visualization for k-mers of size 23.

4 Conclusions and future work

We proposed a method for reducing number of k-mers required in genome assembly. It is done by detecting *single-base substitution* errors and discarding k-mers containing these errors. K-mers from reads with errors which did not contain a defective base were considered as *unique*, because the chance of another error in same read is with current technologies very unlikely (in the *Figure 1*).

At first, we have compared the number of k-mers (correct and incorrect) and the coverage of bases over the whole genome from raw - not repaired - k-mers (row **b** in the *Tables 1-3*) with results after running our algorithm (row **c** in the *Tables 1-3*). The results showed that we were successful in reducing the number of incorrect k-mers, while number of covered bases stayed the same. On the other hand we have also lost a small number of correct k-mers. Even our algorithm had no influence on the achieved results on experimental data, where no base losses were observed, it need not to be the case for larger datasets.

We have also checked the numbers of *unique* k-mers. According to our experiments the number of correct k-mers was significantly higher than the number of incorrect ones, but their coverage over the whole genome was much smaller as we hoped for.

Presented results show the importance of a correct choice for k-mer size. Successful application of our method suppose that the k-mer size is not much higher than half of the mean of reads length.

In the future we will focus on development of methods applicable to larger datasets and also on other kinds of errors that could be usable on real DNA datasets.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0752/14.

References

[1] Pevzner P, Tang H, Waterman M: An Eulerian path approach to DNA fragment assembly. In: *Proceedings of the National Academy of Sciences of the United States of America*, (2001), pp. 9748-9753.

[2] Zhenyu L., Yanxiang C., et al.: Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-bruijn-graph. In: *Briefings in Functional Genomics*, (2012).

[3] Simpson J., Wong K., Jackman S., Schein J., Jones S., Birol I.: ABySS: a parallel assembler for short read sequence data. In: *Genome Res 2009*, (2009) pp.1117-1123.

[4] Liu Y., et al.: Parallelized short read assembly of large genomes using de Bruijn graphs. In: *Bioinformatics 2011*, (2011).

[5] Butler J., MacCallum I., Kleber M., Shlyakhter I., Belmonte M., Lander E., Nusbaum C., Jaffe D.: ALLPATHS: de novo assembly of whole-genome shotgun microreads. In: *Genome Res 2008,* (2008), pp. 810-820.

[6] Zerbino D., Birney E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. In: *Genome Res 2008*, (2008), pp. 821-829.

[7] Melsted P., Pritchard J.K.: Efficient counting of k-mers in DNA sequences using a bloom filter. In: *BMC Bioinformatics 2011*, (2011).

[8] Marcais G., Kingsford C.: A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. In: *Bioinformatics,* (2011), pp. 764-770.

[9] Kelley D.R., Schatz M.C., Salzberg S.L.: Quake: quality-aware detection and correction of sequencing errors. In: Genome Biology 2010, (2010).

[10] Li R., Fan W., Tian G., et al.: The sequence and de novo assembly of the giant panda genome. In: *Nature 2010*, (2010), pp. 311-317.

[11] Richter D., Ott F., Auch A., Schmid R., Huson D.: MetaSim—A Sequencing Simulator for Genomics and Metagenomics. In: *PLoS ONE*, (2008).

[12] Ewing B., Hillier L., Wendl M., Green P.: Base-calling of automated sequencer traces using phred. I. Accuracy assessment. In: *Genome Res. 8,* (1998), pp. 175-185.

Computer Graphics, Multimedia and Computer Vision

Transcription of Piano Music

Rudolf BRISUDA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xbrisuda@is.stuba.sk`

Abstract. Music transcription can be solved in several ways. We present the-state-of-the-art in automatic polyphonic transcription and solution of automatic pages turning for piano music. We analyze problems of music transcription which could be used for this purpose. We focus on keystroke detection (Note Onset Detection based on Spectral flux) and detection of tones (simple and computationally efficient method to polyphonic pitch detection based on Summing Harmonic Amplitudes) in this keystroke. Whereas detection of keystroke often fails to track position in song, we propose an algorithm which corrects position within the song by polyphonic pitch detection. Proposed algorithm repairs Spectral flux with Polyphonic pitch detection algorithm and it outperforms the Spectral flux itself.

1 Introduction

Pianists have often problem with turning pages while playing songs. Therefore, they often missed a part of the song because they use the hand to turn the page. They have to learn the songs by heart if they want to play flawlessly or not use "ninja moves" to turn the pages. Many musicians use to store and display music sheets by the tablets which provide new possibilities. For example, algorithm of music transcription should be able to determine where the pianist in the song is and the algorithm could automatically assess when to turn the page.

There are hardware solutions based on foot pedals. One problem still remains, musician still need to pay an attention to additional device. We focus on automatic turning of the pages by using a microphone. One of the advantages of the use of this algorithm could be a higher portability and no need of additional devices.

Our aim is to develop a solution which will analyze the sound captured by a microphone in real-time. We focus on certain types of algorithms belonging to music transcription and we try to solve this problem in the simplest way. For this purpose, we decide to use the algorithm to onset (mainly) and Polyphonic Pitch Detection (PPD). First, piano keystrokes will be detected with some accuracy and it will be corrected with detected notes. Keystrokes will be tracked in the played song by comparing sound data with input data of music sheets and when they reach end of the page, the page will be turned. Both of the algorithms operate with some accuracy and tracking song only by one approach provides poor results.

[*] Bachelor study programme in field: Informatics
Supervisor: Andrej Fogelton, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 331–336.

2 State of the art

Pitch detection algorithms are designed to detect pitch or fundamental frequencies from sound signals (e.g. music or speech). These algorithms have been developed primarily with the interest in speech recognition. There are many complex methods which reflect this nontrivial problem [4,8,10,11,14]. The algorithms can be divided into the following categories: time domain method, frequency domain method, combination of time and frequency methods and models of human ears.

Time domain methods (TDM) operate directly with the input signal as a fluctuating amplitude. They look on the waveform with the aim to find repeating patterns which indicate periodicity. The principle of the frequency domain method involves dividing of the input signal into the frequencies. These frequencies represent the spectrum which shows their strength. The typical analysis include Short Time Fourier Transform (STFT) [13]: division of signal into segments, applying window and subsequently on each segment performing Fourier Transform. This shows peaks which may correspond to pitches (fundamentals frequencies), harmonics (integer multiples of the fundamental frequencies or redundant parts. The aim is to find the pitch out of a spectrum. Unfortunately the strongest component may not be the fundamental one [13].

The time domain and frequency domain methods by themselves are only suitable for very small set of piano songs. This song may contain only monophonic sound (one pitch at a time). The methods are not suitable to chord detection (multiple simultaneous pitches, polyphonic). Problems in time domain approach occur at signals which are not only periodic e.g. signals with noise or polyphonic signals (containing multiple fundamental frequencies simultaneously). Also, the frequency domain approach by itself has a problem with polyphonic detection, but it is possible. Attempts to polyphonic pitch detection are mainly applied in the frequency domain approach [13]. The basic principle include frequency spectrum, which results to amplitudes of peaks. This approach has to be reinforced by several other decision-making and search mechanisms. Many algorithms of these methods perform detection on clean monophonic signal well but failed at noisy signals or polyphonic signals.

Pitch detection is complex problem for monophonic sound, where pitch detection algorithms estimate one pitch at a time. However, there is a need to polyphonic pitch detectors, which can extract multiple pitch at a time or pitches in presence of the noise. This problem is referred to as music transcription or music information retrieval (converting a low-level representation of music into a higher-level representation – MIDI or even music sheets). There are several researches which analyze this problem [1, 6, 9, 12]. Whereas the musical note does not include only the pitch but duration, loudness and timbre [2]. However, detection multiple concurrent pitches [5, 7, 16] is the core of the problem [1]. Further substantial problem is a real-time processing. One way to increase efficiency is to use an iterative principle (e.g. [7]).

Analyses of state of the art in this area with connection with the real-time processing, we found that page turning could be only addressed with the one part of music transcription – note onset detection. This detection based on the control of input data (keystrokes in music sheets) can determine at what position in the song we currently are.

3 Transcription

Music transcription is process of converting musical record into music sheets. This task implies to estimate the pitch, tempo, note onsets, timing of notes, loudness, etc. The task is even more difficult if you are dealing with polyphonic music. If keys on the piano are simultaneously pressed then amplitude in time domain significantly rises. For this reason we focus on the specific problem of music transcription (onset detection) which allows to isolate this change. After our evaluation, we found that accuracy at different input data is not sufficient. Therefore we decide to use control algorithm (PPD) with tracking of played song which used only this method. Both the algorithms operate with some accuracy. After research of the available and implemented methods, we found two methods which are appropriate in terms of efficiency and portability to android device. We analyze two selected and used problems of Music Transcription (spectral flux to onset/keystrokes detection and summing harmonic amplitudes to PPD). Our method is also appropriate to real-time tracking of song.

3.1 Spectral flux

Spectral flux measures the change in magnitude in each frequency bin [3]. Equation 1 presents summing the positive differences between actual S and the previous frequency LS across all frames, where L is length of spectrum frame.

$$f(t) = \sum_{i=0}^{L} S(i) - LS(i) \tag{1}$$

Keystrokes are determined by peak picking algorithm over $f(t)$. Pre-processing by appropriate threshold function is needed.

3.2 Summing harmonic amplitudes

In [7], there is proposed conceptually simple and computationally efficient fundamental frequency estimator. The estimation is based on summing harmonic amplitudes. It operates in the following steps:

- calculate spectral whitened signal of input signal,
- calculate strength (salience) (Equation 2) of fundamental frequencies candidates as weighted sum of the harmonic amplitudes where, $g(\tau, m)$ is learned by brute force optimization and $f_{t,m}$ is frequency of fundamental frequency candidate.

Spectral whitening suppresses timbre information before actual estimation. Reason of this processing is to make system robust for different input sound sources [7]. It performs by flattening rough spectral energy by inverse filtering [15]. This is done in frequency domain.

$$s(t) = \sum_{m=1}^{M} g(\tau, m)|Y(f_{t,m})| \tag{2}$$

3.3 Estimation of tracking within the song

The problem of tracking within the song only with onset detection is principally with songs characterized by the presence of noise, high tempo, volume level and duration of each individual note. This can result to spurious or omitted keystrokes. There is a need for another control algorithm. We decide to use PPD, which can give clues about type of playing notes. Whereas the problem is the same for both of them, we create solutions which estimate song tracking on the basis of keystrokes with the support of detected notes.

The output of PPD consists of one or simultaneously played notes for each time frame. Length of the frame depends on Fast Fourier Transform window size. Therefore, there are regularly received estimated notes without any information about duration of played notes. Detected peaks from onset detection, thus can give clue about the duration of the notes and also range for note searching. However, peak is not the place where note goes from zero to duration, we add notes' data between the two peaks of some length in addition to currently examined notes. Whereas tracking has to be robust for all durations of song, we empirically found that better results gives the length of $TBTP/3$, where TBTP is the Time Between Two Peaks. So we define note duration time as $TBTP+TBTP/3$.

The algorithm works primarily with onset detection, so we establish decision rules where the detected keystrokes have the largest priority if another check failed. First of all, the algorithm checks

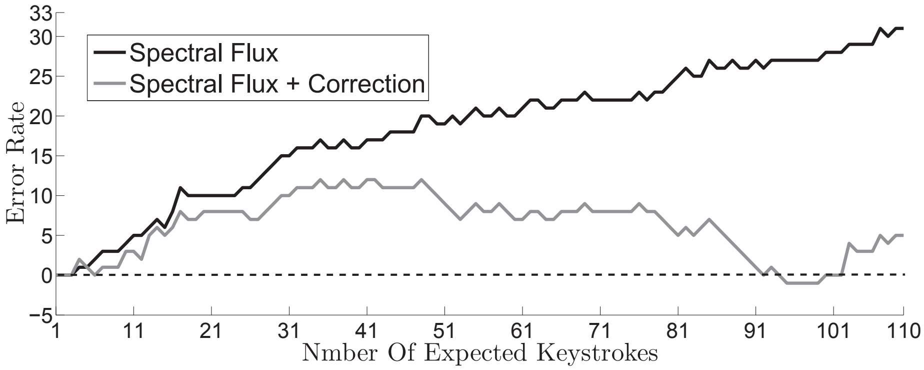


Figure 1. Tracking within the song by spectral flux and correction. Song tempo: 112.

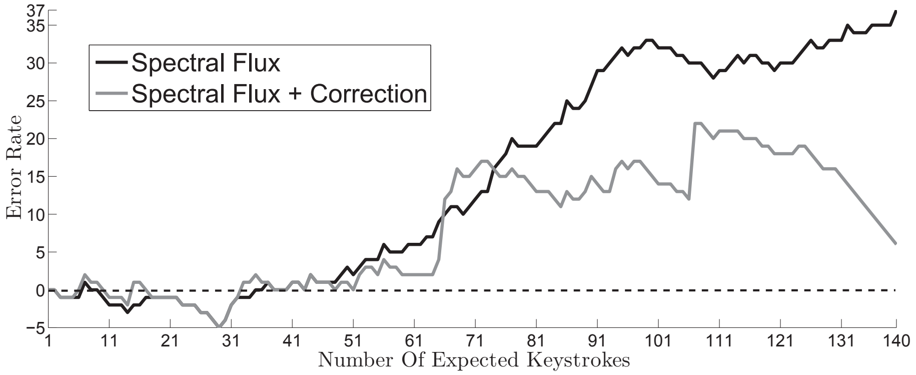


Figure 2. Tracking within the song by spectral flux and correction. Song tempo: 120.

if notes in TBTB are equal with some notes from keystrokes of the input music sheets. If yes, the algorithm considers the keystroke as correct and waits for next keystrokes. If this test failed, we assume a problem with spurious or omitted keystrokes.

We try to eliminate the spurious keystrokes by searching previous keystrokes within the input music sheets. The reason is that if there is a short note duration time, we assumed that there can be an occurrence of previous note, because the note could sounds longer. We try to locate the omitted keystrokes in note duration time by search of keystroke sequence of input music sheets. How many notes in sequence are found, so much are added to the total keystrokes. We also create probabilistic model of comparing the detected notes with input because PPD works in some accuracy. It works on the principle of comparing notes with a note range ($+-0$, $+-1$, $+-2$). We empirically found, if there are results from $+-0$ or $+-1$ in the same test, better results are reported with value which represents their average. This average include number of founded notes. We consider that both results in this range are caused by the inaccuracy of the PPD. We also assume error range $+-2$ in the case of failure of the first two tests of the range. Other results are evaluated on the basis of the results in the presented ranges in sequence.

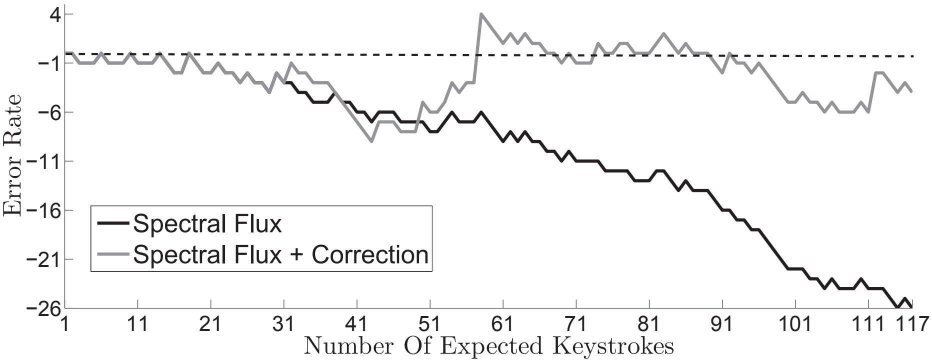


Figure 3. Tracking within the song by spectral flux and correction. Song tempo: 200.

Table 1. Accuracy of the both methods by keystrokes. Spectral flux: TP - correct identified, FP - spurious. Our correction: TP - correct added, FP - false added, TN - correct removed, FN - false removed.

songs		spectral flux		our correction			
tempo	keystrokes	TP	FP	TP	FP	TN	FN
112	110	96.36%	34	0	25	16	20
120	140	77.14%	68	9	30	29	29
200	117	58.11%	22	19	32	1	3

4 Test and evaluation

Two types of input data (song with corresponding music sheets in the form of MusicXML) are used to test our algorithm. We construct MusicXML parser which eliminates keystrokes with relevant notes from music sheets which gives clue of the tracking within the song. We manually annotated the first pages of three songs at each keystroke. Evaluation of the tracking within the song is measured by shift (error rate) against the expected number of keystrokes.

Figures 1, 2 and 3 shows comparison between our algorithm with the method based on the spectral flux only in spurious and omitted keystrokes. Since the spectral flux itself cannot control tracking, each shift has an impact on the final result.

Accuracy of the both method is shown in Table 1. Measurement of our correction include correct added of unidentified keystrokes by spectral flux, false added, correct removed of spurious keystrokes by spectral flux and false removed keystrokes. Our algorithm shows that the wrong identification of spurious and omitted keystrokes brings better results.

5 Discussion and conclusion

We have analyzed algorithms of music transcription and propose algorithm to tracking within the song based on these algorithms. There are additional algorithms related to music transcription which could deal with this problem of tracking within the song, so it is not necessary to perform all the process of music transcription.

Accuracy is influenced by output of both algorithms which still remains to problem of music transcription (robust algorithms which could deal with different types of songs). Tests claim that synthesis of both algorithms in the despite of their varying accuracy reaches better results. This results are affected by the false detection of spurious and committed keystrokes. In the final analysis, the algorithm provides better results compared to spectral flux itself, what is demonstrated by the tests at three different songs.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., Klapuri, A.: Automatic Music Transcription: Breaking the Glass Ceiling. In: *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 379–384.

[2] BYRD, D.B.: Problems of Music Information Retrieval in the Real World. *Computer Science Department Faculty Publication Series*, 2002, p. 4.

[3] Dixon, S.: Onset detection revisited. In: *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006, pp. 133–137.

[4] Gold, B.: Computer Program for Pitch Extraction. *J. Acoust. Soc. Amer.*, 1962, vol. 34, pp. 916–921.

[5] Klapuri, A.P.: Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *Speech and Audio Processing, IEEE Transactions on*, 2003, vol. 11, no. 6, pp. 804–816.

[6] Klapuri, A.: Signal Processing Methods for the Automatic Transcription of Music. Technical report, Tampere University of Technology, 2004.

[7] Klapuri, A.: Multiple fundamental frequency estimation by summing harmonic amplitudes. In: *in ISMIR*, 2006, pp. 216–221.

[8] Noll, A.M.: Cepstrum Pitch Determination. *J. Acoust. Soc. Amer.*, 1967, vol. 41, pp. 293–309.

[9] Paiva, R.P., Mendes, T., Cardoso, A.: Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience, and Melodic Smoothness. *Comput. Music J.*, 2006, vol. 30, no. 4, pp. 80–98.

[10] Phillips, M.S.: A Feature-Based Time-Domain Pitch Tracker. *J. Acoust. Soc. Amer.*, 1985, vol. 77, pp. S9–S10.

[11] Rabiner, L.R., Cheng, M.J., AaronE.Rosenberg, McGonegal, C.A.: A Comparative Performance Study of Several Pitch Detection Algorithms. *IEEE Trans. on ASSP*, 1976, vol. 24, no. 5, pp. 399–418.

[12] Reis, G., de Vega, F.F., Ferreira, A.: Automatic Transcription of Polyphonic Piano Music Using Genetic Algorithms, Adaptive Spectral Envelope Modeling, and Dynamic Noise Level Estimation. *IEEE Transactions on Audio, Speech and Language Processing*, 2012, vol. 20, no. 8, pp. 2313–2328.

[13] Roads, C.: *The Computer Music Tutorial.* MIT Press, Cambridge, MA, USA, 1996.

[14] Schafer, R.W., Rabiner, L.R.: System for Automatic Formant Analysis of Voiced Speech. *Journal of the Acoustical Society of America*, 1970, vol. 47, pp. 634–648.

[15] Tolonen, T., Member, S., Karjalainen, M.: A computationally efficient multipitch analysis model. In: *inria-00350163, version 1 - 6*, 2000, pp. 708–716.

[16] Yeh, C., Roebel, A., Rodet, X.: Multiple Fundamental Frequency Estimation and Polyphony Inference of Polyphonic Music Signals. *Trans. Audio, Speech and Lang. Proc.*, 2010, vol. 18, no. 6, pp. 1116–1126.

Low-Cost Acquisition of 3D Interior Models for Online Browsing

Filip MIKLE, Matej MINÁRIK, Juraj SLAVÍČEK, Martin TAMAJKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
miklefilip@gmail.com, matejminarik.fiit@gmail.com,
juraj.slavicek@gmail.com, martin.tamajka@gmail.com

Abstract. Kinect as a depth camera has brought new opportunities to gaming industry. Recently, developers have found a few other ways to use Kinect in different kinds of applications. We use Kinect as a scanning device to create 3D virtual models of interiors. We demonstrate our solution on real estate market. Our models are means of saving time during real estate selection. Our solution supports point cloud alignment process with accelerometer and other orientation sensors. In order to decrease memory requirements for scanned models, we find flat surfaces and reduce number of polygons representing them. Created models are presented to users through a web application.

1 Introduction

In last few years scanning of 3D objects experienced a rapid expansion. This expansion was caused mainly by introduction of cheap RGB-D camera Kinect at the end of the year 2010. Besides its use in the field of entertainment, its usability in other fields emerged, including scanning of interiors.

We use Kinect as a scanning device for creating realistic 3D models of real estates. This process consists of capturing individual point clouds of interior, their *registration* (process of transforming different sets of data into one coordinate system), transformation to surface representation in form of triangle mesh, compressing size of final mesh due to high memory requirements and finally presenting final result in web application. The challenging part of this solution tracking position of Kinect in the interior. We decided not to use standard approach using artifical markers. Instead, we use combination of IMU (Inertial Measurments Unit) and natural key features from interior. IMU is also used to correct and smooth data captured from Kinect.

Our main goal is to create new, time saving way of selling, buying and renting real estates. We believe that provided models, thanks to their realistic character, will reduce number of in-person visits of real estate during the selection process. This will be achieved by presenting real estates in immersive way, supplemented realistic sounds (steps) and adjustable height of eyes.

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Jakub Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

2 Related work

When Kinect was released, researchers and developers started discovering another purpose for this sensor, besides interactive games. Scanning of interiors in order to create 3D virtual models has been presented in several solutions.

Kinect Fusion is available solution, created by Microsoft. It fuses all of the data from Kinect sensor into a single global model in real-time. The current sensor pose is simultaneously obtained by tracking the live depth frame. Their solution works well for mapping medium sized interiors with volumes of less than 7 cubic meters. Scanning has to be made with patience and drawbacks, when the algorithm cannot estimate current position relative to global model [1].

Kintinuous provides an option to scan extended scale environments in real-time. That is achieved by altering original algorithm such that model being mapped can vary dynamically and by incrementally adding resulting points to a triangular mesh representation of the environment. This solution is an extension to Kinect Fusion [2].

ReconstructME is an application, developed by individual. We downloaded this application, which had serious compatibility issues. Created models had low detail, scanning was slow and application provided very few settings related to scanning and obtained models.

Skanect is a professional solution implemented by ManCTL. Skanect is easy-to-use solution, which can capture dense 3D information at 30 frames per second. Models created by Skanect were more accurate than previous ones, but, similarly to Kinect Fusion, dimensions of model had to be known and set before scanning.

3 General overview

Our main goal was to create a technology, which (using Kinect) allows simple and automatic creation of accurate 3D models of interiors. As far as this area is concerned, many standardized techniques were developed till now, mostly taken from the field of robotics and computer vision. The most important of these are algorithms used within image *registration* process as ICP and RANSAC together with associated keypoint-getting algorithms, such as AGAST.

Due to limits of Kinect causing inaccuracy in depth frames obtained from Kinect, which would make image *registration* more difficult, we decided to enrich these traditional techniques by supplementing additional information received from IMU (Inertial measurement unit), adjusting inaccuracy caused by Kinect. Apart from this, this additional information increases probability that non-deterministic image *registration* algorithms (primarily RANSAC) succeed.

Because *registration* is performed in real-time, it was indispensable to take care of computational effectiveness of point cloud *registration*. In order to reach this goal, we used parallelism and General-purpose computing on graphics on graphics processing unit.

During the process of *registration*, a lot of points provide redundant spatial and color information. This redundancy is not possible to be avoided, and it increases spatial complexity, too. This makes raw, non-optimized models consisting of individual points non-suitable for presentation (some models created by Kintinuous reached size of terabytes [2]). To remove this restriction, we decided to transform point clouds into surface representation and optimize it.

In overall process, from interior scanning through model optimization to model presentation, we identified following:

1. It's necessary to always *track camera position*, relative to some reference point. As reference point (point with coordinates [0.0, 0.0, 0.0]) we decided to take position of Kinect when capturing first frame.
2. It's necessary to *register frames taken by depth and RGB camera into model*. To reach this goal, we have to compare neighbor (near, respectively) frames and find similarities.
3. Because data obtained from Kinect are noisy, we need to use additional information from accelerometer, gyroscope and magnetometer

4. It's necessary to *transform point cloud* got as the result of point 2 to *surface representation*, ideally to one of standard format (.ply, .obj), because there's already plenty of standard tools, that are able to work with them (e.g. Unity Web Player).
5. It's necessary to *minimize surface representation* got as the result of point 4, preserving as much details of it as possible.

Respecting these requirements, and trying to modularize our solution, so it allows to work on it in parallel, we identified four modules (*Figure 1*). The main idea of this modularization is that each module depends on at most one other module a vice versa.

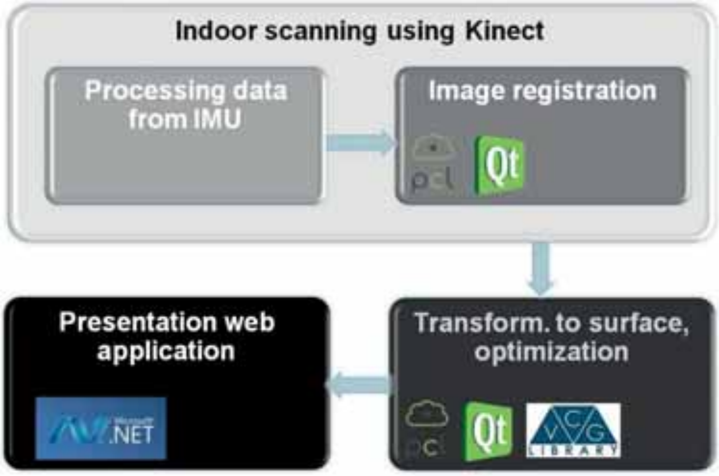

Indoor scanning using Kinect
Processing data from IMU
Image registration
pcl
Qt
Presentation web application
.NET
Transform. to surface, optimization
pcl
Qt
VCG LIBRARY

Figure 1. Modules and their dependencies.

3.1 Interior scanning and model creation

To create dense and accurate models of interiors, we need to obtain two sets of data from Kinect, namely frames from its depth and RGB camera. These work as input to our algorithms.

First of all, we need to merge color and depth frame, both of them having resolution 640 x 480 px. It's not possible to merge these frames directly, only pairing element on position X,Y in color frame with element on position X,Y in depth frame, because color and depth sensors do not have center in the same point, there's small (but important) shift between them. We decided to use in-built functionality provided by Kinect-SDK to merge these frames. The output of this step is point cloud consisting of points with spatial and also color information.

Second, and the most important step, is to register image into existing model. Simply, our model consists of large number of individual point clouds got in previous step. Because X, Y and Z coordinates of points in point cloud are calculated relatively to Kinect, and not absolutely to the rest of the world, it's needed to transform these relative coordinates into real world coordinates:

1. Relative transformation of most-current point cloud to the previous one
2. Absolute position of Kinect in space, and it's view vector

To calculate relative transformation of most-current point cloud to the previous one means aligning them into the same coordinate system. In other words, we need to find rotation and translation, that if applied to previous frame, this is aligned into same coordinate system as the last one. This is non-trivial problem, made even more difficult by noisy points contained in point clouds (points without known depth, or with non-accurate one). Looking for transformation does not guarantee to provide absolutely accurate result. We decided to use ICP algorithm [3], which in cooperation with RANSAC iteratively tries to converge to sufficiently good result. This algorithm estimates some affine transformation using SVD (Singular value decomposition - method used to find rigid transformation between point sets), applies it to the source (previous, not most current) point cloud, and computes root squared distance, given by formula

$$E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left\| m_i - \left(Rd_j + t\right) \right\|^2 \tag{1}$$

ICP algorithm ends when this distance no longer decreases, or when maximal number of iterations is reached. Otherwise, it tries to find other affine transformation using SVD and iterates.

Because ICP algorithm is quite complex (mainly when using RANSAC), it is not possible to use all points obtained from Kinect. It's needed to select some subset of points, called keypoints, that represent individual point clouds. Algorithm used to find these points should be rotation, translation, scale and lightness change resistant, which should guarantee, that the same points will be identified as keypoints after Kinect moves. Because algorithms such these are often computational expensive, we decided to use parallel approach.

After this transformation is found, we need to recalculate position of camera, using estimated transformation got as the result of previous step.

3.2 Scanning support

One of the inputs to iterative closest point (ICP) algorithm is initial estimation of device`s transformation, expressed by roll, pitch and yaw angles.

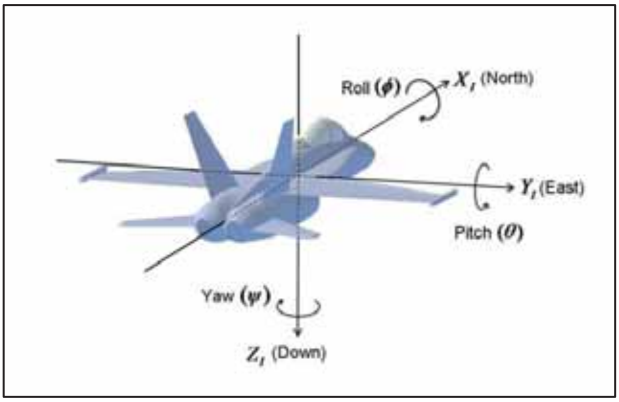


Figure 2. Inertial frame of reference [4].

This initial transformation can be obtained from visual information or by assuming constant transformation. There are cases, when we cannot estimate transformation from visual information, because we cannot obtain adequate number of key points (white walls, homogeneous floors, etc.). In order to successfully estimate camera transformation in these cases, we need another source of information.

Here comes in handy earth`s gravity and magnetic field vector, with strength and direction. In other words, we need accelerometer, magnetometer or magnetic compass and gyroscope. These sensors together create *inertial measurement unit (IMU)* and to fuse all data together, *sensor fusion* approach is important. We can obtain gravity vector from accelerometer and magnetic north reference from magnetometer, but both sensors suffer from noise, mostly caused by fabrication issues and measurement errors. Gyroscope measure angular velocity in all three directions (pitch, roll, yaw). Gyroscope data are less noisy, but suffer from drift.

In order to obtain smooth transformation measurements, we need to fuse gyroscope data with fixed reference by accelerometer and magnetometer. There are several ways to fuse obtained measurements. One of them uses Kalman filter [5]. *Kalman filter* is powerful, but mathematical model is not intuitive and deep understanding is needed. Another way to fuse these data is *complementary filter*, which is more intuitive and gives reasonably accurate results.

$$\text{ComplementaryYaw} = \text{FC}^{1} * \text{GyroscopeYaw} + (1 - \text{FC}) * \text{MagnetometerHeading} \tag{2}$$

$$\text{ComplementaryRoll} = \text{FC} * \text{GyroscopeRoll} + (1 - \text{FC}) * \text{AccelerometerRoll} \tag{3}$$

$$\text{ComplementaryPitch} = \text{FC} * \text{GyroscopePitch} + (1 - \text{FC}) * \text{AccelerometerPitch} \tag{4}$$

Obtained results are expressed by Euler angles, which are easy to understand, but they suffer from “Gimbal lock”. This does not pose a problem to our solution, because we don’t expect such position of Kinect while scanning. To overcome this issue, we could use quaternions. The result is a 4-dimensional vector, which can be subsequently computed into pitch, roll and yaw representation.

3.3 Model optimization

Point cloud representation of 3D model got as the result of previous steps is not suitable for manipulation. In order to manipulate with it, we transform point clouds to triangulate mesh which is a structure of connected triangles forming solid surface. First, we create bounding box to our point cloud which is wrapping object to all points of point cloud. Its shape is determined by shape of point cloud. Next, created bounding box is divided into uniform (each voxel has same size) cube voxel grid. Each point in point cloud is then assigned to voxel according to its coordinates. For each face of cube of each voxel is computed its normal. On the basis of comparing face normal to normal of camera position, which is practically viewpoint of camera, we can determine whether face can be seen from camera position or not. After confirmation that face is seen is divided into two triangles that are parts of final mesh model.

Because point cloud of common apartment may contain tens or hundreds millions of points, derived mesh has high memory demands. In order to prevent problems in presenting phase, model needs to be simplified. This is achieved by Quadric Edge Collapsing algorithm that removes redundant edges and collapses affected vertices. The challenge is to determine right simplification ratio in order to keep balance between lower memory demands and quality of the model.

3.4 Presentation to end user

Users who want to browse through our scanned 3D models are able to do so via our web application created mainly in ASP.NET. The model browsing itself is created with WebGL and Three.js, both of which are javascript libraries designed for the purpose of allowing great-looking graphics to be displayed in most web browsers.

Even though the created 3D models will be smaller in size thanks to our reduction method, they are still too huge to be loaded by the browser at one time. Therefore we have to cut the models into smaller parts. The user will have to download only the first part and while viewing it, the other parts will download as well. This way we reduce the wait time for the end user.

The main goal of the web application is to allow users to view our 3D models but also provides other functionality that helps users to feel like they are really located in the selected real estate. They can freely move through the environment with the use of mouse and keyboard and look behind every corner. We give them the option to adjust their height to match theirs in real life and when they move through the environment they can hear footstep which makes their experience more immersive.

[1] There are several ways, to obtain fusion constant (FC), but it is suggested to estimate it within interval <0.9; 1>. This constant can be also estimated empirically.

4 Conclusions and future work

To sum up, the intent behind project Real Deal is to find a fast method how to create high quality 3D models with relatively small size for real estate agencies and thus replacing the outdated forms of presenting estates - photos and panoramas. We achieved this by using the sensor Microsoft Kinect alongside with a group of other sensors including accelerometer, magnetometer and gyroscope and using their output to help our matching algorithm to be faster and more accurate.

The entire project consists of three independent parts. The first part includes the matching algorithm that creates a 3D model from the data gathered by Kinect and our other sensors. The next part is another algorithm that transforms our created 3D models from point cloud representation into meshes and shrinks to a smaller size so they can be later viewed in a web application which represents the last part of our project.

In the future, outside of fixing and improving our creation and minimization process of 3D models we mainly aim to add functionality to our web application. This involves the option of adding and removing parts of the 3D model such as various furniture or walls and replacing them with different ones. We also plan on allowing the user to change the color and style of each object in the scene. As far as performance is concerned, our goal is to move parallelizable computations to GPU.

References

[1] Newcombe, Richard A. and Izadi, Shahram and Hilliges, Otmar and Molyneaux, David and Kim, David and Davison, Andrew J. and Kohli, Pushmeet and Shotton, Jamie and Hodges, Steve and Fitzgibbon, Andrew. KinectFusion: Real-time Dense Surface Mapping and Tracking. *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*. Washington, DC, USA : IEEE Computer Society, 2011, pp. 127-136.

[2] T. Whelan and M. Kaess and M.F. Fallon. Kintinuous: Spatially Extended KinectFusion. *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*. Sydney, Australia : s.n., 2012.

[3] *A Method for Registration of 3-D Shapes*. Besl, Paul J. and McKay, Neil D. 2, Washington, DC, USA : IEEE Computer Society, 1992, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 14, pp. 239-256. 0162-8828.

[4] [Online] http://www.chrobotics.com/wp-content/uploads/2012/11/Inertial-Frame.png.

[5] Welch, Greg and Bishop, Gary. *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA : University of North Carolina at Chapel Hill, 1995Manna, Z., Pnueli, A.: Verification of Concurrent Programs: the Temporal Framework. In Boyer, R. et al., eds.: *The Correctness Problem in Computer Science*. Academic Press, London, (1981), pp. 215–273.

Generating a Saliency Map Using Superpixels

Veronika OLEŠOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
v.nika.olesova@gmail.com

Abstract. Modeling visual attention belongs to one of the most active research areas that have been examined by scientists for over 25 years. Various types of approaches that model this visual attention have been proposed. However, it is still relatively difficult to propose a model that could perfectly simulate human perception. Our work is aimed at creating a model for generating a saliency map, thus a model capable of identifying those areas in the picture that could most likely attract a human attention. We have adopted an existing method which uses superpixels as the basic unit and implemented a modification based on border prior. Experiment proved that our modification achieves better results in precision than the original method using one of the largest data sets.

1 Introduction

In our daily lives we are surrounded by incredible amount of information, which we are not able to process all at once. We need to restrict our attention only on certain area or objects at a time so we can process this information one after another. Scientists have been examining what underlies our attention to help us avoid information overload. They came up with the idea to create a saliency map for a given image that stores information about human visual attention of this image.

The saliency map is a topographically arranged map to represent the saliency of the visual scene and it gives us information about where in the image the areas that attract our attention are. It can reflect several salient objects or areas which are sorted by their saliency.

Saliency map is often used as a prior for a classification system to detect objects. These maps are useful for many applications such as image compression, predicting eye movements, autofocus and visualization.

The main problem of existing models generating saliency maps is that they usually work with specific cases and are not able to cover all of them.

2 State Of The Art

There are a lot of differently oriented models to creating a saliency map that have achieved good performance in predicting human fixations. The most common models that are often used for

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Vanda Benešová, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

comparison are A Model of Saliency-based Visual Attention for Rapid Scene Analysis [4], Graph-Based Visual Saliency [3] and SUN: A Bayesian Framework for Saliency Using Natural Statistics [8]. We will describe the main ideas of these models in this section. In more detail we will analyze another model, called Superpixel-based saliency detection [6], which is the basis of our work.

A Model of Saliency-based Visual Attention for Rapid Scene Analysis proposed by Itti is inspired by the architecture proposed by Koch and Ullman, who came up with the idea that the different visual features should be combined into one single topographically oriented map. Most of the later works use Ittis model for comparison since it is the earliest model of saliency map.

The title of the second mentioned approach is SUN because it depends on the statistics of natural images. The saliency map of this framework can be generated either by bottom-up, top-down or a combination of those approaches. By choosing bottom-up approach, saliency is represented by self-information and by choosing top-down, it is defined as log-likehood.

Graph-Based Visual Saliency consists of three main steps. First, feature maps need to be extracted at multiple spatial scales. To do that, a scale-space pyramid is obtained from image features: intensity, color and orientation, which is similar to model of Itti. The second step is to form an "activation map" using these feature maps. In the final step the activation map is normalized to emphasize the most important information and then combined into a single saliency map.

2.1 Superpixel-based saliency detection

This model [6] consists of three major steps. At the beginning it is important to simplify the input image by using superpixel segmentation and color quantization. Then, similarity between each superpixel has to be found. Finally, the global contrast and spatial sparsity is computed for each superpixel.

The advantage of the superpixel representation is that computational elements are greatly reduced and the segmentation result will be better since superpixels preserve the information about the shape of the object and are more robust to noise.

2.1.1 Image simplification

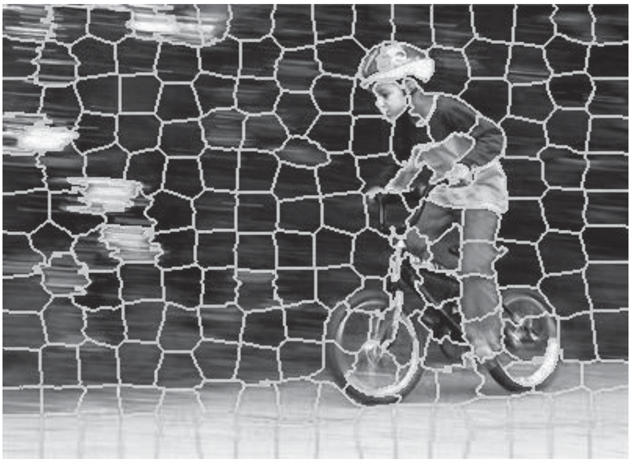


Figure 1. Superpixel segmentation.

The image is converted to the CIE L*a*b*, perceptual uniform color space, which is designed to approximate human vision. The first simplification consists of creating superpixels using SLIC algorithm [2]. This divides a picture into approximately 200 smaller regions. The result of superpixel segmentation using SLIC algorithm can be seen in Figure 1. Then, the number of distinct colors has to be reduced by applying the color quantization. The image histogram is created by quantizing each color into $q \times q \times q$ bins. For each bin, mean color and number of pixels belonging to this bin is computed. Bins that cover more than certain number of pixels are preserved and the rest are merged into ones that have the smallest difference between their quantized colors.

2.1.2 Superpixel similarity

Each superpixel is assigned to a color histogram which is calculated based on the one created in the previous step. The histogram is normalized so that the summation of values in each histogram is equal to 1. Two types of similarities for each pair of superpixels are computed. The color similarity of two superpixels is computed as the sum of intersection of their histograms:

$$Sim_c(i,j) = \sum_{k=1}^{m} \min\{H_i(k), H_j(k)\} \tag{1}$$

The spatial similarity is defined as:

$$Sim_d(i,j) = 1 - \frac{\|\mu_i - \mu_j\|}{d} \tag{2}$$

where d is the diagonal length of the image and μ is the center of the superpixel.
By combining those similarities, the resulting similarity is obtained:

$$Sim(i,j) = Sim_c(i,j) * Sim_d(i,j) \tag{3}$$

2.1.3 Superpixel saliency

Authors [6] suggested that color contrast can be easily seen between the salient object and the background. They also noticed that spatial distribution of salient object superpixels is sparser than background superpixels. Because of this, global contrast of each superpixel and their spatial sparsity are evaluated for measuring the final saliency.

Global contrast of each superpixel is defined as:

$$GC(i) = \sum_{j=1}^{n} W(i,j) \cdot \left\|mc_i - mc_j\right\| \tag{4}$$

where mc is the mean color of superpixel and the weight is defined as:

$$W(i,j) = \left|SP_j\right| \cdot Sim_d(i,j) \tag{5}$$

where $\left|SP_j\right|$ stands for the number of pixels in the superpixel. The spatial sparsity of a superpixel is computed as:

$$SS(i) = \frac{\sum_{j=1}^{n} Sim(i,j) \cdot D(j)}{\sum_{j=1}^{n} Sim(i,j)} \tag{6}$$

where $D(j)$ is a distance between the center of image and the superpixel j.

Results are normalized and refined based on an assumption that superpixels with higher similarity should have more similar saliency values. The final saliency value for each superpixel is defined as the multiplication between refined global contrast and spatial spread.

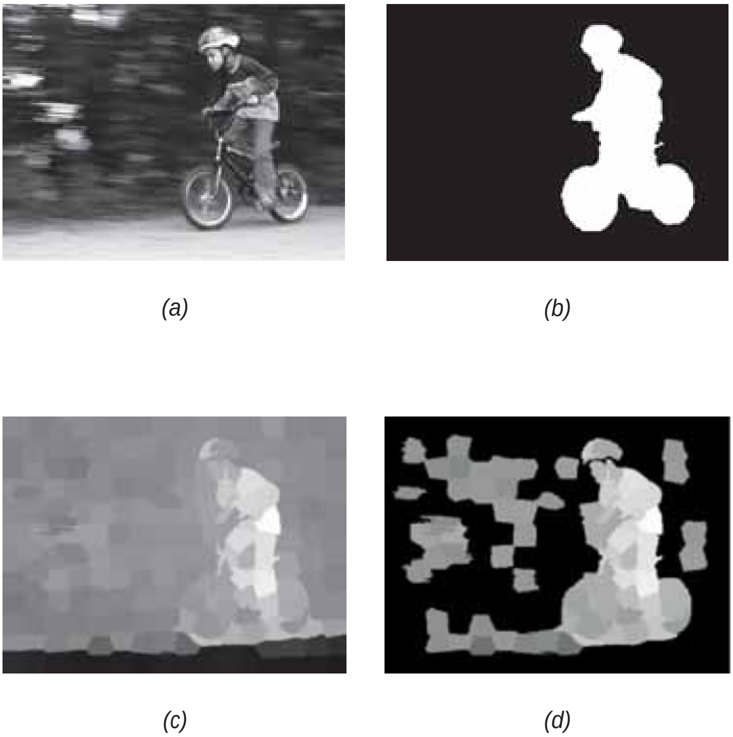
3 Our Contribution

We have extended the original model by adding the border prior, which achieves better results. This prior comes from the basic rule of photographic composition, that is, most photographers will not crop salient objects along the view frame. In other words, the image boundary is mostly background [7].

Huaizu Jiang and others [5] made the following survey: "…we made a simple survey on the MSRA-B data set with 5000 images and found that 98% of pixels in the border area belongs to the background."

However, if there is a salient object that only slightly touches the boundary, the whole object could be missed.

In our algorithm we label the superpixels that touch any of the image borders as background and find other superpixels that are very similar to them. Each of these superpixels is considered background and its saliency is automatically zero. In the Figure 2 we can see the difference between the saliency map which uses this prior and the one that does not.



(a) (b)

(c) (d)

Figure 2. (a) Original image, (b) ground truth, (c) saliency map without border prior, (d) saliency map with border prior.

4 Test and Results

The MSRA[1] database is the largest object database containing 20 000 images in set A and 5 000 images in set B. Achanta [1] has created the database[2] containing 1 000 manually segmented ground truths corresponding to 1 000 images from the set B. To test our work we generate a saliency map for each of 1000 images from MSRA database and compare them to their ground truth. However, images in the MSRA database contain only a single salient object and most of them are large and near the image center.

[1] Downloaded from *http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm*

[2] Downloaded from *http://ivrgwww.epfl.ch/supplementary_material/RK_CVPR09/*

Precision and recall are statistical measures that are very often used to measure how well the saliency model is able to predict human fixations. Precision is a measure of accuracy and recall is a measure of completeness.

First a saliency map is computed for each test image and then a segmentation is generated by simply thresholding the map by assigning the pixels above the given threshold as salient (white background) and below the threshold as non-salient (black background). Pseudo-code for calculating the precision and recall rate looks like this:

```
if (value_of_saliency_map > threshold)
    {
          segmented_foregound_pixels++;
          if (value_of_ground_truth != 0)
                hit++;
    }

    if (value_of_ground_truth != 0)
          ground_truth_foreground_pixels++;
precision = hit / segmented_foregound_pixels;
recall = hit / ground_truth_foreground_pixels;
```

By sliding the threshold from minimum to maximum value, we achieved the precision-recall curves that we use for the comparison between various methods.

We have experimented with the saliency maps obtained by the algorithm without and with the border prior implemented. The graph of comparison between those two algorithms (original model and border prior) is in Figure 3. We can see that our algorithm updated with the border prior achieves better results in precision. There is no saliency map that would have the precision smaller than ~0.55. In addition, this graph shows the difference between another 3 models including Graph-Based Visual Saliency (GB) [3], A Model of Saliency-based Visual Attention for Rapid Scene Analysis (IT) [4] and Frequency-tuned Salient Region Detection (IG) [1]. We used datasets[2] containing 1000 saliency maps for each model created by Achanta et al.

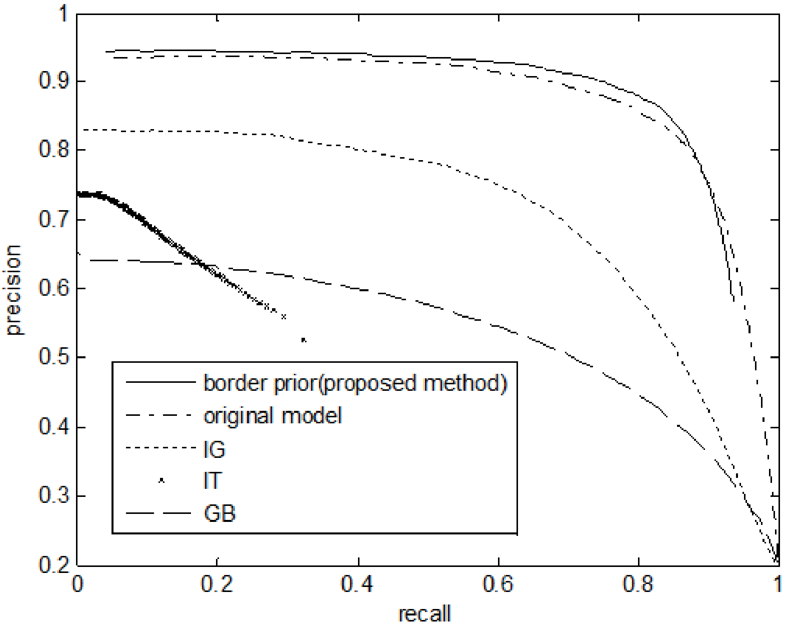


Figure 3. Comparison between different saliency models.

5 Conclusion and Future Work

We have presented a modification to the existing method [6] to creating a saliency map. This modification outperforms the original method achieved by adding a border prior. We compared our method to 4 other models with the very promising results.

However, results provided by this method are still not perfect and other modifications are required. We assume that using only color contrast, spatial distribution and border prior is not enough and it would be vital to use higher features such as face detection. The biggest drawback is that this method prefers the objects that are located in the center of the picture which is caused by the spatial sparsity. Although it is important in some cases, in others it is undesirable. The solution to this problem could be assigning less weight to this spatial sparsity or finding another location instead of center of the picture.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Achanta, R., Hemami, S. et al.: *Frequency-tuned Salient Region Detection*. 2009.

[2] Achanta, R., Shaji, A. et al.: *SLIC Superpixels.* EPFL Technical Report no. 149300, June 2010.

[3] Harel, J., Koch, C., Perona, P.: *Graph-Based Visual Saliency*. 2007.

[4] Itti, L., Koch, C., Niebur, E.: *A model of saliency-based visual attention for rapid scene analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254–1259, 1998.

[5] Jiang, H., Wang, J. et al.: *Salient Object Detection: A Discriminative Regional Feature Integration Approach*. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2013.

[6] Liu, Z., Meur, O., Luo, S.: *Superpixel-based saliency detection.* 2013.

[7] Wei, Y., Wen, F. et al.: *Geodesic Saliency Using Background Priors*. 2012. ISBN: 978-3-642-33711-6.

[8] Zhang, L., Tong, M. H., Marks, T. K. et al.: *SUN: A Bayesian framework for saliency using natural statistics. Journal of vision*, vol. 8, no. 7, pp. 32.1–20, Jan. 2008.

Light Field Rendering in Web Browsers

Michal POLKO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
michal.polko@gmail.com

Abstract. In this paper, we describe a method for transferring and interactive rendering of previously acquired light fields in web browsers. The main challenge to address is to handle constraints imposed by browser environment and graphics hardware. Therefore, to transfer light field data to web browsers, we propose compression, using both known image compression formats and video compression methods. To render and enable manipulation of light fields in web browsers, we also describe use of cache in graphics memory, since light field data has much higher memory requirements than off-the-shelf graphics hardware can provide.

1 Introduction

The light field is a 4D function that describes the amount of light travelling in every direction through every point in space. It is useful for many practical applications, such as processing scenes without creating 3D models of them or creating 3D models of the acquired scenes [4].

In recent years, the light field has become known for its commercial application in photography[1]. The main advantage of the light field photography is the ability to modify image parameters, such as focus, aperture or viewpoint, after exposure [7].

However, processing of the light field poses a great challenge. A lot of research has been done on its acquisition with specialized devices [7, 8], compression [11] or rendering [7] with the use of graphics hardware [2].

The main contribution of this paper is an adaptation of techniques for light field processing to web browser environment. In recent years, web browsers have gained a lot of functionality and become a popular platform for applications, even complex ones.

To enable interactive rendering of light field in web browser, we exploit ubiquitous graphics hardware with the use of WebGL standard[2], which is the driving force behind many web applications that process image data.

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Peter Drahoš, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

[1] https://www.lytro.com/

[2] https://www.khronos.org/registry/webgl/specs/1.0/

2 Compression

One of the ways to capture light field is to take a lot of images of the scene from different viewpoints. However, to acquire the light field in acceptable quality (dense angular resolution), hundreds of images have to be taken. The required space for all these images is measured in hundreds of megabytes, or even gigabytes [2] and it is unacceptable to transfer this amount of data through the internet to user's web browser.

Therefore, to overcome bandwidth limitations, suitable compression has to be applied to light field data. Since the images of the acquired light field are very similar, it is a good idea to exploit this similarity as in [11].

2.1 Our method

We propose a lossy compression method, which is based on using known image compression formats in combination with methods used in video compression.

The first step of our method is to divide images of the scene into two groups, the I-frames (intra frames), which can be decoded on their own, and P-frames (predicted frames), which contain only difference data (block motion vectors) between I-frames they reference and themselves.

To ensure good quality of encoded output, each P-frame references at least two I-frames. The sample division of these frames is shown in the Figure 1.

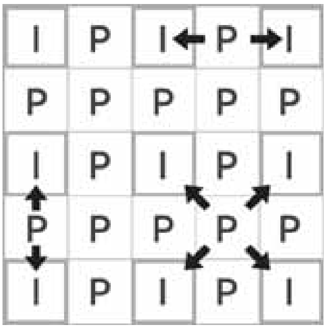


Figure 1. Division of I-frames and P-frames in a light field dataset with 5 x 5 angular resolution. The top-left frame represents an image of the scene taken from the top-left viewpoint, the top-right frame represents the top-right viewpoint, etc. Arrows shows all three options of P-frame to I-frame references.

The I-frames are compressed with JPEG (Baseline) lossy compression. Although this compression does not achieve state-of-the-art compression ratio, its main advantage is widespread support in current web browsers.

The next step of our approach is to divide I-frames and P-frames into blocks of the same size. For each block, we try to find a 2D motion vector M_v that satisfies the following rule: the P-frame image data contained in the block should be as similar as possible to referenced I-frame image data contained in the block with offset M_v. This search is repeated for each referenced I-frame to find the best match.

The resulting data (2D motion vector and I-frame index with best match) are then encoded as an image, where one pixel contains data for exactly one block in the following way:

- Red channel: I-frame index relative to P-frame index
- Green and Blue channel: values of X and Y components of the M_v motion vector.

This image is then compressed with DEFLATE lossless compression (PNG image format). As is the case with I-frames and their JPEG compression, we have chosen PNG format for its widespread support in web browsers.

3 Rendering

Rendering light field from a set of images with focus, aperture and viewpoint parameters requires a simple shift-to-add algorithm [7]. Images necessary for the algorithm are determined by viewpoint and aperture parameters only; focus parameter affects just their relative position.

The main issue with light field rendering on graphics hardware is the limited amount of graphics memory. It is not possible to have all images of the light field present in graphics memory, since the images typically have much higher memory requirements than off-the-shelf graphics hardware can provide [2].

3.1 Our method

Our approach is based on [2] and adapted for web browser environment and our method for compression from Section 2.1.

3.1.1 Cache

If we want to render the light field on graphics hardware with the shift-to-add algorithm, we have to move the necessary images from main memory to graphics memory.

However, it is not possible do so on-demand, because the bus bandwidth is limited and the web browser has to decode the image file before uploading its data to graphics memory. This processing takes too much time and responsiveness of application would suffer.

Therefore, we create an image cache in graphics memory. It contains not only images necessary for rendering with current render parameters, but also images that might be needed in future if the user changes render parameters. Images for future use are uploaded while the user is idle, so that the responsiveness of application is preserved.

Our approach is connected with the compression method outlined in Section 2.1. The cache contains both I-frames and compressed P-frames, which has the following advantages:

- Compressed P-frames are much smaller than reconstructed ones, which means faster transfer between main memory and graphics memory, and a smaller memory footprint.
- Reconstruction of P-frames on graphics hardware is much faster in comparison with general purpose processor.

The disadvantage is increased rendering time, since it is necessary to perform the reconstruction of P-frames at each render.

To store images in graphics memory, we use the virtual textures technique [1]. We use two virtual textures, one for I-frames with size up to 8192 x 8192 and another for P-frames with fixed size 1024 x 1024.

We also use an additional texture that acts as an indirection table for I-frame and P-frame virtual textures. For each image index, it contains the X and Y position of the image (if present) in the corresponding virtual texture, encoded in the Red and Green color channels, respectively.

3.1.2 Cache management

To determine which images should be present in the cache, we could use one of the cache management algorithms, such as LRU (Least Recently Used), but they are generally not suitable for our problem, since they do not deal with prefetching [2].

Our approach to cache management is simple:

1. Upload all images necessary for rendering with current parameters.
2. If there is a free space in the cache, speculatively load images around the images uploaded in step 1 (see Figure 1 for illustration), until the cache is full.

4 Evaluation

4.1 Compression

To evaluate our proposed compression method, we have created a sample encoder in the Go language that implements the format outlined in Section 2.1 with the following settings:

- P-frame block has size of 16px × 16px.
- The motion vector search is done with a brute-force algorithm in range <-8px, 8px> in both axes with 0.25px precision. Bilinear interpolation is used to obtain pixel values in subpixel precision.
- To compute similarity of two image blocks, SAD (sum of absolute differences) algorithm is used.
- JPEG quality is set to 75.

To further reduce file size, the encoder optimizes output of standard Go image libraries and removes unnecessary metadata from image files.

The compression test was performed on three light field datasets:

- "Chess", from [9], is an acquired scene of a chess board.
- "Cards", from [9], is an acquired scene of a crystal ball and cards.
- "Sintel", from [3], is constructed from rendered images.

The results of our test are shown in the Table 1. To measure reconstruction quality of our compression method, we have calculated the following values for each image in dataset:

- Average of PSNR (peak signal-to-noise ratio) applied to R, G and B channels. Good results should be generally achieved if PSNR values are above 30 dB, but it depends on the specific image.
- Average of SSIM (structural similarity) index applied to R, G and B channels. The SSIM index measures similarity between images with respect to human visual perception, so it should provide more accurate information than PSNR [10]. The result is value between -1.0 and 1.0, where 1.0 means the images are identical.

Table 1. Results of the compression test.

	Chess	Cards	Sintel
Spatial resolution	848 x 480	768 x 768	768 x 768
Angular resolution	17 x 17	17 x 17	19 x 19
Total images	289	289	361
Uncompressed raw size	337 MB	488 MB	609 MB
Compressed size	3.41 MB	8.85 MB	9.64 MB
Compression ratio	1 : 99	1 : 55	1 : 63
Average I-frame PSNR	39.05 dB	33.67 dB	33.43 dB
Average I-frame SSIM	0.999	0.996	0.972
Average P-frame PSNR	37.91 dB	32.35 dB	33.41 dB
Average P-frame SSIM	0.999	0.994	0.974

4.2 Rendering

4.2.1 Memory usage

The virtual textures technique used in our approach has fixed memory requirements. We use two RGB textures, 8192 x 8192 and 1024 x 1024, which means memory usage of 192 MB + 3 MB. The memory usage of the indirection table texture is negligible.

4.2.2 Performance

Our rendering approach has high demands on memory bandwidth and texturing performance of graphics hardware, since it uses a number of texture fetches for:

- bilinear interpolation between images
- rendering I-frames (2 fetches per pixel per image)
- reconstruction and rendering of P-frames (4 fetches per pixel per image)

Therefore, it is not well suited for graphics hardware with highly restricted memory bandwidth, such as the integrated graphics solutions found in most modern CPUs.

To evaluate its performance, we have measured FPS (frames per second) of Google Chrome 32 web browser, while rendering the "Chess" light field dataset with different aperture values. The test was performed on:

- Desktop with Windows 7, equipped with Intel Core i7 2.8 Ghz and AMD Radeon HD 5850 (1 GB of dedicated graphics memory).
- Notebook with OS X 10.9, equipped with Intel Core i5 1.7 Ghz and integrated Intel HD 3000 (384 MB of shared graphics memory).

The results of the test are shown in the Figure 2.

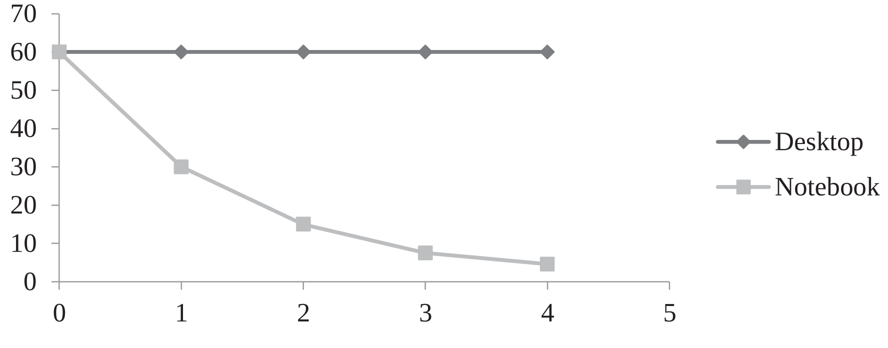


Figure 2. Results of the performance test. Horizontal axis represents aperture value (higher value means more images have to be processed); vertical axis represents FPS value (capped at 60 FPS).

5 Conclusion

We have presented a method for light field compression, implemented a sample encoder and evaluated its performance. The results show significant decrease in file size requirements for three different light field datasets, while preserving high reconstruction quality. In all presented cases, the final size of the dataset is acceptable for transferring over the internet.

For further work, our approach to compression has significant room for improvement in terms of compression ratio and reconstruction quality. It is reported that 1:1000 compression ratio is achievable [5]. In our approach, better performance would be enabled by using more advanced image compression, such as HEVC-MSP [6], but we would also have to create our own encoder, since no web browser supports it natively.

Similarly, our method for exploiting similarity between images could be improved with more ideas from advanced video compression. Currently, the method is limited to light fields with dense angular resolution, because the P-frames only contain motion vectors, so any greater movement between frames will result in mediocre reconstruction quality.

We have also described a method for rendering of light field with caching in graphics memory. Evaluation of this approach shows a perfect 60 FPS result on fast graphics hardware while using restricted amount of memory for the cache.

However, the main issue of our approach is performance on slow graphics hardware. This could be improved with more efficient approaches, but then the limitations are web browser capabilities and restrictions of the WebGL standard.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Andersson, S., Göransson, J.: Virtual Texturing with WebGL. Gothenburg: Chalmers University of Technology, (2012).

[2] Birklbauer, C., Opelt, S., Bimber, O.: Rendering Gigaray Light Fields. In: *Computer Graphics Forum*, (2013), vol. 32, no. 2.4, pp. 469–478.

[3] Johannes Kepler University of Linz, (2013). JKU | CG Research Area: Light Fields http://www.jku.at/cg/content/e60566/e155404 (2013-12-01)

[4] Levoy, M.: Light Fields and Computational Imaging. In: *Computer*, (2006), vol. 39, no. 8, pp. 46–55.

[5] Lütolf R.: A Multiresolution Representation for Light Field Acquisition and Processing. Zürich: ETH Zürich, (2004).

[6] Mozilla Corporation, (2013). Lossy Compressed Image Formats Study (October 2013) http://people.mozilla.org/~josh/lossy_compressed_image_study_october_2013/ (2013-12-01)

[7] Ng, R.: Digital Light Field Photography. Stanford University, (2006).

[8] Stanford University, (2013). The (New) Stanford Light Field Archive http://lightfield.stanford.edu/acq.html (2013-12-01)

[9] Stanford University, (2013). The (New) Stanford Light Field Archive http://lightfield.stanford.edu/index.html (2013-12-01)

[10] Wang. Z, Bovik, A. C.: Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures. In: IEEE Signal Processing Magazine, (2009), vol. 26, no. 1, pp. 98–117.

[11] Zhang, C., Chen, T.: A Survey on Image-Based Rendering – Representation, Sampling and Compression. In: Signal Processing: Image Communication, (2004), vol. 19, no. 1, pp. 1–28.

Object Recognition Using Superpixels and Feature-Based Methods

Tomáš KUNKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
tomas.kunka@gmail.com

Abstract. Visual object recognition is trivial task for humans. For computer vision this tasks are quite challenging. In this paper we deal with object recognition using local features-based methods. We decided to process image on higher level than pixels. First, we segment the image into selected number of superpixels. These superpixels are used as basic processing units of image. We compute different features for each superpixel. We want to use these features for comparing superpixels in predefined pattern. Our own descriptor is then based on measure of similarities between those compared superpixels. This way, we want to capture also information about superpixel neighbourhood. Features used for superpixel description are also the key part of our research.

1 Introduction

Sight is one of the most important human senses. Our vision enables us to perceive and understand the world around us. The purpose of computer vision is to program a computer to understand a scene or features in an image. Human vision analyzes objects in the 3D space with help of cues such as object color or contour. On the other hand, computers work with 2D output of image sensors, which cause a great loss of data, making problems from this area even harder.

Object detection and recognition algorithms are computationally complex. Gradual improvement of computing hardware also increased interest in this area. Nowadays, even personal computers have enough computing power to run complex algorithms of computer vision. There are many areas where object detection and recognition is necessary. Popular area is image retrieval based on content, where similar images can be searched for based on the query image. Automotive industry is another great example of applications of computer vision. Advanced driver assistance systems are more and more often embedded into the mass production of cars. Main functions of such systems are pedestrian or animal detection, car detection or even recognition of incoming road signs. Nowadays each government or company wants to guarantee the safety of their people. Surveillance systems become essential tool for fulfilling this serious task. Computer vision enables these systems to automatically detect moving objects in the scene and recognize dangerous events without human intervention.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Vanda Benešová, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 355–360.

Many approaches to this task have been implemented over multiple decades. In our work we use feature-based approach. In this approach selected features are extracted from the objects to be recognized and the images to be searched. Selecting the right features is very important and is essential part of our research. Many object detection and recognition systems operate on pixel level, but we use innovative method based on pixel clusters – superpixels.

2 Related work

The idea of solution based on superpixels is not completely new. Liu et al. [4] propose a novel re-identification method based on superpixel features. They extract local C-SIFT features based on superpixels as visual words, and use appearance details to describe detecting objects. For purpose of fast person search, they build a TF-IDF vocabulary index tree.

Superpixels are used in wide range of applications. In the paper by Haas et al. [3] a 2D medical image retrieval system which employs interest points derived from superpixels in bags of visual words is presented. Authors show that using the centers of mass of superpixels as interest points yields higher retrieval accuracy when compared to often used Difference of Gaussians.

Method proposed by Pantofaru et al. [5] picks straightforward approach in combining multiple segmentations. They explore the problem of how to best integrate multiple information from multiple bottom-up segmentations of an image to improve object recognition robustness. Combining multiple sets of superpixels, created by multiple segmentations, provide better options for feature extraction.

Fulkerson et al. [2] propose a method to identify and localize object classes in images. They do not operate at pixel level. Instead they use superpixel as the basic unit of class segmentation. They construct a classifier on the histogram of local features found in each superpixel. They regularize this classifier by aggregating histograms in the neighbourhood of each superpixel and then refine results further by using the classifier in a conditional random field operating on the superpixel graph.

Tighe et al. [7] presents effective nonparametric approach to the problem of image parsing, or labelling image regions. In this work authors specify problem to labelling superpixels produced by bottom-up segmentation. It works by scene level matching with global image descriptors, followed by superpixel-level matching with local features (*Table 1*) and efficient Markov random field optimization for incorporating neighbourhood context. These features give us a good example of features with high description value.

Table 2. Superpixel features[7].

Type	Name	Dimension
Shape	Mask of superpixel shape over its bounding box (8 x 8)	64
	Bounding box width/height relative to image width/height	2
	Superpixel area relative to the area of the image	1
Location	Mask of superpixel shape over the image	64
	Top height of bounding box relative to image height	1
Texture/SIFT	Texton histogram, dilated texton histogram	100 x 2
	SIFT histogram, dilated SIFT histogram	100 x 2
	Left/right/top/bottom boundary SIFT histogram	100 x 4
Color	RGB color mean and std. dev	2 x 2
	Color histogram (RGB, 11 bins per channel),dilated histogram	33 x 2
Appearance	Color thumbnail (8 x 8)	192
	Masked color thumbnail	192
	Grayscale gist over superpixel bounding box	320

3 Proposed method

Solving problem of object detection and recognition at pixel level is computationally quite complex task. Exploring all image pixels separately can be very expensive. As we process bigger and bigger images, number of pixels is increasing rapidly. We need to employ different approach and reduce the number of processed units. We decided to use segmentation algorithms on input image pixels and cluster them into bigger segments effectively reducing their number and also creating more natural representation for human perception. Finally we are working with created segments called superpixels. Creation of superpixels is based on clustering pixels with similar color and spatially close pixels.

3.1 Approach

We decided to solve the problem of object recognition in several steps. Scheme of solution is outlined in Figure 1.

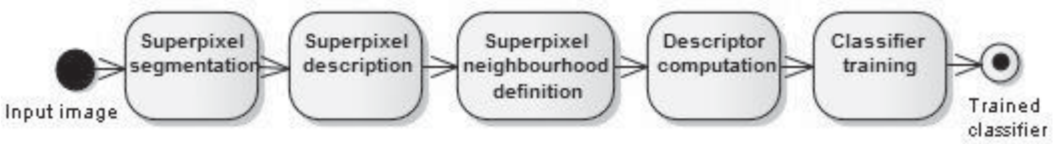


Figure 1. An overview of our solution.

1. **Superpixel segmentation** – dividing input image to smaller parts. Generated segments should follow contours of objects in the image.
2. **Superpixel description** – we need to store information about each superpixel. Our work is focused on finding right features with high description value. Selecting right features is essential for distinguishing superpixels belonging to different classes.
3. **Superpixel neighbourhood definition** – we need to capture also superpixel neighbourhood as it provides additional information useful for recognition. Neighbourhood is formed with spatially close superpixels picked with special pattern. Closer description of this patter is in section 3.4.
4. **Descriptor computation** – descriptor is build from values of similarity measure computed between superpixels picked by pattern.
5. **Classifier training** – classifier will be trained with data generated in the step of descriptor computation.

3.2 Superpixel segmentation

First of all we need to over-segment input image, thus creating desired number of superpixels. Output of this process is input image enriched with contours of superpixels. We selected this contours to be the main representation of superpixels. For each superpixel created by over-segmentation algorithms we create an instance of Superpixel class. This class serves as container storing information about contour in vector of points. Additionally we store center of mass (x, y coordinates) and of course selected local features, which will be later used for comparing two superpixels and determining measure of similarity.

We decided to experiment with image segmentation and use three different segmentation algorithms. First algorithm used is *Simple iterative clustering* (SLIC) [6] , one of the latest and fastest superpixel algorithms. It clusters pixels in the combined 5D color and image plane space. Algorithm generates coherent, compact and nearly uniform superpixels (Figure 2).

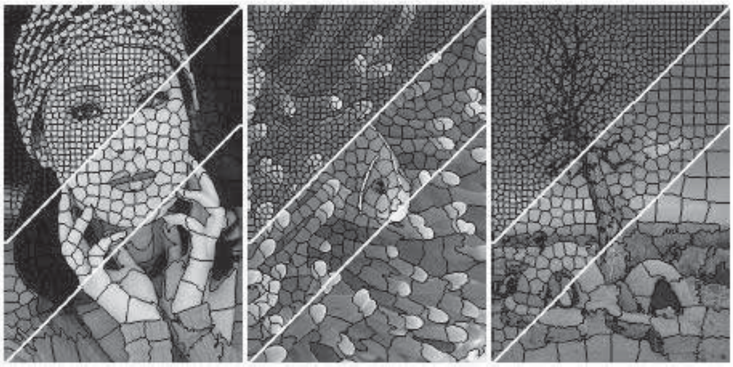


*Figure 2. Image segmented using our algorithm into superpixels of (approximate) size 64, 256, and 1024 pixels. The superpixels are compact, uniform in size, and adhere well to region boundaries.[6]***.**

As second segmentation algorithm we choose *Morphologic Superpixel Segmentation* (MSS) [8]. Authors propose fast algorithm comparable to state-of-art algorithms which can be used in near real- time applications for object segmentation and object recognition. Paper provides a fast method of segmenting an image into superpixels by morphological approach. Morphological image reconstruction is used to eliminate irrelevant spatial local extreme intensities in the image. Using this method also remove irrelevant edges. Then markers for morphological watershed segmentation are generated. For that purpose, flooding procedure in a marker input image is derived from difference between the original image and image blurred with Gaussian filter.

Last selected algorithm is proposed by Arbelaez et al. [1]. Work investigates two fundamental problems in computer vision: contour detection and image segmentation. They present contour detector which combines multiple local cues into a globalization framework based on spectral clustering. Output of this detector is then transformed in segmentation algorithm into a hierarchical region tree. This way, problem of image segmentation is reduced to that of contour detection.

3.3 Superpixel description

Each superpixel needs to be described with several statistical features. This description is used in later phase when we need to compare two superpixels. In order to do this, we need features that can unambiguously distinguish superpixels from different classes. This shown as nontrivial task and therefore we focus our research on evaluation of features. So far we compute different features for each superpixel. Features are divided into following categories:

- Color
  - mean value and standard deviation of RGB, HSV, HLS and Lab image
  - RGB color histogram
- Texture
  - entropy
  - HOG features
  - Gabor features
- Shape
  - minimal bounding box(width, height, angle)

We develop testing framework for an experimental comparison selected features. Superpixels described by features useful for object recognition should have low intra-class variation and high inter-class variation. We assume that superpixel lying inside some object should have higher similarity measure with neighboring superpixels which also lie inside the object, than with neighboring superpixels lying outside the object (Figure 3).

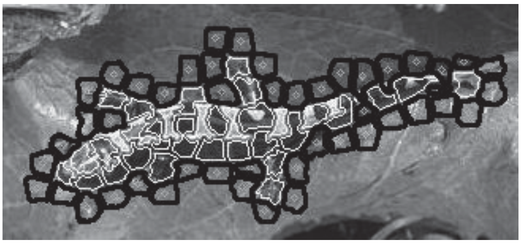


Figure 3. Superpixels(white – lying inside object, black – lying outside object).

3.4 Descriptor computation based on superpixel neighbourhood

Our purpose is to propose our own descriptor for superpixels suitable for later use in object detection and recognition frameworks. In our descriptor we want to capture information also about neighbourhood of described superpixels. We do not want to select only spatial closest superpixels, but we experiment with selection of such neighbourhood with different patterns (Figure 4*).*

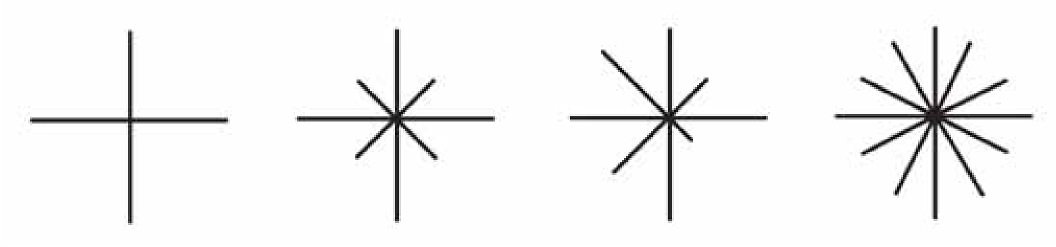


Figure 4. Example of different possible patterns.

We place this pattern in selected superpixel – concretely to centre of mass (*Figure 5.*). Endpoints of pattern lines select other superpixels in the neighbourhood of superpixel. Then we compute a similarity between central superpixels and those selected by our pattern. These similarities are the core concept of our descriptor.

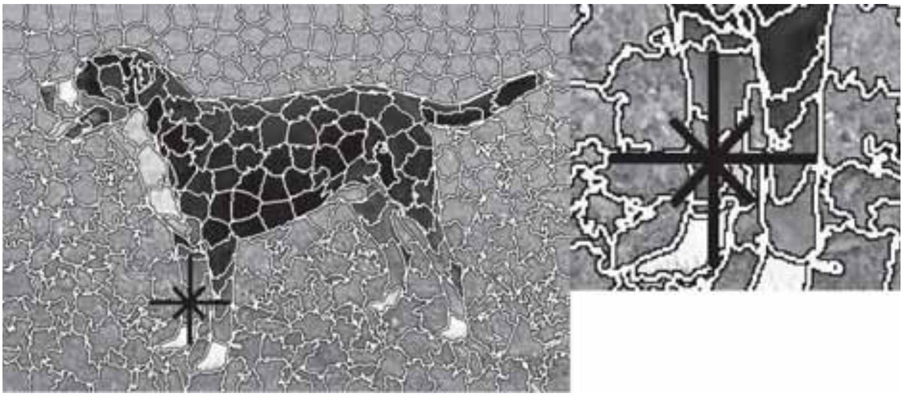


Figure 5. Selection of neighbourhood superpixels with star pattern.

4 Conclusions

In this paper we present a new approach for building descriptor which can be used in applications of object recognition. We over-segment an input image to create number of uniform areas – superpixels. Superpixels are used as basic units for work with images instead of pixels. In order to build proposed descriptor based on similarities of superpixel in a small neighbourhood, we have to do an experimental comparison of features used for superpixels description. With right features we can safely compare superpixels. Compared neighbouring superpixels are not selected only by spatial distance, but with special pattern which is also part of our research.

We want to evaluate variety of features suitable for superpixel description. For experiments we want to use publicly available reference image database[1].

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Arbelaez Pablo, Maire Michael, Fowlkes Charless, and Malik Jitendra. 2011. *Contour Detection and Hierarchical Image Segmentation*. IEEE Trans. Pattern Anal. Mach. Intell. 33, 5 (May 2011), 898-916.

[2] Fulkerson B., Vedaldi A., and Soatto S., *Class segmentation and object localization with superpixel neighborhoods*, in Proc. Int. Conf. Comput. Vis., 2009, pp. 670–677.

[3] Haas, S. and Donner, R. and Burner, A. and Holzer, M. and Langs G, *Superpixel-based Interest Points for Effective Bags of Visual Words Medical Image Retrieval*. Proceedings of the MICCAI 2011 Workshop on Medical Content-based Retrieval for Clinical Decision Support (MCBR-CDS 2011), Toronto, Canada.

[4] Liu C., Zhao Z., Person *Re-identification by Local Feature Based on Super Pixel*. ; In Proceedings of MMM (1). 2013, 196-205.

[5] Pantofaru Caroline, Schmid Cordelia, Hebert Martial. 2008. *Object Recognition by Integrating Multiple Image Segmentations*. In *Proceedings of the 10th European Conference on Computer Vision: Part III* (ECCV '08), David Forsyth, Philip Torr, and Andrew Zisserman (Eds.). Springer-Verlag, Berlin, Heidelberg, 481-494.

[6] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, *SLIC Superpixels*, EPFL Technical Report no. 149300, June 2010.

[7] Tighe Joseph, Lazebnik Svetlana. 2010. *Superparsing: scalable nonparametric image parsing with superpixels*. In *Proceedings of the 11th European conference on Computer vision: Part V*(ECCV'10), Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer-Verlag, Berlin, Heidelberg, 352-365

[8] Wanda Benešová, Michal Kottman. 2013. *Fast Superpixel Segmentation Using Morphological Processing*

[1] http://research.microsoft.com/en-us/projects/objectclassrecognition/

Evaluating the State of the Board Game from a Still Image

Michal LIHOCKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
michal.lihocky@gmail.com

Abstract. This work analyzes the field of computer vision within the domain of board games while it focuses mainly on the experimental evaluation of selected methods and algorithms for the purpose of retrieving the information about the game. In this paper we propose a new approach for the extraction of information from the board game and introduce a prototype capable of evaluating the state of such game from a still image using the proposed approach. Our prototype addresses the rare domain of race games played on a cruciform board and it is therefore first of its kind.

1 Introduction

Over last decades, there has been significant growth of the usage of various devices that allow us to capture images and videos. As the amount of available data of such as kind grows, so grows the attention that is being paid to the possible practical usage of the information contained within. Extracting information from images and videos for various purposes is covered by the fields of image processing and computer vision, influence of which can be observed in many different domains.

Our work is focused on the computer vision within the domain of board games. The use of various approaches and methods of computer vision within this domain has been analyzed and many of these have been selected for the experimental evaluation with attempt to make a case study while implementing the prototype capable of determining and evaluating the state of the board game based on the information extracted from the still image.

In this paper, we propose a new domain knowledge-based approach for the extraction of information about the board game from a still image and also introduce and describe a prototype that has been developed for the purpose of evaluating the properties and possible usage of the proposed approach.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Juraj Štefanovič, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

2 Application domain introduction

The attention within the domain of board games is mostly being paid to the *chess* and *Chinese chess* and less to the ancient Asian game called *"Go"*, while the computer vision systems addressing these games are mainly focused either on the development of autonomic chess-playing robots [1,2,3], the game reconstruction from video [4,5] or the development of reactive grasping systems in the game board playing environment [6].

There are several tasks within this domain, for which computer vision can be used. Among the most common ones are:

- determining the position of the game board,
- detection of game pieces,
- classification / recognition of game pieces, and
- determining the positions of game pieces according to the specific game board.

2.1 State of art

Tam, Lay, and Levy [7] have analyzed various approaches for the segmentation of game board fields and they have divided them into multiple groups. According to them, the most frequently used approaches are based on: *line detection*, *corner detection* or *template matching*. Just like Neufeld and Hall [2], they have also concluded that for the chessboard the most suitable are approaches based on the *line detection* taking advantage of the structure of the game board.

Neufeld and Hall have analyzed various approaches for locating the game pieces for the game reconstruction. They have pointed out that most of solutions either use the modified chess set or they are based on the differential image while relying on the initial positions of pieces with no true capability of game piece recognition (such as [1]). Yet, they point out multiple advantages of more robust solutions based on the object recognition despite of their complexity and the difficulty of their implementation (such as Gambit [3]).

2.2 Problem areas

Despite of the amount of effort being put into the research within this domain, there are still many unresolved problems that authors of similar works have to deal with. These are the most common problems that can be observed within this domain:

- negative impact of the camera angle [1,2,3,4,7],
- negative impact of the illumination [1,2,4,8],
- unreasonable environment constraints or the necessity of initial calibration [2,3,4],
- overlapping of game pieces during their detection or recognition [4], and
- problems during the extraction of static pictures from the video such as the detection of move completion [4,5].

3 Our work

For our experimental work, we have decided to target the very rare domain of race games played on a cruciform board. We have chosen the German version of game *Ludo*, which in Slovakia is known under the name *"Človeče nehnevaj sa"*.

Yet, apart from the solely experimental evaluation of selected algorithms and methods of computer vision in a rare and unexplored environment, we have also decided to use the knowledge and experiences of authors of similar works with attempt to address the open problem areas within the domain.

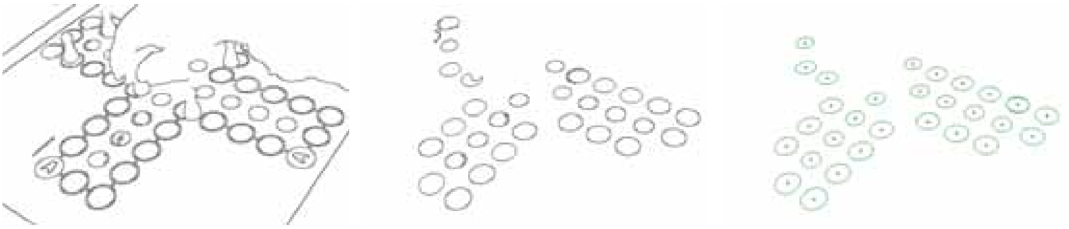
4 Proposed approach

One of the biggest challenges for a computer vision system evaluating the state of the race game is to locate the game board model in a scene. Once the exact position of game board is found, the game pieces can be found and classified and eventually the score for each player can be calculated.

4.1 Retrieving the position of the game board

A possible approach to locating the game board might be template matching or feature matching using algorithm such as RANSAC. The game board is treated as a whole (i.e. top-down approach) and although its position can be determined, no other information will be obtained.

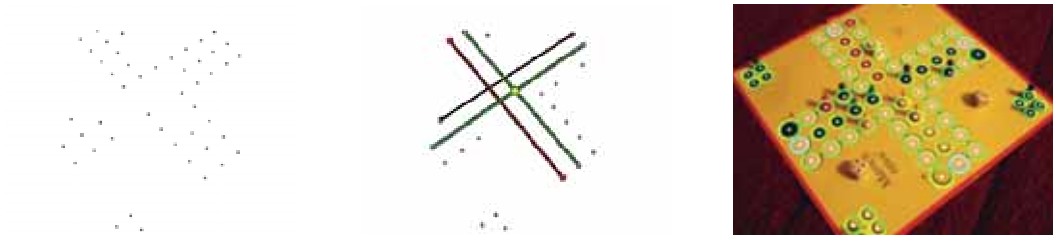
We have used the bottom-up approach, which retrieves the information about some fields and uses it to locate the game board. We take advantage of the elliptic shape of the game fields and use contour analysis to obtain the empty fields. At first we use morphological gradient and Canny edge detector to extract contours. Then we filter out the unwanted contours and use the ellipse fitting algorithm, ensuring that only contours enclosing fields have been retrieved (see Figure 1). These fields are *presumably empty* since the shape of the occupied fields would be affected by the game pieces placed on them.



a) b) c)

Figure 1. The extraction of empty fields: a) contours retrieved from the Canny edge detector b) contours after the unwanted contours have been filtered c) ellipses that have been fit to these contours.

After the set of *presumably empty* fields have been obtained, we use centers of these fields to locate the game board model (see Figure 2). We take advantage of cruciform shape of the board while looking for the cross in the scene. At first, we obtain the probabilistic Hough lines and their intersections. We then determine the *virtual middle* and find the most appropriate intersection near to this *virtual middle, representing* the *true middle* of the game board, which is used to retrieve the aforementioned cross. The end points of this cross are then used to determine the homography that describes the relation between the game board model and the scene. The Figure 2c shows how the game board model can be warped to the scene's perspective and using the found homography.



a) b) c)

Figure 2. The detection of game board model: a) empty fields extracted from ellipses that have been fit to contours b) the result of analysis of intersections of probabilistic Hough lines c) game board model displayed in the scene after being warped to the scene's perspective using the extracted homography.

However, looking for the *"most appropriate"* intersection is much harder than it might seem. We have developed a robust solution based on the analysis of intersections' displacements while the length as well as the angle between the lines that created the intersection is being analyzed. Examples of results of this behavior are shown on the Figure 3.

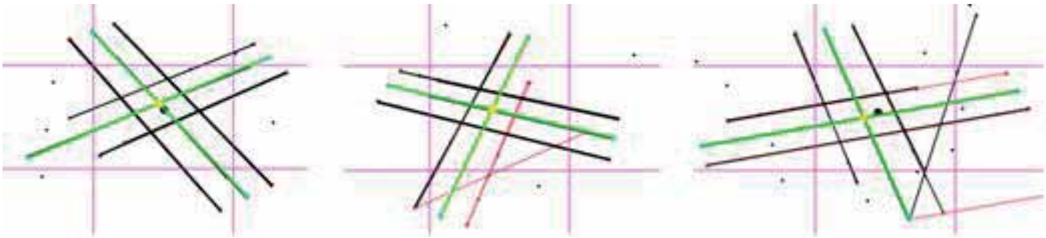


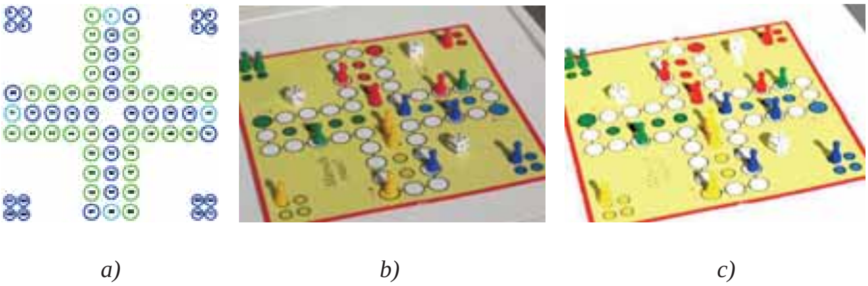
Figure 3. Examples of successful retrievals of key points used for finding the homography (black circle depicts the virtual middle and yellow circle the true middle of game board, red lines are lines that have been ignored due to unreasonable placement and therefore only intersections of black lines are being analyzed).

The output of the aforementioned approach consists of the set of the *presumably empty* fields, including their size and the position within the scene, and the homography, i.e. a matrix describing the relation between the game board model and scene. However, to evaluate the state of the game, we also need to find the game pieces, analyze their concrete position regarding the game board model and analyze their color to determine a player they belong to. Only then, the algorithm that would evaluate the score of each player might take a place.

4.2 Detection of game pieces

Since we have the information about the exact position of every field within a scene, we can iterate over each of those fields and cut off the small segment enclosing this area and analyze it to determine whether there is a game piece or not. However, to maximize the performance of the detection of game pieces with attempt of making this approach usable in real-time applications, the number of such segments needs to be reduced as much as possible.

Although there is a set of *presumably empty* fields available, it might contain false positives, which would have fatal impact on the whole process of evaluating the state of the game. We therefore take advantage of the domain knowledge and analyze the color of the fields within a scene. We have created the complex representation of game board model (see Figure 4a) that apart from other information also distinguishes between white fields and colored fields. The idea is simple: if there is a piece on the field, the inner area of a field cannot be white (see Figure 4b). Relying on this strong assumption we analyze the color of small area in the centers of selected fields to mark fields that are definitely empty. Figure 4c depicts this idea by showing how it would look like if the color of whole image would be modified.



a) b) c)

Figure 4. Taking advantage of the color of the fields and the domain knowledge of the game board. a) game board model b) example of input image c) example of post-processing of color of input image.

Taking advantage of all of the aforementioned information that is available at this point, we obtain the small set of *probably occupied* fields. We then transform the input image to the color-invariant HSV space use this image to extract segments, made solely by the V-channel, that enclose the area of interest for each of these fields (see Figure 5). SVM classifier is then used to determine whether there is a game piece in a given segment or not, yielding the set of *definitely occupied* fields.

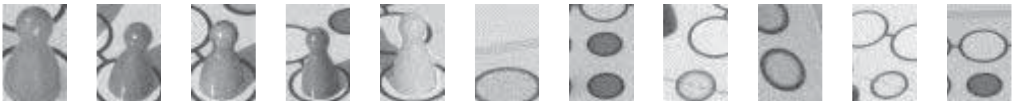


Figure 5. Examples of retrieved segments (showing the both positive and negative samples).

4.3 Evaluation of the state of the game

Once we know that there is a game piece in given segment, we also analyze the H-channel from the HSV space (*hue*) within an area covering the piece. The simple color classifier has been developed that in our case divides the pieces into 4 classes based on the color: blue, yellow, green and red. All of the retrieved information is then used by the custom algorithm that calculates the score for each player, based solely on the positions of his pieces (see Figure 6).

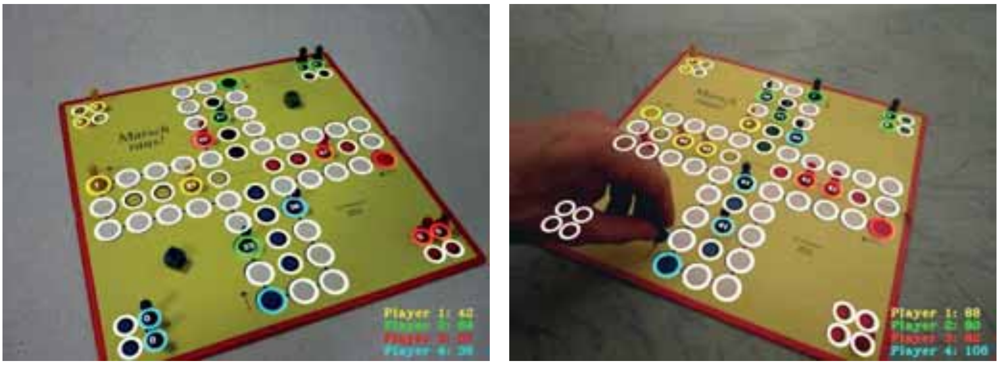


Figure 6. Calculating players' scores using the retrieved information.

5 Analyzing the influence of the camera tilt

In order to analyze the negative influence of the camera tilt, a simple test with a set of 11 images with well-measured tilt in range from 0 to 70 degrees has been used (see Figure 7). The results showed that the accuracy remains high up to the tilt of 50 degrees, after which the count of the detected empty fields decreases rapidly due to the skewed perspective.

6 Conclusions and future work

The unique bottom-up approach for the detection of the board game based on the contour analysis and analysis of intersections of probabilistic Hough lines has been proposed and an experimental prototype capable of evaluating the state of the game board with a potential of becoming usable in a real world has been implemented. This prototype has been written in C++ language, taking advantage of new features proposed in C++11 standard, and using the OpenCV library and the Microsoft Visual Studio 2012 IDE.

The approach has been developed and continuously tested on the dataset of 33 images. Our SVM classifier correctly detects more than 95% of game pieces and the color classifier has incorrectly classified color in less than 2% of cases, which has happened only due to the piece being overlapped by another piece of different color.

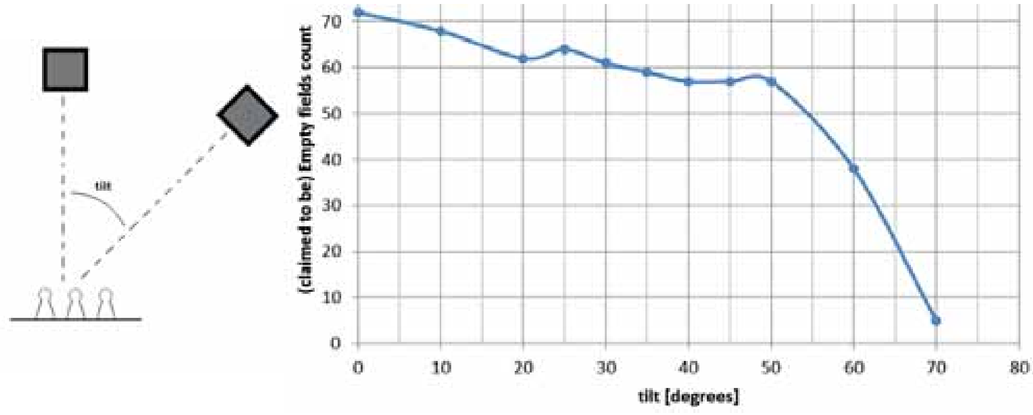


Figure 7. Analyzing the influence of the camera tilt on accuracy of the proposed approach.

Apart from the increasing the precision of aforementioned classifiers, the future work will be focused mainly on the real-time processing of video stream from web camera. Current solution is capable of processing only 3 images per second. We aim for the performance of 5 to 10 processed frames per second, so that usage while playing the game might be demonstrated. We also plan to implement a simple prototype for the detection of the game board based on feature matching using the RANSAC algorithm as an alternative solution comparable to our current approach.

References

[1] Sokic E., Ahic-Djokic M.: Simple computer vision system for chess playing robot manipulator as a project-based learning example, *In International IEEE Symposium on Signal Processing and Information Technology (ISSPIT), Sarajevo*, IEEE, pp. 75-79, 2008

[2] Neufeld, J.E., Hall, T.S.: Probabilistic location of a populated chessboard using computer vision, *In IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Seattle, WA*, IEEE, pp. 616-619, 2010

[3] Matuszek, C., Mayton, B., Aimi, R. et al.: Gambit: An autonomous chess-playing robotic system, *In ICRA, Shanghai*, IEEE, pp. 4291-4297, 2011

[4] Piskorec, M., Antulov-Fantulin, N., Curic, J. et al.: Computer vision system for the chess game reconstruction, *In MIPRO, Opatija*, IEEE, pp. 870-876, 2011

[5] Yanai, K., Hayashiyama, T.: Automatic "Go" record generation from a TV program, *In Proc. of International Multi-Media Modeling Conference Proceedings, Beijing*, IEEE, 2006

[6] Wörn, H., Irgenfried, S., Haase, T.: Multi-fingered reactive grasping with active guided camera systems, *In Proc. of the 27th Annual ACM Symposium on Applied Computing*, ACM, pp. 268-273, 2012

[7] Tam K., Lay J., Levy D.: Automatic grid segmentation of populated chessboard taken at a lower angle view, *In DICTA, Canberra*, IEEE, pp. 294-299, 2008

[8] Fang, J., Kondo, N., Yin, J., Liu, X., Xiao, K.: Illumination invariant Chinese chessboard reconstruction based on color image, *In ICCAS-SICE International Joint Conference, Fukuoka*, IEEE, pp. 461-465, 2009

Smartphone versus Mouse with Keyboard Interaction within Virtual Reality

Maroš URBANČOK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
maros.urbancok@gmail.com

Abstract. Virtual reality brings new opportunities in various fields. Many different devices have been developed to control it, ranging from special controllers to devices for tracking the human body. Most of these control devices use various sensors such as accelerometer, gyroscope, magnetometer and so on. Since smartphones have these sensors too, it opens up the possibility of using them to control VR. In our work, we focused on these possiblities starting from the shortcomings of the few existing solutions. We proposed a control based on metaphors used in other domains and in other devices that we adapted to a smartphone. We created a virtual reality control method, tested it and came to the conclusion that it allows more enjoyable and simpler control for the users, compared to keyboard and mouse.

1 Introduction

More and more often, we encounter virtual reality (VR) in the form of computer games, various computer models or special simulators. Moreover, there are various ways in which we can control it, from the traditional keyboard and mouse (m&k), which may be limiting in certain cases [5], through motion control, to specifically targeted controllers [7]. Such controllers either work with motion sensors such as accelerometer, gyroscope, magnetometer, or their motion is recorded by means of a special camera. Advanced smartphones have some of these sensors as well, which enables us to consider how to use them to control VR.

Accelerometer measures the forces applied to the device. These consist of dynamic (acceleration) and static (gravity) component. The outputs of the sensor are values that represent the sum of these components. If you want to work with only one component, it is necessary to separate it from others. Moreover, the sensor is not capable of detecting the rotation of the device around the axis Z (yaw) [2].

Magnetometer uses the gravitational field of the Earth. The disadvantage of this sensor is that it can be easily influenced by any form of magnetic fields of other devices or anything with a magnetic field.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Alena Kovárová, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Gyroscope, unlike the accelerometer and compass, does not depend on the Earth's gravitational and magnetic fields. It measures the angular acceleration of the device in its three axes. The sensor does not detect the position or tilting of the device, only its movement [8].

Focusing on VR controlling with a smartphone, it is useful to know the current views of the 3D world and the way it is controlled. There are two main views [3]:

- Standard – The environment can be only rotated. This rotation is opposite to the direction of the movement gesture carried out.
- Fixed world – There is created an illusion of being a part of the virtual world.

The second control method is for us more interesting, because individual movements and actions can be mapped to real situations and control methods to real behaviour. That can help us in creating solutions for controlling 3D with a smartphone.

2 Related work

VR control using sensors like an accelerometer is a suitable alternative to the touch screen, which is more natural than m&k control [6].

The accelerometer in a mobile phone, as a device movement tracker, can be used e.g. as a full-fledged game controller in a simple tennis game [4] or even as a 3D space controller [1]. In case of 3D space controller, it is possible to navigate through the VR only in certain directions and the rotation is carried out only in one dimension using the touch screen.

In another application [2] when working with data from the accelerometer, the inaccuracies that arise during movement were taken into consideration. Therefore, a neutral position was created, in which no action was taken and the movement of the device was ignored. The faster the movement outside this area was, the faster the 3D object moved. Furthermore, it was found out that for the full control of the VR it is sufficient for the phone to send information about the position changes 15 times per second, regardless of the number of users (in a multi-user application).

Table 1. Ways to control each DOF in other applications.

DOF	application 1 [1]	application 2 [6]	application 3 [3][1]	application 4 [2]
roll	-	tilting the device left/right[2]	-	The solution uses data from the accelerometer, but there are no further specifications of control methods. But we can say with certainty that it can work only with 3DOF at once and works with the standard view in VR.
pitch	dragging on the touch screen in y-direction	tilting the device forward/backward[2]	tilting the device forward/backward	
yaw	tilting the device left/right	turning device sideways[2]	turning device sideways	
moving forward/ backward	tilting the device forward/-	tilting the device forward/backward[2]	virtual joystick	
moving left/right	-	tilting the device left/right[2]	virtual joystick	
moving up/down	-	turning device sideways[2]	-	

[1] The VR is right in the phone, but the application uses sensors and touch screen to control it. In addition, the standard view in the VR was used.

[2] The authors used three moves of the device, but to achieve the ability to control 6DOF, they used virtual button to switch between the rotation and translation. The control method is not further specified.

Apart from VR control on the remote computer, a mobile application was created, with VR included directly in the mobile phone [3]. In this case, the user controlled a ball and, with certain restrictions, he was able to control the view of the VR, too. Accelerometer and gravity sensor were used to change the perspective. The ball was controlled with a virtual joystick.

Based on the findings we can say with certainty that none of the solutions is able to control all 6DOF simultaneously and only one works with the fixed world view ([1]). In addition, all applications can work with only one VR environment. Here, we see space to create our own solution to control VR using a mobile phone, where we can work with all 6DOF without being limited to only one VR. Table 1 contains an evaluation of control methods of existing applications. We took into consideration all applications that use mobile sensors in control mechanisms, regardless of the focus of the application.

3 Interaction techniques

Based on the existing solutions, we decided to create a solution that allows users to work with all 6DOF and is not limited to one virtual reality. This means that our solution is applicable to any VR. When we designed the control, we focused on intuitiveness, simplicity and naturalness. Therefore, we created several control metaphors. They are based on real life situations and consist of gestures existing in other domains, which we have adapted to the mobile phone. It is by means of these gestures and metaphors that we want to achieve the intuitiveness.

3.1 Metaphors

We focused on three metaphors: car, human walking, and helicopter. These differ from each other in the way of controlling some DOF but also in the way of movement in VR. They are designed for first person view. The first two are adapted to moving on the ground, the third to flying. From available sensors, we decided to use the accelerometer and the touch sensor. Each one is able to control 3DOF independently from the other, so we can simultaneously control all 6DOF. We decided to create a virtual joystick on the touch screen.

3.1.1 Human walking metaphor

In this metaphor it is possible to move forward, backward and sideways. The direction of view resembles moves of the head. By using the virtual joystick on touch screen (Figure 1 (a)-(d)) the view in the VR can be controlled in the directions upward/downward and left/right. By tilting the device (using the accelerometer) the user moves in VR (Figure 1 (e)-(h)).

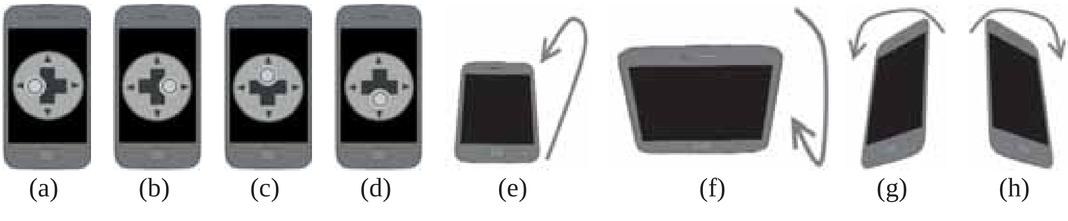


Figure 1. Human walking metaphor: look left (a), right (b), up (c), down (d), move forward (e),move backward (f), move left (g), move right (h).

3.1.2 Helicopter metaphor

The metaphor "helicopter" serves mainly for unrestricted movement in the VR environment, which resembles the real helicopter. It also includes "flying" to resemble flight simulators. In this case, The flight simulator users mostly control by a joystick or keyboard movements of the helicopter and the mouse the perspective. In our metaphor, the user will move in the VR by moving the phone in space and by its tilting (Figure 2 (g) - (l)). An accelerometer is used to detect the change of the position. To change the point of view are used a virtual joystick (Figure 2 (a) - (d)) and the virtual keys (Figure 2(e) - (f)). Joystick alone can control only 2DOF.

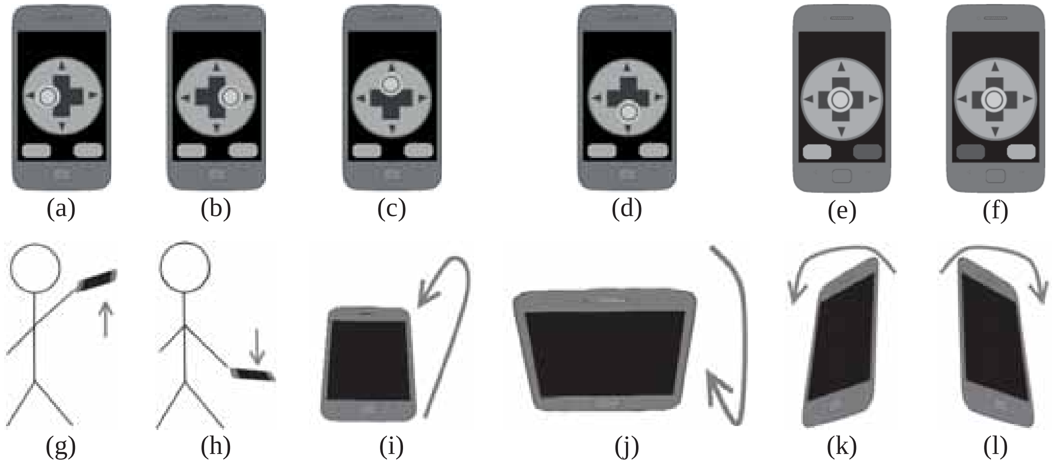


Figure 2. Helicopter metaphor: look left (a), right (b), up (c), down (d), roll clockwise (e), roll counter clockwise (f), move up (g), down (h), forwards (i), backwards (j), left (k), right (l).

3.1.3 Car metaphor

Car metaphor is based on human walking metaphor. This metaphor should resemble driving a car. The device is therefore held as a steering wheel, horizontally with both hands and by turning it we can change the direction of movement (Figure 3 (e) - (f)). Forward movement is performed by tilting the device forward (Figure 3 (g)), just like when pressing the accelerator pedal. We cannot directly map backward movement to any activity while driving but we think that the most natural way is tilting the device backward (Figure 3 (h)). The control of the point of view is also performed with virtual joystick (Figure 3 (a) - (d)).

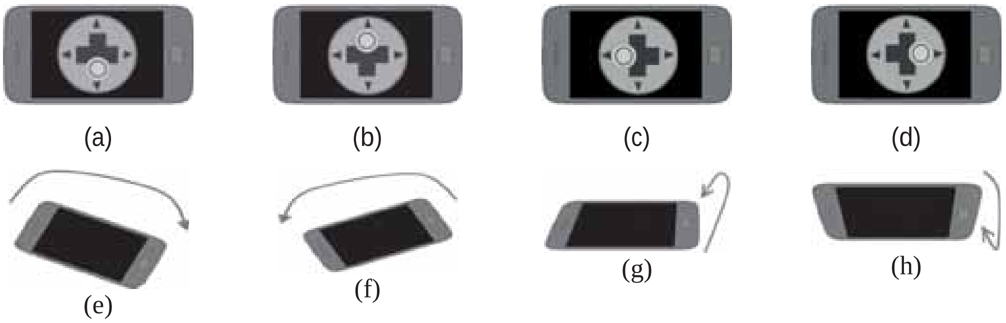


Figure 3. Car metaphor: look down (a),up (b), left (c), right (d), move up (g),down (h),forward (i), backward (j), left (k), right (l).

3.2 System architecture

Architecture of the proposed system is depicted in Figure 4. It shows the individual components and way they communicate with each other. We used client-server architecture. Client is the mobile phone with Android OS. It communicates through a wireless network with the computer that is the server part of the solution. The virtual reality controlled is also on this server and is then displayed on the screen. The virtual reality is a virtual version of the faculty building[3].

[3] http://stavba.fiit.stuba.sk/virtfiit11

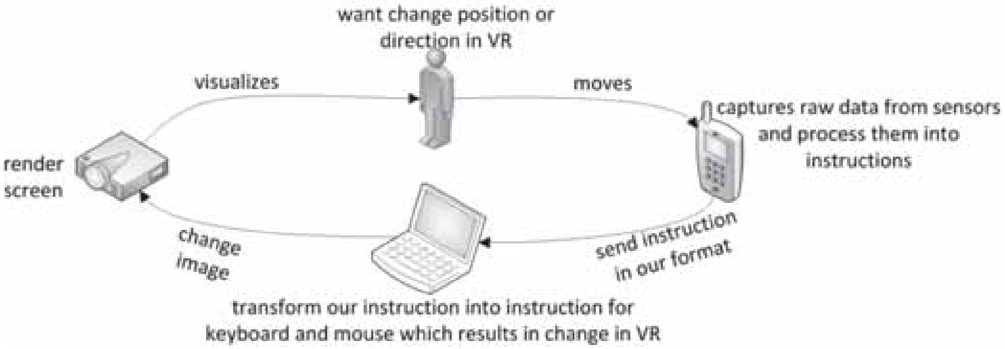

want change position or direction in VR
visualizes
moves
captures raw data from sensors and process them into instructions
render screen
change image
send instruction in our format
transform our instruction into instruction for keyboard and mouse which results in change in VR

Figure 4. System architecture.

4 Evaluation

Before we embarked on the testing itself, we performed pre-test to eliminate the most common functional and design errors (6 people). In the next testing we have done qualitative and quantitative testing with 6 persons (5 men, 1 woman) aged 20-24. 3 of them (#1, #2, #4) were experienced gamers, 1 slightly experienced (#3) and 1 inexperienced (#5). 5 participants were not familiar with controlling a 3D virtual reality with a smartphone, 1 was (# 6).

Before testing, we let the users get familiar with the controls. In quantitative testing, we measured time that took them to move from place A to place B using three techniques: 1. using mobile phone and human walking metaphor , 2. using mobile phone and car metaphor, 3. using m&k with the human walking metaphor. We made 6 measurements for each participant and each technique. The average values that the participants reached in each technique are in Figure 5.

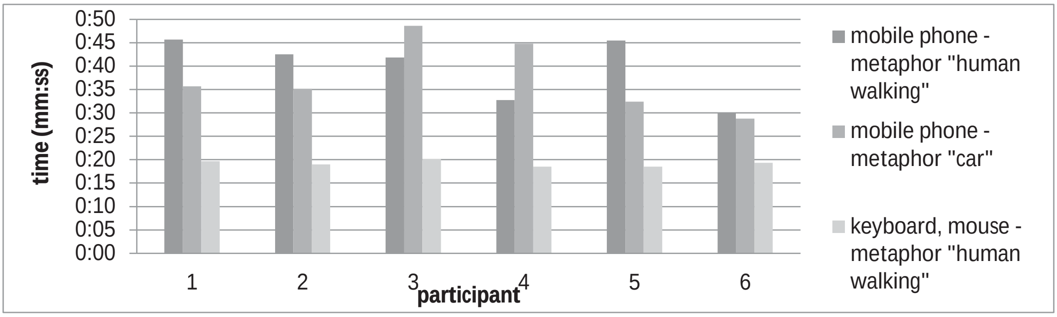


Figure 5. Average times of each tester using different control methods.

Then, we performed qualitative testing using INTUI questionnaire [9]. It consists of 16 questions that are focused on the impressions and feelings about our control. The values (range 1-7) were modified, so that the larger numbers mean better results. After averaging all the values of all testers for every control method we got: phone - car metaphor: 4.71, phone - human walking metaphor: 4.52, m&k: 5.28. We believe that the best results were achieved with the m&k because the users have got wide experience of using them from other applications. Nevertheless, in connection with certain questions, our control methods achieved higher values. Users considered our control methods easier to remember (mobile - 6.4, 6.4, m&k - 5.2), more fascinating (mobile - 5.2, 5.6, m&k - 3) and our control methods were considered a good experience (mobile - 5, 4.6, m&k - 2). On the other hand, our control methods required more attention and concentration (mobile - 2, 2.4, m&k - 5.4). We believe that this is the case because it is necessary to pay attention to the position of the device.

5 Conclusions

We focused on the possibilities of VR control using a mobile phone. There are many devices to control VR and new are still being developed, including the mobile phone which has various sensors. These are rarely used to control VR. Existing solutions mostly do not use all 6DOF, which we consider as a main disadvantage, since it is not portable. Therefore we devise our own solution with 6DOF based on three different metaphors. The helicopter metaphor was for users unknown, so in our test (6 testers) we compared car and walking metaphors with the m&k control. The qualitative test results showed the users consider our ways of control as something new, unusual and it was easier for them to learn it. Our quantitative test results are weighted in favour of the m&k control, but it is caused by testers' lack of experiences with the mobile control. Otherwise, our control would be equal, if not better.

We have already found some implementation weaknesses in our code. After we fixed some of them, we have tested it by one tester and we achieved results close to the results achieved with the m&k. Therefore, we believe that after fixing all weaknesses, we will obtain results where our metaphors will be equal to the m&k control. And to make it even more user friendly, we plan to make the control customizable, automatically adaptable and support calibration. These features could positively affect our control methods.

References

[1] Bauer, J. et al.: Using smart phones for large-display interaction, In: *International Conference on User Science and Engineering*, IEEE, (2011), pp 42-47.

[2] de Souza, M. et al.: Using Acceleration Data from Smartphones to Interact with 3D Medical Data, In: *Proc. of the 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, (2010), IEEE, pp. 339-345.

[3] Hürst, W., Helder, M.: Mobile 3D graphics and virtual reality interaction, In: *Proc. of the 8th International Conference on Advances in Computer Entertainment Technology*, ACM, (2011), pp. 28:1-28:8.

[4] Hyunju, C. et al.: Motion recognition with smart phone embedded 3-axis accelerometer sensor, In: *International Conference on Systems, Man, and Cybernetics*, IEEE, (2012), pp. 919-924.

[5] Kin, K. et al.: Eden: a professional multitouch tool for constructing virtual organic environments. In: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, (2011), pp. 1343-1352.

[6] Klompmaker, F. et al.: Towards Multimodal 3D Tabletop Interaction Using Sensor Equipped Mobile Devices, In: *Mobile Computing, Applications, and Services*, Springer Berlin Heidelberg, (2013), pp. 100-114.

[7] Patel, H. et al.: Human centred design of 3-D interaction devices to control virtual environments, In: *Int. J. Hum.-Comput. Stud.*, (2006), vol. 64, no. 3, pp. 207-220.

[8] Sachs, D.: Sensor Fusion on Android Devices: A Revolution in Motion Processing, Google Tech Talk, [Online; accessed March 23, 2013]. Available at: www.youtube.com/watch?v=C7JQ7Rpwn2k.

[9] Ullrich, D., Diefenbach, S.: INTUI. Exploring the Facets of Intuitive Interaction. In: *Mensch & Computer*, Oldenbourg Verlag, (2010), pp. 251-260.

Computer Networks, Computer Systems and Security

Performance Comparison of Selective TCP with Modern Variants of the TCP Protocol

Peter VRANEC*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
vranec11@fiit.stuba.sk

Abstract. A number of flow control driven approaches based on packet loss detection have been proposed and used in computer networks, but in modern high bandwidth and high distance networks these protocols may not be sufficient to effectively utilize a network capacity. Therefore, new flow control methods based on RTT measurements were presented. However, methods based purely on this approach have also serious limitations. So, we decided to compare the protocols based on the above two types of flow control. We compare some modern TCP variations like Compound TCP protocol designed by Microsoft, Westwood and CUBIC with the Selective TCP protocol, which is a modification of New Reno. We show and confirm by simulations the significant performance grow in case of Selective TCP.

1 Introduction

The TCP protocol was built for wired networks in 1980s to deal with network congestion problems compared to random losses which may occur in wireless networks, where the error rate is several magnitudes higher. Basically, TCP utilizes the network by changing size of a congestion window (*cwnd*) clocked by an arrival of acknowledgements which greatly depends on round trip times (RTT) of packets. So, whenever the RTT is large enough, and *cwnd* has already been decreased by packet losses, the TCP connection may not be able effectively utilize the network.

These criteria are often met in wireless networks and in networks called long fat networks (LFN). According to the RFC 1072 a network is considered an LFN when its bandwidth delay product (BDP) is considerably larger than 10^5 bits. Optimum network throughput can be achieved, when the sender can send an adequate amount of data, before being required to stop and wait for a response from the receiver. If that amount of data is less than the BDP, then the network bandwidth is not fully utilized. The default window size cannot go beyond 65535 bytes, so for example in a network with 20 Mbps bandwidth and RTT of 50 ms, the BDP is 20×1024 kbps × 50×10^{-3} s = 122 kB, which is greater than 65 kB maximum window size. That is why a new TCP tuning technique was proposed in RFC 1323, called the Window scaling factor, which can scale the window size up to 1 GB.

[*] Bachelor degree study programme in field: Computer and Communication Systems and Networks
Supervisor: Dr. Peter Trúchly, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 375–380.

2 Current state

In this section we provide a short introduction into the tested TCP variants with special focus on how these variants work with the congestion window. We decided to test selected modern and most used modifications of TCP such as Compound TCP [1], Westwood [2], CUBIC [4] compared to old TCP Reno, so that we can see benefits of the modern protocols. We concentrate and give a deeper review of Selective TCP [3], as a modification of New Reno [5] that uses timers to determine the cause of a loss. The special focus on Selective TCP lies in its simplicity to identify the type of packet loss simply based on the difference of propagation delays between wired and wireless links. Authors in [8] evaluated the performance of three TCP variants: Compound TCP, Cubic TCP and TCP New Reno and have shown that Cubic TCP outperforms other two variants in high speed wired networks but in wireless networks all three protocols are unable to maintain a high goodput. A detailed evaluation and comparison of Westwood+, New Reno and Vegas TCP congestion has been performed in [9] with results that Westwood+ provides goodput improvements ranging from 23% to 53% with respect to New Reno.

2.1 Compound TCP

Compound TCP (CTCP) is both loss based and delayed based modification, it means that it uses a packet loss as an indicator of congestion and also it monitors the RTT changes to deduce the congestion [1]. CTCP maintains a delay window and a congestion window, which behaves in the same way as the Reno's, to determine the send window which is the sum of these two windows. The delay's window behaviour greatly depends on RTT. If a delay is small the window increases rapidly, but when the delay starts to grow the window is reduced by the estimated queue size:

$$Diff = \left(\frac{win}{baseRTT} - \frac{win}{RTT}\right) * baseRTT \tag{1}$$

The *baseRTT* is an estimation of the transmission delay over the network path; RTT is the actual round trip time. The *win/baseRTT* component defines the expected amount of data to be sent to the network, while *win/RTT* defines the actual amount of data that is being transmitted. The *Diff* tells the amount of data that was sent to the network but does not pass the network bottleneck.

2.2 CUBIC

CUBIC is an enhanced version of the congestion control algorithm, binary increase congestion control (BIC) TCP protocol, which aims for enhancing its fairness to other TCP flows. The main specialty about CUBIC is that a growth of the window size is not dependent on the RTT, as we see in other TCP variants, but on real-time [4]. It means that the growth of the window size is independent of RTT and is determined by a cubic function driven by the elapsed time since the last loss event.

$$Wcubic = C(t-K)^3 + W_{\max} \tag{2}$$

$$K = \sqrt[3]{W_{max} * {}^{\beta}/_{C}} \tag{3}$$

C is a constant (scaling factor), *β* is a constant multiplication decrease factor, *Wmax* is a window size when the last loss occurred. *K* is a cube root of *Wmax* multiplied by a factor of multiplication. This function ensures good intra – protocol fairness and also RTT fairness, because the *t* time will be always the same as opposed to the different RTTs.

2.3 TCP Westwood

TCP Westwood (TCPW) is a sender side modification of TCP Reno, which modifies the congestion window algorithm. TCPW measures at the sender side a flow of returning ACKs, and from these measures it estimates the bandwidth, which is then used at a congestion event for the congestion window and slow start threshold new values selection. The estimation is based on amount of acknowledged data in an ACK segment and the time elapsed since the last ACK [2].

3 Selective TCP

Selective TCP is a modification of New Reno protocol, where the sender side implements also a timer based congestion detection called Inter-Arrival-gap [3]. Let's assume a topology in Figure 1, where T is data transmitter, RT is router, BS is base-station and R is the receiver.

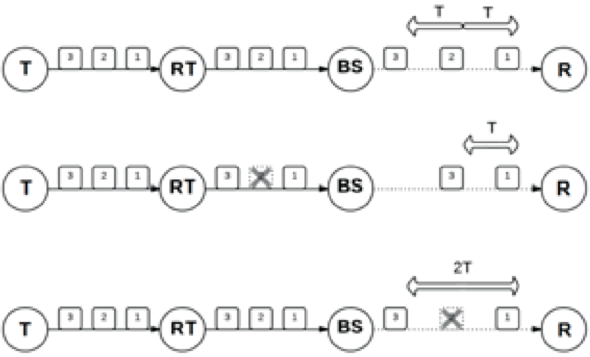


Figure 1. Network topology with three possible situations (cases).

In the first case there is no packet loss, so the receiver measures a *T* gap between consecutive packets 1 and 2. In the second case when the loss occurred due to buffer overflow in the base station (BS), the receiver also measures a *T* gap between packets 1 and 3, because we neglect the time required for packet processing in the base station. In the last case, the second packet was lost in wireless transmission, so the gap between two consecutive packets is *2T* that is when the receiver evaluates the packet as lost due to errors in the wireless link (see formula (4)).

$$(n+1)Tmin < Tg < (n+2)Tmin \tag{4}$$

Tg is the time difference between the last packet in a sequence and out of sequence packets, in our case the time between packets 1 and 3. *Tmin* is a minimal inter-arrival time gap between two consecutive packets without loss. Using the formula (4), the Selective TCP can determine the cause of the loss, to influence the sender's cwnd reduction algorithm. After a packet loss due to wireless link the receiver sends a SNACK (Selective Negative ACK) segment with the information about the lost segments. In the congestion event loss the receiver sends the newly calculated bandwidth (see formula (5)).

$$BW = \frac{n_p * s_p * 8}{(t_n - t_{n-1}) * 1000} kbps \tag{5}$$

t_n is the time of the last in-sequence packet, t_{n-1} is the time of the penultimate in-sequence packet, s_p is packet size in bytes, and n_p is a number of packets received in the given time interval.

4 Simulation scenario

In our simulations we measured the congestion window of each TCP modification and the actual throughput of each modification. We used the topology depicted in Figure 2. There are two sender nodes, first one generates four UDP CBR traffics (each 128 kbps), which are used for simulating a traffic at a constant bit rate, like video streams, and the second one transmits one TCP FTP (file transfer protocol) flow, that is commonly used for large data transfers upon TCP. All links have 1 ms propagation delay with 2 Mbps bandwidth except for the wireless links, between receivers and the base station, with 1 Mbps. In simulations we simulated the random wireless losses with two state Markov error model [6], with an packet error rate of 5 %. All simulations have been realized using network simulator v. 2 [7].

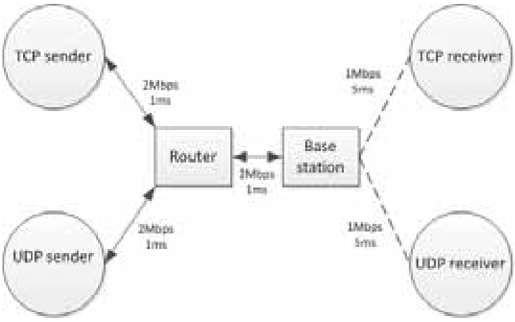


Figure 2. Simulation scenario topology.

5 Simulation results

The first tested parameter was *cwnd*. As we can see the Selective TCP‘s *cwnd* (see Figure 3) highly outperforms all other modifications. The reason why the congestion window is so high, is that in a congestion event the window is calculated using the formula (6), which is then sent to the receiver. The formula uses the delay between two consecutive packets and the actual *cwnd* value to calculate the new value, and that is why the new value cannot be multiplicatively decreased. The reason why there are sharp drops is that in some cases the sender behaves exactly as the original New Reno, for example in the 170 s the *cwnd* was set to the actual *ssthresh* value of 10, as the result of exiting the fast recovery, due to packet drops, which is the expected behaviour.

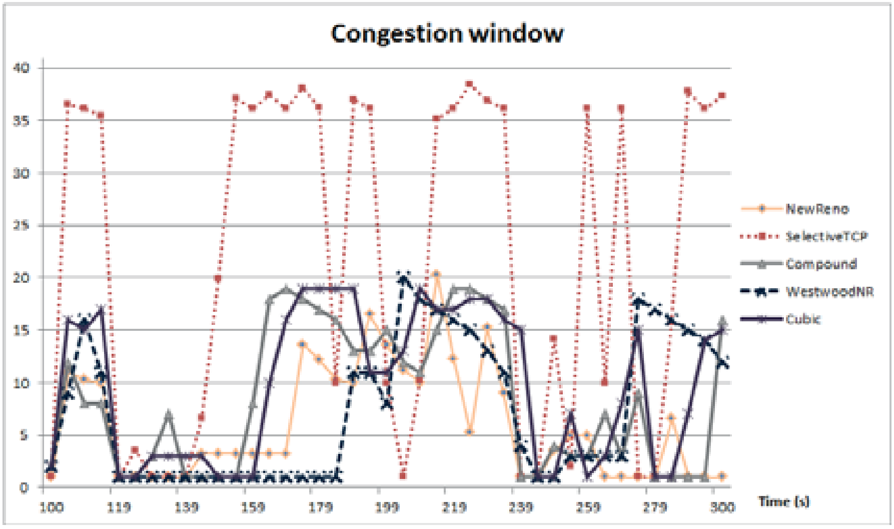


Figure 3. Congestion window versus simulation time.

We tested the overall performance (see Figure 4) in terms of maximum sequence numbers of packets received by the receiver over timer. We can see that Selective TCP shows the best performance as expected because of its highest *cwnd*, the second one is CUBIC, followed by Compound TCP and TCP Westwood and the last one New Reno. About 15 s earlier starts the Selective TCP increasing the *cwnd* as the CUBIC and Compound TCP, mainly due to a good bandwidth estimation after exiting the fast recovery at the receiver side. Afterwards all the modifications follow the linear increase till 239 s, where the window stops growing. By the end of the 269 s we see that the Westwood is the first one to start again growing the window.

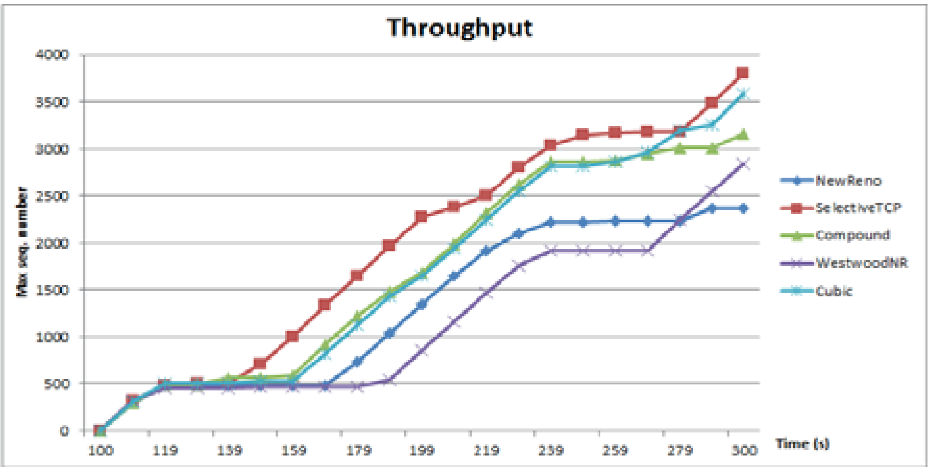


Figure 4. Throughput of various TCP modifications.

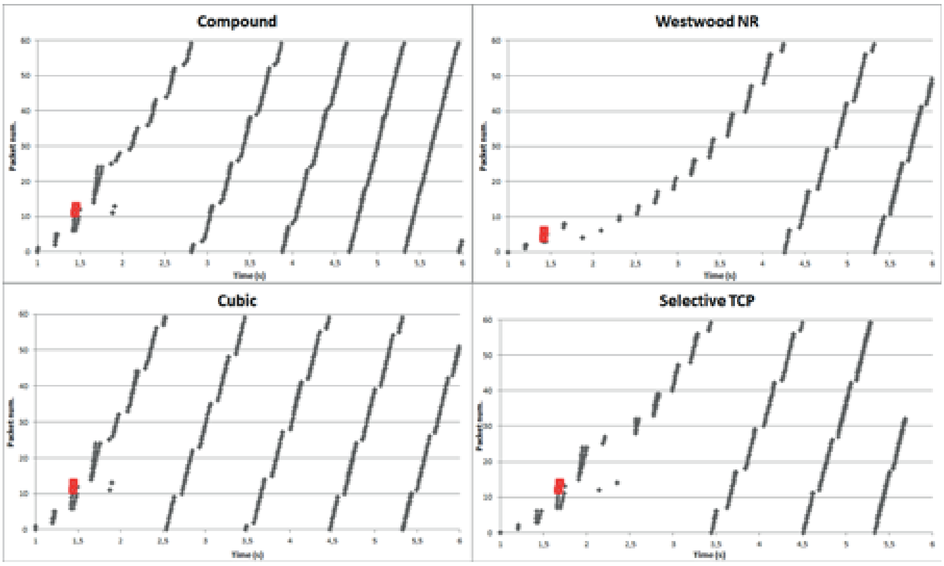


Figure 5. TCP Westwood, Selective TCP, Compound TCP, CUBIC.

We also tested the behavior of individual modifications when certain amount of packet drops occurred during congestion window. We can see (Figure 5) how each modification reacts to a packet loss shown as a packet flow in time. The best results are achieved by the Cubic TCP, as a result of a more aggressive *cwnd* increase algorithm (see formula (3)). The WestwoodNR shows expected linear increase of *cwnd* for every RTT in congestion avoidance phase, while the Selective TCP shows a typical behavior of a fast recovery, after which (time 2.5 s) we see a congestion avoidance phase with linear increase. The maximum overall number of Kbytes sent is by the Compound 620 Kbytes, 597 Kbytes for the CUBIC, and 335 Kbytes for WestwoodNR.

6 Conclusion

This paper concentrated on the simulation based performance comparison of modern TCP modifications (Compound TCP, CUBIC, TCP Westwood, TCP New Reno) with the focus on a promising Selective TCP modification. The Selective TCP has the best performance in throughput as it maintained the highest *cwnd* through the simulation. However, the overall best TCP modification among the tested ones is the Compound, mainly because of its unique *cwnd* size control. The results show an unexpected good performance of Selective TCP, which is interesting, and requires some more tests to confirm them, which may have been just a special circumstance in the topology, that primarily focused on random errors in wireless links. Also the fact that speeds in the topology where quite low as opposed to the current networks, what may not reveal the real potential of these modern protocols built for high speed networks. Also the modifications (as TCP Westwood) based purely on RTTs may have been underutilized because of the low latencies used in the topology. However, our objective was to analyse basic characteristics and performance of most used TCP protocols in a simple network configuration.

Acknowledgement: This work was partially supported by the Grant no. 1/0676/12 of the Slovak VEGA Grant Agency.

References

[1] Tan, K., Song, J., Zhang, Q., Sridharan, M.: A Compound TCP Approach for High-speed and Long Distance Networks. In: *IEEE Infocom*, (2006), Spain.

[2] Casetti, C., Gerla, M., Mascolo, S., Sanadidi,, M.Y., Wang, R.: TCP Westwood: end-to-end congestion control for wired/wireless networks. *Wireless Networks (2002)*, vol. 8, no. 5, pp. 467-479.

[3] Paul, R., Trajkovic, L.: Selective-TCP for Wired/Wireless Networks. In: *Proceedings of the 2006 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (IEEE SPECTS)*, (2006), pp. 339-346.

[4] Ha, S., Rhee, I., Xu, L.: Cubic: A new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review*, (2008), vol. 42, no. 5, pp. 64-74.

[5] Floyd, S., Henderson, T.: *The New Reno modification to TCP's fast recovery algorithm*. IETF RFC 2582, Apr. 1999.

[6] Gurtov, A., Floyd, S.: Modeling wireless links for transport protocols. *ACM Computer Communication Review*, (2004), vol. 34, no. 2, pp. 85-96.

[7] The Network Simulator - ns-2. [Online accessed March, 2014]. Available at: http://www.isi.edu/nsnam/ns.

[8] Abdeljaouad, I., Rachidi, H., Fernandes, S., Karmouch, A.: Performance Analysis of Modern TCP Variants: A Comparison of Cubic, Compound and New Reno. In: *25th Biennial Symposium on Communications*, (2010), Kingston, ON, pp. 80-83.

[9] Macura, A., Missoni, E., Kordic, Z.: Comparison of Westwood, New Reno and Vegas TCP Congestion Control. In: *Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium*, (2012), vol. 23, no. 1, Vienna, Austria, ISSN 2304-1382.

GPRS Modem Emulator

Miroslav BABJÁK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xbabjak@fiit.stuba.sk

Abstract. This paper provides an overview of OsmocomBB project and design of GPRS modem emulator application, which should provide most of the basic functions of Mobile Station in GSM mobile network. We show the architecture and communication protocol used in OsmocomBB project. We also looks at how will this application communicate with BTS and what we need to add GPRS support. In the time when this paper was written was created only the prototype of this application.

1 Introduction

Even if the GSM is the most widely used network there are missing complex tools for GSM/GPRS network simulation and experimenting. The Osmocom group created many applications for GSM networks which allow create a small GSM network. But for do this you have to buy some hardware like sysmoBTS and compatible hardware phone.

We want create a GSM network without using any special hardware by using only PC. Every node in GPRS core architecture can be replaced with existing applications only BTS and MS need special hardware for communication witch each other, because they communicate via air interface.

We will create a MS application without any hardware. Based on analysis I will chose the OsmocomBB project which implements almost all function of MS even if it need special hardware phone. Also we have to add a GPRS support which is missing in this application.

We will modify the IMSI attach procedure to attach also to GPRS (GPRS/IMSI combined attach). Also we will add procedures for handling the PDP context like PDP context activation and deactivation. In the PDP context activation request is information about QoS (Quality of service) and APN (Access Point Name). If the PDP context activation is successful the GGSN assigns the PDP address (IP address) to MS which sent a request.

After the PDP context is activated and MS has the PDP address, it can send the data through PDTCH (Packet Data Traffic Channel). Data is send in RLC/MAC blocks which contains MAC header, RLC header and RLC data. Besides data also it sends the RLC/MAC control blocks which contain the control messages. These messages are send through PACCH (Packet Associated Control Channel).

[*] Master degree study programme in field: Computer Engineering
Supervisor: Martin Nagy, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 381–386.

2 OsmocomBB

OsmocomBB (Open Source Mobile Communication Baseband) is open source project developed by Osmocom group written in C language. They developed other open source projects focused on GSM like OpenBSC (Base Station Controller application).

The OsmocomBB implements the phone-side GSM protocol stack from layer 1 to layer 3. As radio interface is using the compatible phone (Motorola C115/C117/C118/C121/C123) which is connected via serial cable to PC.

Part of the OsmocomBB project is shared library and few applications which can be divided to three groups:

- Baseband firmware
- Firmware management software
- GSM layer 2/3 applications

2.1 Baseband firmware

Baseband firmware is running on the phone and provides decoding information from radio waves to format which application can understand. Some interesting application running on mobile:

- *Layer1* – This application is GSM layer 1 proxy, which allows you run full GSM implementation on PC. It communicates with higher layer through prepared interface.
- *Menu* – Allows choose the application stored in Flash memory of the phone.
- *Rssi* – Can monitor the RSSI (Received Signal Strength Indicator)
- *EMI (Electro Magnetic Interference)* - Can be used for generation the RF (Radio Frequency) interference

2.2 Firmware management software

Applications listed below are still part of firmware, but they are running on PC. They provide support for firmware running on phone like upload firmware to the phone flash memory. These applications are:

- *Osmocon* – This application is interface between Layer1 application running on phone and layer2/3 application running on PC. It communicates with phone via serial cable and with layer2/3 via UNIX socket.
- *Osmoload* – This is used for work with phone flash memory. It allows write, dump examine the flash memory.
- *Calypso pll* – This can be used for calculate Calypso DPLL (Digital Phase-Locked Loops) multiplier + divider
- *Rita pll* – Same as Calypso pll, but for Rita DPLL.

2.3 GSM layer 2/3 applications

These applications provide functionality of GSM layer 3 with layer 2 (LAPDm). They running on PC and communicate with layer 1 through UNIX socket.

- *Mobile* – This application implements most of GSM phone functions like voice calls and SMS transmission and reception.
- *Cell_log* – This can scan the carrier frequencies in the area and obtain information about them.

- *CCCH_scan* – Obtain information about power measurement and CCCH channel like paging.
- *BCCH_scan* – This dump information from BCCH channel like system information.
- *CBCH_sniff* – This can provide information about cell from broadcast channel like GPS location of cell.

2.4 OsmocomBB software stack

OsmocomBB application need to process radio signals from antenna and create appropriate structures for upper layers. This processing is done in phone using the modified firmware.

When the phone receives a RF (Radio Frequency) signal he passes it to Rita mixer. Rita mixer is hardware chip developed by Texas Instrument which does conversion into analog I/Q baseband. The analog I/Q baseband are sending to Iota ABB, which is Analog to Digital Converter (ADC) for GSM baseband signals. After that is digital baseband signal send to BSP (Baseband Serial Port) of the Calypso DBB (Digital Base Band). Calypso DBB is popular DBB for simple phones without some extra functionality.

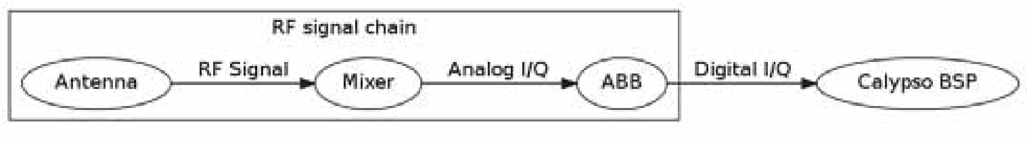


Fig. 1. Radio signal processing.

In the Calypso DBB is signal send from BSP to DSP (Digital Signal Processing) where is signal process, demodulated, decoded etc. and then is pass to ARM (Advanced RISC Machine) processor. In the ARM core is OsmocomBB layer1 application which processes MAC block from DSP and transfers them to L1CTL protocol and send this to UART (Universal Asynchronous Receiver/Transceiver).

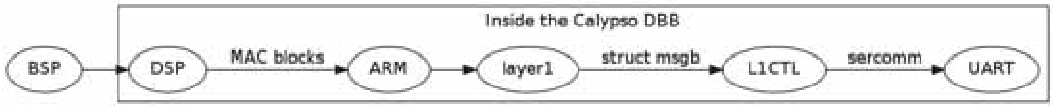


Fig. 2. Signal processing in Calypso DBB.

After the data are send to UART they are available on serial interface on PC. The Osmocon application which is running on PC provides forwarding L1CTL messages to Layer 2/3 application via UNIX domain socket.

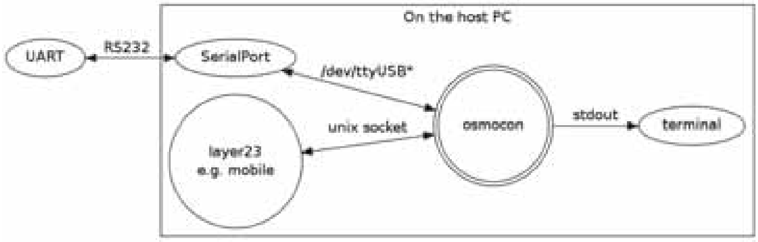


Fig. 3. Data flow from serial port to layer 2/3.

2.5 L1CTL

L1CTL is protocol defined by OsmocomBB for communication between layer1 and higher layer. The layer1 application creates L1CTL messages based on data on radio interface. This messages are send to layer 2/3 applications through osmocon. L1CTL defines 30 types of messages like:

- L1CTL_CCCH_MODE_REQ
- L1CTL_TCH_MODE_REQ
- L1CTL_DATA_REQ
- Etc.

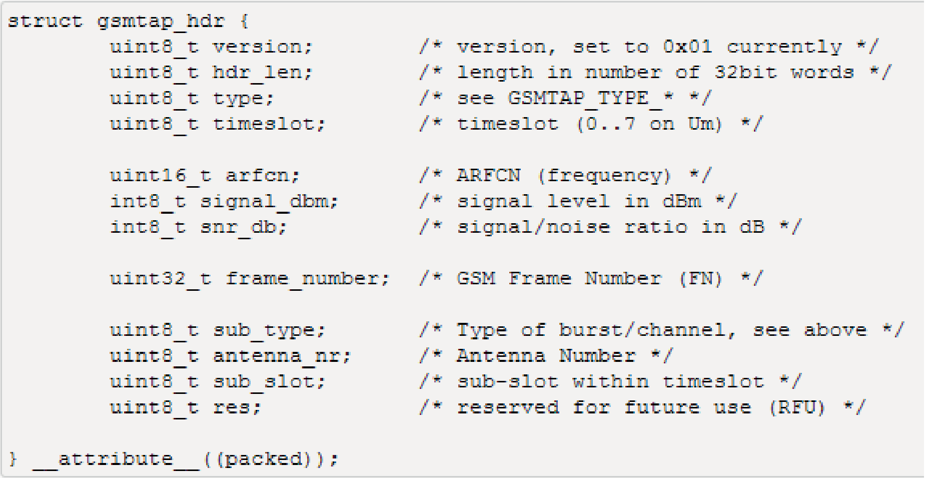
Each message type has appropriate data structure for relevant data based on message type.

3 Design of GPRS modem emulator

GPRS modem emulator can be divided into two parts. At the first we have to remove hardware phone and replace it with some application, which will provide basically the same functionality. And the second part is to add support for the GPRS to the mobile application in OsmocomBB project.

3.1 Communication with BTS

At first we have to create an application, which will be proxy server between mobile application from OsmocomBB and the BTS. It means that it will be radio interface between MS and BTS, where layer 1 application will receive messages in L1CTL format from mobile, parse it, create data structure which can understand the BTS and send it to BTS. Communication between BTS and layer1 will be through socket. It can be the UNIX socket or IP socket (UDP or TCP) it depends on BTS design. Message structure also depends on BTS design and it can be in any format.



```
struct gsmtap_hdr {
        uint8_t version;        /* version, set to 0x01 currently */
        uint8_t hdr_len;        /* length in number of 32bit words */
        uint8_t type;           /* see GSMTAP_TYPE_* */
        uint8_t timeslot;       /* timeslot (0..7 on Um) */

        uint16_t arfcn;         /* ARFCN (frequency) */
        int8_t signal_dbm;      /* signal level in dBm */
        int8_t snr_db;          /* signal/noise ratio in dB */

        uint32_t frame_number;  /* GSM Frame Number (FN) */

        uint8_t sub_type;       /* Type of burst/channel, see above */
        uint8_t antenna_nr;     /* Antenna Number */
        uint8_t sub_slot;       /* sub-slot within timeslot */
        uint8_t res;            /* reserved for future use (RFU) */

} __attribute__((packed));
```

Fig. 4. GSMTAP header in OsmosomBB project.

If we use the UNIX socket the easiest way is keep the same message structure as it use in mobile application, which is L1CTL. But if we use the IP socket the best structure will be GSMTAP.

GSMTAP is a pseudo-header for transmit the GSM frames from air interface (Um) through IP networks. In front of GSM protocol message is added the header of GSMTAP protocol (Fig. 5). It's inspired by Radiotap which do something similar for 802.11 (Wi-Fi) networks.

Layer 1 will be standalone application which will create the UNIX socket for layer2/3 applications and connect to BTS socket.

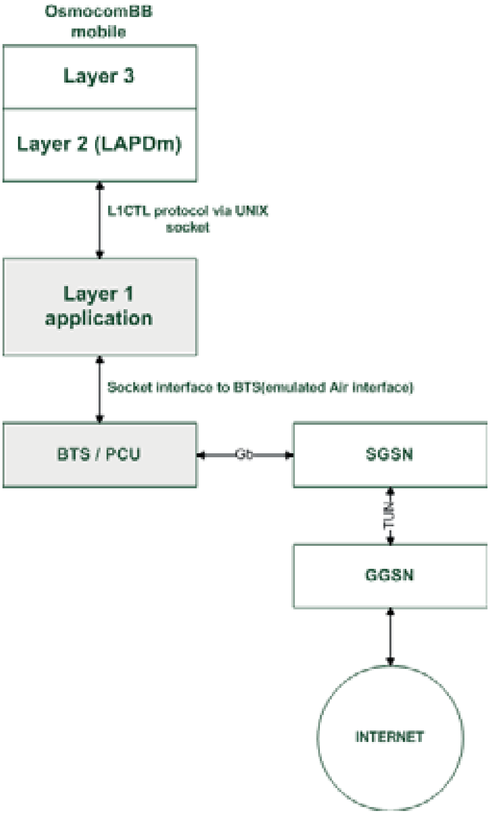


Fig. 5. Block schema of layer 1 application.

3.2 GPRS support

Because the OsmocomBB doesn't support the GPRS we have to add new structures and function for handle the GPRS messages like:

- PDP Context Activation
- PDP Context Deactivation
- Sending GPRS data and signalling

Also we need to modify structures where the information about MS are stored, because we need add information about created PDP context like PDP address, TLLI (Temporary Logical Link Identifier) number, etc. It's possible that the L1CTL protocol will be modified for support more messages needed for GPRS.

For discovery what exactly is sending and in what format we can use the osmoBTS, which allow us debug messages sending though air interface between MS and osmoBTS.

4 Conclusion

Application like this can be used for research in mobile networks and for academics purposes, because we can capture all communication between relevant nodes in packet sniffers like Wireshark.

In this paper we introduce OsmocomBB project, his architecture and communication protocol. Also we show design of GPRS modem emulator application based on this project with some improvements like GPRS support and no special hardware requirement.

In this time is created only the prototype of this application and we still working on this project. The prototype application is able to communicate with upper layer like mobile application and allow complete the initialization procedure in mobile application where is reset the hardware, check the SIM card and so on. The prototype doesn't support all the L1CTL messages yet, but it has prepared the communication interface for both sides (BTS and mobile application).

Acknowledgement: This work was partially supported by Slovak National research grant 1/0676/12 and by Tatra banka foundation under the contract No. 2012et011.

References

[1] Schiller, Jochen: Mobile Communications, London: Person Education Limited, 2003, ISBN 0-321-12381-6.

[2] OsmocomBB [online], [cit. 2013-12-04], Available from: http://bb.osmocom.org/

[3] Osmocom: OsmoBTS, [online], [cit. 2013-12-09]. Available from: http://openbsc.osmocom.org/trac/wiki/OsmoBTS

[4] SEURRE, Emmanuel: GPRS for mobile internet, Boston: Artech House, 2003, 419 s. ISBN 15-805-3600-X.

[5] 3GPP: TS 23.002 rel. 12.2.0 – Network Architecture, 2013.

Optimization of Data Flow in Service Provider Networks

Ivana HUCKOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
huckova.ivana@gmail.com

Abstract. Network resources in terms of available bandwidth are not always sufficient to provide desired quality of service, especially for real-time traffic. Network performance parameters such as delay, jitter and packet loss are significant indicator of suitability of the path for this type of traffic. To provide QoS traffic engineering methods are used. In this paper we focus on traffic engineering in MPLS networks and the question of quality of service, since it is the main reason of deploying TE. We propose an online TE server to optimize the data flow in the network and maximize the utilization of the network resources.

1 Introduction

Customer traffic often suffers from congestion due to the bottlenecks in the network which leads to degradation of service's quality. Traffic engineering (TE) as a way of efficient resource optimization is being deployed to address this problem. By balancing the traffic load distribution in the network and minimizing bandwidth consumption, traffic engineering provides the maximization of network's utilization [9].

Besides the network utilization, TE also deals with the question of quality of service (QoS). Many applications require certain QoS guarantees, such as end-to-end delay, jitter or loss probability. These requirements need to be addressed by TE mechanisms in order to provide satisfying services to customers [1].

TE routing approaches can be classified into various categories based on different aspects: IP-based and MPLS-based TE, online and offline TE, interdomain and intradomain TE, unicast and multicast TE [9]. This work is focused on MPLS-based TE with regard to QoS.

2 Related work

There were many different approaches, mechanisms and solutions proposed and developed in the area of MPLS traffic engineering so far. Two main aspects of MPLS TE can be defined as 1)

[*] Master degree study programme in field: Computer Engineering
Supervisor: Dr. Martin Hrubý, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 387–392.

finding of suitable path in the network and the creation of Label Switched Paths (LSPs) and 2) distribute the traffic among the LSPs to maximize network utilization.

Many studies and research has been done in the question of routing the LSPs [3],[7],[9]. The main advantage of these algorithms is their simplicity. They can, however create bottlenecks and lead to network under-utilization [2].

Relatively little research has been done in the area of distributing the traffic among created LSPs. Authors in [8] proposed a method based on traffic flow distribution and splitting for traffic engineering in MPLS networks. MPLS Adaptive Traffic Engineering (MATE) proposed in [5] represents another algorithm focused on distributing the traffic over multiple LSPs. Our work is focused on the traffic distribution and network utilization since this area is not being as analyzed as the routing of LSPs.

3 Proposed solution

In our work we propose an online TE server to provide traffic distribution, sufficient QoS demands for real-time traffic and maximize network utilization. The proposed server will use LSPs created in advance (manually or using some of analyzed algorithms) and it will steer the traffic across the network using these LSPs. The operation of the server is divided into several steps:

1. The analysis of the network and existing LSPs
2. The end-to-end measurement of network performance parameters of LSPs
3. Calculation of the cost of each LSP
4. The assignment of the traffic to LSPs
5. Optimization (if necessary)

The connection of the server to the network will be provided as shown in Figure 1. The server will communicate with the Provider Edge (PE) router via SSH and SNMP connection. SSH connection will be used mainly to obtain specific router configuration details and to apply changes in configuration. SNMP will be used to gather measured parameters via IP SLA probes [4].

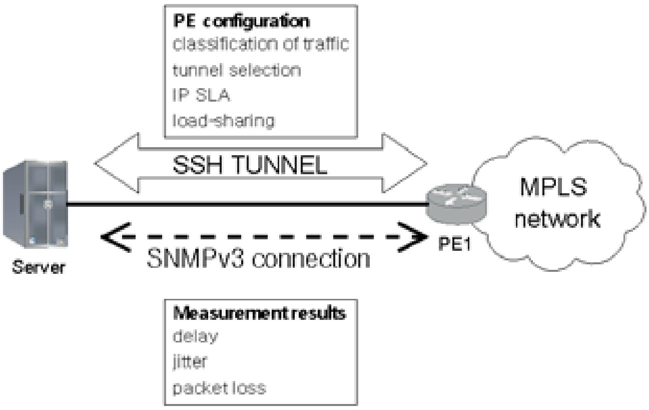


Figure 1. Server connection to the network.

Traffic entering the network will be classified into four classes defined by the requirements it has. One class (Class1) will be dedicated for real-time traffic with strict default requirements to achieve sufficient QoS. All other classes (Class2, Class3 and Class4) will be used for non-real-time data traffic and the requirements of each class will be defined only by specific bandwidth guarantees. Traffic will be treated according to the class it belongs to in descending order. Each class will have defined the amount of overall bandwidth it can use in the network to avoid the traffic-class starvation.

It is important to periodically measure various network performance parameters, such as end-to-end delay, jitter or packet loss to provide up-to-date information about each LSP. The measurements will be carried out using IP SLA probes on the edge routers. Since the values of delay, jitter and packet loss are variable in time, it is preferable to work with their statistical values to provide trustworthy values of these parameters to be used. The measured values will be stored in the system's database for further usage.

The cost of LSP is used to decide whether the LSP is suitable for specific traffic class and therefore it has to reflect the parameters for every traffic class. The main difference is between Class1 and other classes since the first class has specific demands on values of delay, jitter and packet loss along with the bandwidth demand. There is no possibility of including network performance parameters of the link into the cost used for path computation at the time of writing this work. Although there is an effort to develop extensions for including the network performance criteria into OSPF, it is not usable at the moment [6]. Due to this fact we decided to use two cost values for each LSP – one as characteristic of network performance parameters and one to describe the bandwidth usage of LSP. The basic mathematical representation of cost value for Class1 is shown in Formula 1 where C_{voice} represents the actual value of the cost of LSP for Class1 traffic, C_{delay} represents the actual value of the cost of LSP according to the delay, C_{jitter} represents the actual value of the cost of LSP according to jitter, C_{loss} represents the actual value of the cost of LSP according to packet loss. The representation of cost value for classes Class2, Class 3 and Class4 is shown in Formula 2 where C_{data} represents the actual value of the cost of LSP for Class2, Class3 and Class4 traffic and free_bw_of_LSP represents the amount of unused bandwidth of LSP. The variables of C_{voice} and C_{data} are considered to be non-dimensional. The main advantage of this approach is the simplicity and computational modesty.

$$C_{voice} = \mathrm{C_{delay}} + \mathrm{C_{jitter}} + \mathrm{C_{loss}} \tag{1}$$

$$C_{data} = \text{free\_bw\_of\_LSP} \tag{2}$$

The C-values for delay, jitter and packet loss ($C_{delay,}$ $C_{jitter,}$ C_{loss}) will be obtained from a reference tables shown in Table 1, Table 2 and Table 3 respectively. Reference tables were proposed to accurately express the quality requirements defined in [10]. It is crucial to have all three parameters (delay, jitter, packet loss) in a specific range to be able to guarantee specific QoS. The proposed reference tables are created in such a way, that even one parameter out of range changes the LSP's cost significantly. No other information is then needed to select the suitable LSP.

The final ranges of the cost values calculated by Formula 1 for Class1 traffic are defined in Table 4. The cost in range from 0 to 3 represents the optimal conditions for real-time traffic. The cost in range from 3.1 to 12 represents that the conditions on the specific LSP are still within a suitable range according to [4]. Values of cost above 12.1 mean that the quality parameters of LSP are not sufficient to provide required QoS with values above 40 representing unusable LSP for real-time traffic.

Table 1. Reference table for values of delay.

delay [ms]	0	5	10	15	20	25	30	35	40	45	50
C_delay	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
delay [ms]	55	60	65	70	75	80	85	90	95	100	105
C_delay	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.0	3.10
delay [ms]	110	115	120	125	130	135	140	145	150	160	170
C_delay	3.15	3.20	3.25	3.30	3.35	3.50	3.65	3.75	4.00	12.10	12.15
delay [ms]	180	190	200	210	220	230	240	250	260	270	more
C_delay	12.20	12.30	12.40	12.50	12.60	12.70	12.80	13.00	40.00	40.00	40.00

Table 2. Reference table for values of jitter.

jitter [ms]	0	1	2	3	4	5	6	7	8	9	10
C_jitter	0.00	0.02	0.04	0.06	0.09	0.12	0.15	0.18	0.22	0.26	0.30
jitter [ms]	11	12	13	14	15	16	17	18	19	20	21
C_jitter	0.35	0.40	0.45	0.50	0.60	0.65	0.70	0.80	0.90	1.0	3.10
jitter [ms]	22	23	24	25	26	27	28	29	30	31	32
C_jitter	3.15	3.20	3.25	3.30	3.40	3.50	3.65	3.80	4.00	12.10	12.15
jitter [ms]	33	34	35	36	37	38	39	40	41	42	more
C_jitter	12.20	12.30	12.40	12.50	12.60	12.70	12.80	13.00	40.00	40.00	40.00

Table 3. Reference table for values of packet loss.

loss [%]	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
C_loss	0.00	0.02	0.04	0.06	0.10	0.14	0.18	0.22	0.25	0.28	0.31
loss [%]	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.0	1.05
C_loss	0.34	0.38	0.43	0.50	0.55	0.60	0.65	0.75	0.85	1.0	3.10
loss [%]	1.10	1.15	1.20	1.25	1.30	1.35	1.40	1.45	1.50	1.60	1.70
C_loss	3.15	3.20	3.25	3.30	3.35	3.40	3.55	3.70	4.00	12.10	12.15
loss [%]	1.80	1.90	2.00	2.10	2.20	2.30	2.40	2.50	2.60	2.70	more
C_loss	12.20	12.30	12.40	12.50	12.60	12.70	12.80	13.00	40.00	40.00	40.00

Table 4. Final values of cost.

Conditions	Optimal	Good	Bad	Unusable
C_voice	0 - 3	3.1 - 12	12.1 - 39	40 and more

The incoming traffic has to be served according to its traffic class. That means that if more traffic flows arrive in one time, they will be assigned to LSPs based on their priority. It is important to emphasize that different approach is used for voice and data traffic. Every traffic class has defined the maximum guaranteed bandwidth in the network. With the use of optimal distribution of traffic in the network however, more traffic can be served and use the network resources. In this case it is crucial to ensure that all traffic within the guaranteed bandwidth is treated in preference of the traffic beyond the guarantees. In other words, traffic from Class4 which is within the guaranteed bandwidth is of higher importance then traffic from Class2 which is beyond the guaranteed bandwidth.

The process of optimization may be considered as the most important part of the whole system. Its purpose is simple – to achieve efficient distribution of traffic across all LSPs with preserved QoS. It can be triggered by three events: 1) traffic trunk cannot be assigned to any LSP; 2) the network performance parameters of LSP carrying Class1 traffic have become insufficient; 3) LSPs are unevenly utilized.

4 Implementation

The whole system will be implemented as a server application running on a host connected to one of the PE routers in the network. To achieve effectiveness, optimal performance and scalability of our solution it is reasonable to use a modular scheme as an implementation method as shown in Figure 3.

The central part of the server will be the daemon which will be used to start up the server's performance by starting all the other components. The network analyzer will use information stored by the daemon to connect to the PE router by a secure SSH connection using the SSH client module. It will use the router's configuration to get information about the network and configured

LSPs. The main responsibility of the trunk handler is to apply the algorithms and manage all traffic demands in the network. The measurement engine will be used to manage the SNMP connection to the router and collect the results of measurements in the network. The main work of the calculator will be to calculate the cost of each LSP based on mathematical formulas (1) and (2). In the database all required information about LSPs, measurements, costs of LSPs, IP SLA configuration and assigned traffic flows will be stored.

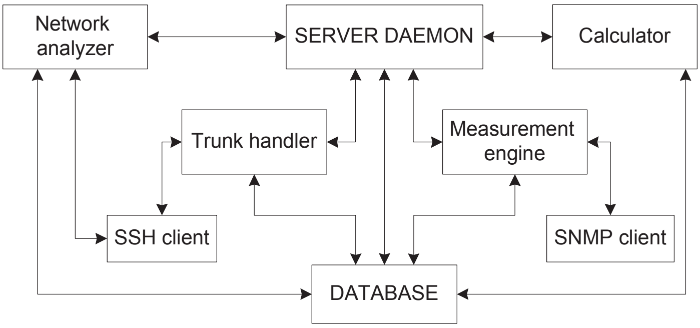


Figure 2. Architecture of the server.

5 Experiments

The proposed server will be implemented and experimentally tested in a laboratory environment. We proposed two testing topologies with the layout of LSPs defined in advance. One of the topologies is shown in Figure 3. The traffic in the network will be simulated by a traffic generator. The evaluation of the server's functionality will be based on provided QoS for all traffic classes. The utilization of the network resources will also be an important factor in evaluating the results of experiments.

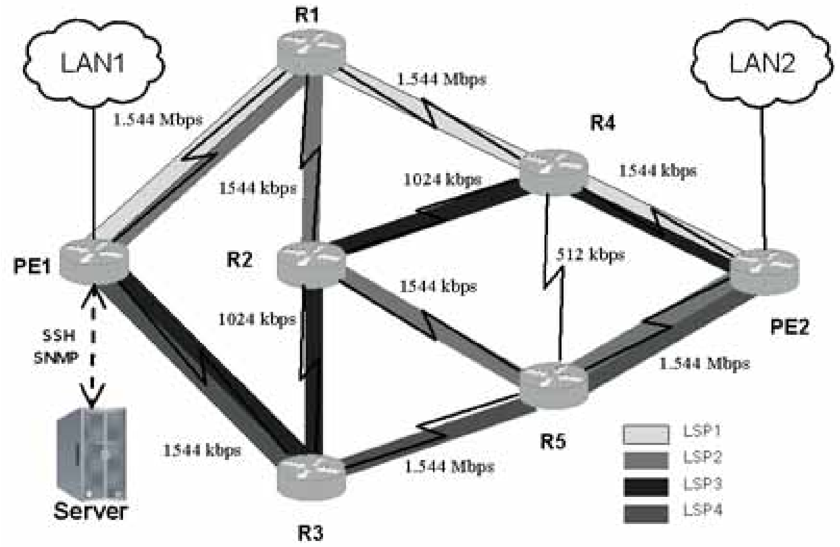


Figure 3. Testing topology.

6 Conclusion

A new online TE server for distribution of traffic with regard to QoS in MPLS networks is proposed in this paper. The selection of suitable LSP for a specific traffic class is based on proposed calculations of cost for each LSP. Periodic measurements of network performance parameters provide up-to-date information about LSPs which is used in the process of optimization. Utilization of network resources and therefore the amount of satisfied traffic flows (with satisfied QoS requirements) is expected to be higher than with the use of other algorithms (WSP or SWP) used to provide MPLS TE. However, further experiments are yet to be performed.

Acknowledgement: This work was partially supported by Grant No. 1/0676/12 of the Slovak VEGA Grant Agency.

References

[1] *Advanced Topics in MPLS-TE Deplyment*, Cisco Systems, White Paper, 2009, 30 p.

[2] Alidadi, A. et al.: *A New Low-Complexity QoS Routing Algorithm for MPLS Traffic Engineering*, 9th Malaysia International Conference on Communications, 2009

[3] Boutaba, R. et al.: *DORA: Efficient Routing for MPLS Traffic Engineering*, Journal of Network and Systems Management, vol. 10, no. 3, September 2002

[4] *Cisco IOS IP SLAs Configuration Guide*, Release 12.4, Cisco Systems, Inc. 2008, 271 p.

[5] Elwalid, A., et. al.: *MATE: MPLPS Adaptive Traffic Engineering*, IEEE INFOCOM, 2001

[6] Giacalone, S., et. al.: *OSPF Traffic Engineering Metric Extensions*, Internet draft, June 2013

[7] Kodialam, M., Lakshman T. V.: *Minimum interference routing with applications to MPLS traffic engineering*, IEEE INFOCOM, 2000

[8] Shi, T., Mohan, G.: *An efficient traffic engineering approach based on flow distribution and splitting in MPLS networks*, Computer Communications 29, Elsevier, 2006

[9] Wang, N., Ho, K. H., Pavlou, G., Howarth, M.: *An Overview of Routing Optimization for Internet Traffic Engineering*, IEEE Communication Surveys. In: The Electronic Magazine of Original Peer-Reviewed Survey Articles, 2008, 21 p.

[10] Szigeti, T., Hattingh Ch.: *End-to-End QoS Network Design*, Cisco Press, 2004, 768 p.

Faster Synthesis of Combinational Logic Based on Multiplexer Trees and Binary Decision Diagrams

Lukáš KOHÚTKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
kohutka555@gmail.com

Abstract. Multiplexers are well known as a basic building element of digital and mixed signal circuits thanks to their ability to perform any Boolean function. Optimization is a significant part of synthesis of combinational logic, since performance has to be improved, area and power consumption have to be reduced. The paper presents a novel faster optimization method for multiplexer trees using basic BDD reduction methods, residual variables, a hash table and top-down approach. An option to automatically replace some multiplexers in the multiplexer tree with basic logic gates has been added in order to achieve better results. This method also works with multiple Boolean functions at once so that we can design circuits with more than one output. Experimental results show that implemented algorithm reduces total amount of multiplexers in optimized multiplexer tree by up to 99,99% in comparison to non-optimized multiplexer tree. In addition up to 63,46% of multiplexers can be replaced with a logic gate OR, AND or XOR, which can reduce total amount of transistors needed to realise given combinational logic by up to 24,23%.

1 Introduction

Multiplexers are common building block for data-paths and they typically account for over 25% of chip size [1]. Functional blocks, as blocks which contain also multiplexers, can be easily disconnected from power supply based on designed power architecture integrated in a system-level model [2]. Multiplexers are also used in combinational circuit redesign for increasing quality of time specification testing and therefore they have to be involved there in optimal overhead or another logic gates have to be used [3]. Multiplexers are also used in self-testing or self-repairing memories [4] where it is important to minimize their usability in controlling test and functional modes and also power consumption. However, it is necessary to design these blocks as effective as possible. In multiplexer circuits it is possible usage of dynamic propagation path control [5] to decrease its dynamic power consumption. To reduce size on a chip and static power consumption

* Master degree study programme in field: Computer Engineering
Supervisor: Peter Pištek, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

of multiplexer tree, the similarity between multiplexer tree structures (built from 2-to-1 multiplexers) and Binary Decision Diagrams [6-7] (further BDD) is used. Therefore, BDD optimization methods can be used to optimize multiplexer tree in combination with other optimization methods. This method combines basic BDD optimization methods [8] with replacing some multiplexers by input signals (known as residual variables) and optionally by basic logic gates. Hash table and top-down approach speed up this method. Our research has been focused on minimization of amount of logic components required to create desired circuit and minimization of time required to design this circuit.

The organization of this paper is following. Section 2 contains theory of BDDs and basic BDD optimization methods. In Section 3 methods improving performance of proposed method are described. Section 4 describes rules for replacing multiplexers by basic logic gates or even residual variables. Section 5 shows experimental results achieved by this method. The last section concludes the paper and summarises benefits of this method.

2 BDD optimization

BDD optimization methods can be divided into two main categories [9]:

1. BDD ordering – results in Ordered Binary Decision Diagram (OBDD), which respects a given order of input variables. Variable ordering can have great impact on effectiveness of BDD reduction.
2. BDD reduction – when applied on OBDD, results in Reduced OBDD (ROBDD) which has lower number of nodes than not reduced BDD. ROBDD respects these two rules:
    a. Uniqueness – no two distinct nodes *u* and *v* represent the same variable and have the same left and right successor, i.e.:

       var(u) = var(v), left(u) = left(v), right(u) = right(v) what implies *u = v*

    b. Non-redundancy – no variable node u has identical left and right successor, i.e.:

       left(u) = right(u)

The proposed method uses BDD reduction described above. BDD ordering is a complex problem because the total number of all possible orderings of input variables is equal to factorial of input variables number. This problem was not aim of the research.

3 Top-down approach and hash tables

The performance of whole synthesis process can be improved by two improvements: top-down approach and hash tables.

Top-down approach means that the creation of BDD is performed in order from top (beginning from root) down to successors. During this creation, both reduction rules “uniqueness” and “non-redundancy” are checked at each BDD node. When one of those reduction rules should apply and reduce the node, this node is not even created. This means that we don’t need to pass by all paths of BDD because some of the paths are closed sooner. The result of this approach is fewer amount of nodes needed to check with reduction rules.

The second improvement uses a hash table to greatly improve the performance of reduction rule “uniqueness”. After creation of each node a hash index from this node by a hash function is generated. This hash function generates hash index using XOR on every k bits of a node’s vector so the resulting hash index is the k-bit number and the hash table has size of 2^k pointers, each representing one linked list of pointers on BDD node. This means that every created node (its pointer) is added to the linked list at index defined by hash index. The performance improvement is achieved by the fact that we don’t need to compare the new node with all existing nodes of same depth (same control variable), but only with the nodes situated in the hash table at the same hash index. More than one hash table (exactly one hash table for each control variable) can be used and thus even less nodes is needed to compare. The lowest 4 control variables can use simple arrays instead of hash tables because the size of node's vector is small enough.

4 Replacement of multiplexers

The smallest ROBDD is transformed to multiplexer tree and can be improved by replacing some of the multiplexers with direct input variable or negated input variable (in both cases these signals are called "residual variable") and some other multiplexers can be replaced with basic logic gate.

In fact, the "residual variable" replacement can be used always and already during the creation of ROBDD.

Let *u* to be a MUX. Let *r*(*u*) to be a control input of *u* and *l*(*u*) to be an inverted control input of *u*. Let *left*(*u*) to be a data input of *u*, which is driven to output when *l*(*u*) = '0'. Let *right*(*u*) to be a data input of *u*, which is driven to output when *r*(*u*) = '0'. Let '0' to be a constant zero value and '1' to be a constant one value. Then the rules for MUX replacement are as follows:

- If *left*(*u*) = '0' and *right*(*u*) = '1', then *u* can be replaced with *r*(*u*). (residual variable)
- If *left*(*u*) = '1' and *right*(*u*) = '0', then *u* can be replaced with *l*(*u*). (residual variable)
- If *left*(*u*) = '0' and *right*(*u*) is neither '0' nor '1', then *u* can be replaced with logic gate AND(*right*(*u*), *r*(*u*)).
- If *right*(*u*) = '0' and *left*(*u*) is neither '0' nor '1', then *u* can be replaced with logic gate AND(*left*(*u*), *l*(*u*)).
- If *left*(*u*) = '1' and *right*(*u*) is neither '0' nor '1', then *u* can be replaced with logic gate OR(*right*(*u*), *l*(*u*)).
- If *right*(*u*) = '1' and *left*(*u*) is neither '0' nor '1', then *u* can be replaced with logic gate OR(*left*(*u*), *r*(*u*)).
- If *left*(*u*) = input variable *v* and *right*(*u*) = negated input variable !*v*, then *u* can be replaced with logic gate XOR(*v*, *r*(*u*)).
- If *right*(*u*) = input variable *v* and *left*(*u*) = negated input variable !*v*, then *u* can be replaced with logic gate XOR(*v*, *l*(*u*)).

5 Experimental results

During the testing, we used IWLS'93 Benchmark Set: Version 4.0. The testing was mainly aimed for evaluating the time needed for synthesis and rate of reduction (number of nodes after reduction in comparison to number of nodes before reduction).

We tested several possibilities for variable ordering complexity *c* (described in Section 2) – 0 (constant), 1 (linear) and *n* (factorial). The hash function uses *k* = 16.

There are three tables (Table 1-3), each for one type of variable ordering complexity. The first column displays the name of benchmark circuit. The column "Before optimization" shows number of nodes in BDD before optimization. The next column "After optimization" shows number of nodes in optimized BDD with BDD reduction (Section 2) and all residual variables (Section 4). The column named "Reduction ratio" shows the percent improvement of optimized circuit in comparison to the original circuit. The column "Logic gates" shows number of nodes after optimization, which can be implemented with use of the basic logic gates. The next column "Logic gate ratio" shows the efficiency of our MUX-1 replacement by logic gates in the optimized multiplexer tree. The last column "Time (ms)" displays total real time in milliseconds required by this synthesis process.

Table 1 shows that even constant variable ordering offers very good reduction ratio, many times more than 74,19%. This method is suitable as a basic element for other algorithms, which combine the ROBDD with other more sophisticated methods for variable ordering problem.

Table 1. Constant complexity of variable ordering problem.

Benchmark Circuit	Before optimization	After optimization	Reduction ratio	Logic gates	Logic gate ratio	Time (ms)
con1	94	15	84,04%	7	46,67%	2
rd53	93	24	74,19%	10	41,67%	4
rd84	1020	70	93,14%	22	31,43%	5
sqrt8	336	39	88,39%	15	38,46%	5
9sym	511	31	93,93%	10	32,26%	2
sao2	4092	188	95,41%	88	46,81%	11
cm152a	2047	9	99,56%	2	22,22%	1
cm151a	8190	1018	87,57%	78	7,66%	37
cm150a	1966082	131069	93,75%	549	0,42%	2671
Parity	65535	3419	94,78%	2	0,06%	90
t481	65535	607	99,07%	174	28,67%	49
5xp1	628	89	85,83%	29	32,58%	13
misex1	473	52	89,00%	33	63,46%	9
Inc	1015	90	91,13%	54	60,00%	12
squar5	248	38	84,68%	23	60,53%	8
xor5	31	7	77,42%	2	28,57%	1
Cordic	16777214	10403	99,94%	3231	31,06%	3295

The results of linear complexity (Table 2) show quite good improvement in some cases and the time needed for synthesis process is still relative short. For example amount of nodes after optimization for circuit "cordic" is 62,53% smaller than the result of constant complexity. This algorithm can be used as a basic element in other more sophisticated methods too, for example using dynamic approaches on each ROBDD created by this linear algorithm.

Table 2. Linear complexity of variable ordering problem.

Benchmark Circuit	Before optimization	After optimization	Reduction ratio	Logic gates	Logic gate ratio	Time (ms)
con1	94	13	86,17%	7	53,85%	9
rd53	93	24	74,19%	10	41,67%	16
rd84	1020	70	93,14%	22	31,43%	39
sqrt8	336	37	88,99%	14	37,84%	24
9sym	511	31	93,93%	10	32,26%	13
sao2	4092	154	96,24%	83	53,90%	69
cm152a	2047	9	99,56%	2	22,22%	9
cm151a	8190	30	99,63%	16	53,33%	54
cm150a	1966082	31	99,99%	16	51,61%	16572
parity	65535	3419	94,78%	2	0,06%	494
t481	65535	314	99,52%	87	27,71%	274
5xp1	628	76	87,90%	21	27,63%	48
misex1	473	48	89,85%	27	56,25%	48
inc	1015	85	91,63%	48	56,47%	69
squar5	248	34	86,29%	22	64,71%	41
xor5	31	7	77,42%	2	28,57%	6
cordic	16777214	3898	99,98%	748	19,19%	89662

Bigger complexities (Table 3) are suitable for only small number of input variables, for example factorial variable ordering is worth to be used for Boolean functions with maximum of 10 input variables. The more input variables the function has, the less complex variable ordering should be used. Even constant variable ordering reaches relative high reduction ratios when used for bigger

Boolean functions (cordic - 99,94%), so that more complex variable orderings do not often bring big improvements and thus are not worth using them.

The last table with factorial complexity does not contain all benchmark circuits because some of them had too much input variables for factorial complexity and thus the time required by synthesis process was too long.

Table 3. Factorial complexity of variable ordering problem.

Benchmark Circuit	Before optimization	After optimization	Reduction ratio	Logic gates	Logic gate ratio	Time (ms)
con1	94	11	88,30%	5	45,46%	327
rd53	93	24	74,19%	10	41,67%	119
rd84	1020	70	93,14%	22	31,43%	64134
sqrt8	336	28	91,67%	12	42,86%	17065
9sym	511	31	93,93%	10	32,26%	162304
sao2	4092	95	97,68%	52	54,74%	8389400
5xp1	628	62	90,13%	18	29,03%	6090
misex1	473	44	90,70%	22	50,00%	4679
inc	1015	78	92,32%	44	56,41%	13622
squar5	248	33	86,69%	20	60,61%	496
xor5	31	7	77,42%	2	28,57%	63

The experimental results also show that speed-up improvements based on hash table and top-down approach allow us to synthesize circuits with more input variables or to use more complicated algorithm for solving variable ordering problem in order to achieve even more optimized combinational circuit.

The testing has shown that even after optimization can be still relative big proportion of nodes used as basic logic gates instead of multiplexers. Sometimes the amount of logic gates is even higher than the number of multiplexers. According to the experimental results, the logic gate ratio generally decreases as we increase the number of input variables. On the other hand, this rate depends markedly on the Boolean function itself. For example circuit “cordic” uses 23 input variables and the logic gate ratio is 31,06% (Table 1). We need generally 10 transistors for each MUX-1 in CMOS technology. If we replace multiplexer with OR-2 or AND-2 gate, we need only 6 transistors (amount of transistors is reduced by 40%). In case of replacing a MUX-1 with XOR-2 gate the number of transistors needed falls from 10 to 8, which means another 20% reduction rate. For example amount of transistors needed to realise circuit “rd53” can be reduced by 15%.

6 Conclusion

Our research presents a new method for automatic synthesis and optimization of combinational logic based on BDD. Such BDD represents a highly optimized multiplexer tree, which is achieved by using basic BDD optimization methods and after that, the method of residual variables is applied. A multiplexer tree obtained through conversion of a BDD generated by this algorithm may require additional inverter circuit to provide residual variable in both positive and negative forms. The area reduced by our algorithm is significantly greater compared to this addition, and since the multiplexers are far more complex, this inverter circuit can be neglected.

The main benefits of this new method are better speed of synthesis process that can be seen mainly at Boolean functions with higher number of input variables. This is caused by using hash tables and top-down approach, which removes BDD nodes as soon as possible and avoids passing by all paths of BDD. The second benefit is a possibility to replace some multiplexers with basic logic gates AND, OR and XOR. For each replaced multiplexer exactly one of these logic gates is always used. For example 41,67% of multiplexers can be replaced in “rd53”, which results in 15% less transistors in CMOS technology.

The computational requirements of this algorithm are relative low thanks to the hash table and top-down approach. Our method can be used in other algorithms aimed for solving variable ordering problem like dynamical ordering optimization algorithms based on heuristics [10-11].

The output of this program is in VHDL or Verilog and can be further synthesized in various technologies, where some can prefer only multiplexers and some replacement of multiplexers by basic logic gates. The output has been verified in both languages using standard simulations provided for these languages (VHDL and Verilog).

This method is able to use more Boolean functions in order to design combinational logic with more than one output. In future work, we plan to add another optimization method based on merging equivalent subtrees from multiple binary decision diagrams with help of using one common hash table in order to improve the synthesis of combinational logic circuits with more than one output. We also plan to improve the solution of variable ordering problem with stochastic optimization algorithms.

We plan also a comparison of our algorithms with pure NAND-based logic synthesis (common technique without multiplexer trees).

Acknowledgement: This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 “Optimization of low-power design of digital and mixed integrated systems“).

References

[1] P. Metzgen and D. Nancekievill, “Multiplexer restructuring for FPGA implementation cost reduction,” in 42nd Annual Design Automation Conf., ser. DAC ’05, 2005, pp. 421-426.

[2] D. Macko , K. Jelemenská, "Managing digital-system power at the system level," in Proc. of IEEE Africon 2013 Sustainable Engineering for a Better Future, 2013.

[3] M. Siebert, E. Gramatova, "Delay Fault Coverage Increasing in Digital Circuits," in Proc. of Euromicro Conference on Digital System Design (DSD), 2013, pp. 475-478.

[4] S. Kristofik, E. Gramatova, "Redundancy algorithm for embedded memories with block-based architecture," in Proc. of IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2013, pp. 271-274.

[5] N. S. Li, J. D. Huang, and H. J. Huang,"Low Power Multiplexer Tree Design Using Dynamic Propagation Path Control," in Proc. of IEEE APCCAS, Dec. 2008, PP. 838-841.

[6] K. S. Brace, R. E. Bryant, R. L. Rudell, “Efficient Implementation of a BDD Package,” In: 27th ACM/IEEE Design Automation Conference, IEEE, 1990, pp. 40-45.

[7] P. Pištek, M. Kolesár, K. Jelemenská, “Optimization of multiplexer trees using modified truth table,” In: 2010 Int. Conf. on App. Electronics (AE), Pilsen: IEEE, 2010, pp. 265-268.

[8] P. W. C. Prasad, et al., “Binary Decision Diagrams: An Improved Variable Ordering Using Graph Representation of Boolean Functions,” International Journal of Computational Science,1(1): pp.1-7, 2006.

[9] R. Ebendt, G. Fey, R. Drechsler, “Advanced BDD Optimization,” Springer, 2005, 222p, ISBN 978-0-387-25453-1.

[10] R. Drechsler, “Evaluation of Static Variable Ordering Heuristics for MDD Construction,” ISMVL 2002: pp.254-260, 2002.

[11] I. Furdu, B. Patrut, “Genetic Algorithm for Ordered Decision Diagrams Optimization,” Proceedings of ICMI 45, 2006, pp. 437-444.

Power Estimation of System-Level Hardware Model

Michal LIĎÁK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xlidak@is.stuba.sk`

Abstract. Power estimation is an indispensable part of hardware development. Since the early stages of a hardware design, at the Electronic System Level (ESL), there are opportunities to reduce power. However, there is relatively small number of available industry solutions for estimating power at ESL and these are often too difficult to use. In this paper, we propose a new method for fast and simple relative power estimation. The power estimation is based on a given SystemC hardware model and switching activity from the simulation. Our method helps designers to determine the most power-demanding part of the design or which one from the designed hardware architectures consumes less power.

1 Introduction

Controlling the power dissipation in the hardware design is a current issue today. There is a need for reducing power not just because of obvious energy savings it could bring, but there are also other limitations to consider (e.g. thermal limitations, reliability, battery life) [1]. In the recent past, there has been a rapid development of various mobile devices, ranging from laptops to cell phones and tablets, or other gadgets running on batteries. For a customer, the performance of such device is not the only important parameter, but its battery life as well. To satisfy the increasing demand for performance of mobile devices, or any other hardware for that matter, the power consumption inevitably grows as well. The ever increasing number of transistors on a single die and transistor switching speed result in thermal limitations; without the power regulation, modern microchips would be impossible to cool [2].

Nowadays, the hardware design process starts from the Electronic System Level (ESL). It offers higher level of abstraction comparing to the Register Transfer Level (RTL) and enables faster and more efficient design of a complex hardware. The sooner we will start the reduction of power consumption, the better the results can be. At the ESL, there are many opportunities for power savings [1]. Power estimation takes place before the reduction itself and that is why it is desirable to know estimated power consumption as soon as possible. But ESL is too abstract and

[*] Master degree study programme in field: Computer Engineering
Supervisor: Dominik Macko, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

there are still many variables of the design unknown at such early stage in the designing process. This makes it more difficult to estimate power directly from this level.

Power in the digital CMOS (Complementary Metal Oxide Semiconductor) circuits consists of two basic components: dynamic and static (leakage power). Dynamic power could be further divided to switching and short-circuit power. Switching power is consumed when a transistor changes its state. Short-circuit power is drained during a brief moment of switching a transistor, when the current is conducting directly from source to ground. Static power represents the power dissipation even without any switching activity [3]. With smaller manufacturing processes (of 90 nm and below) the need for managing the leakage power emerged as well, as it can easily become the primary source of power dissipation in a design [4].

In this paper we propose a novel system-level power estimation method which could help to overcome mentioned problems of power estimation at this level. We have designed a software tool implementing our method for a relative power estimation of a hardware design. Section 2 describes related work in this area. In Section 3, we introduce our proposed power estimation method and its software implementation is given in Section 4. Finally, we conclude our work in Section 5.

2 Related work

There is number of known approaches for estimating power consumption directly at ESL. In this work, some of them, namely spreadsheet based, power models based, and power macro-model based power estimation have been analysed.

In spreadsheet approach [4], the designer does not need any special tools to estimate power. Spreadsheets are produced by FPGA vendors and the designers can get an estimate on power consumption of a given area of design by filling in forms in data sheets. In this way they can get early power estimation based on which some further designing decisions can be made. The main problem of this approach is its inaccuracy and the need of filling multiple forms which is time-consuming for the user.

Power models [5] represent power data for every instruction from a processor’s instruction set acquired by repeated measurement of current drained by the processor executing sequences of instructions. This approach could be more accurate but the process of creating such a model is very demanding in terms of time and computing power and it can take several months to complete.

Power macro-model [6] is made of *n*-dimensional tables of *N* different variables that affect power consumption, i.e. input signal probability, average transition density etc. The resulting values or power estimation are a function of these variables.

Although this area has been extensively researched, there is relatively small number of available industry solutions for estimating power at ESL and these are either too extensive or time-consuming to learn and use. Still the RTL power estimation is mostly in use and even the proposed solutions for ESL power estimation often just use and/or guide tools working at the lower levels.

Several available industry tools for system-level power estimation have been analysed as well. Xilinx Power Estimator [7] and Altera® Quartus® II PowerPlay [8] are spreadsheet based tools requiring only the Microsoft Excel software. Other tools were implemented as software programs: Synopsys PrimeTime PX [9], Cadence® Incisive® Palladium® DPA [10], Mentor Graphics Vista Architect [11]. In general, these tools are compatible only with the UNIX-based systems, they are quite extensive and/or a part of the large-scale designing solutions what makes them difficult to use.

The main issue of all mentioned solutions is their complexity and difficulty to use for less experienced users. To get the power estimation of a design is often very time-consuming or even impossible to achieve, if the design is not developed from a scratch within the given tool. Therefore our work was aimed for a light-weighted and simple power estimating tool that could provide the designer fast power estimation of the entered design.

3 The system-level power estimation method proposal

Based on the analysis of the existing power estimating approaches, our own estimation method was designed. The method is aimed mainly to be simple and fast. Because of that, we do not implement a way to execute simulation of a design; we rather use third-party variables switching data from the design simulation. This file will be stored in a Value Change Dump file (VCD) and it will be the first program input. The second input file will be the design source code itself stored in a SystemC file. The third input is optional and it is in the form of a text file with user-specified parameters of simulation.

Both components of power consumption will be estimated: static (leakage) and dynamic power.

To determine the leakage power of a design, firstly, we need to find out the size of each module in a SystemC source file. In this way we could identify also the contribution of each module in total static power consumption of the design. The module size will be evaluated based on the number of characters needed to describe that module. The size of embedded modules is also needed to be taken into consideration. Secondly, we take into account the count of the chosen language structures which may take more time to compute, such as the conditions *if-else* or cycles *for*. These will be weighted with a predefined constant to appreciate their greater computing demand and thus a greater power demand.

Dynamic power is also evaluated for each separate module by determining the count of the variables changes of a given module in the VCD file. The variables are weighted based on bit length and a type of a signal, e.g. different weights for a memory than for a port variable. We can assess the bit length from the VCD file and signal type information from the SystemC source file.

Both, static and dynamic component of power consumption need to be weighted as well to be able to compare two separate designs of the same hardware. Constants will be defined to reflect the impact of each of two components in total power dissipation of a design.

The benefit of this approach is the direct power estimation from ESL which makes it really fast and can be used in a very early stage of a design. It requires only a minimal input from the user. On the other hand, it just offers the estimation of relative power consumption and therefore it can be only used to compare two different designs of the same system, or alternatively to compare power consumption of different modules of a single hardware design.

The described operation of the proposed method is displayed in Figure 1. The process starts at the left hand side with input files analysis, continues to evaluate two components of power consumption and ends up with the total relative power consumption estimation of the design.

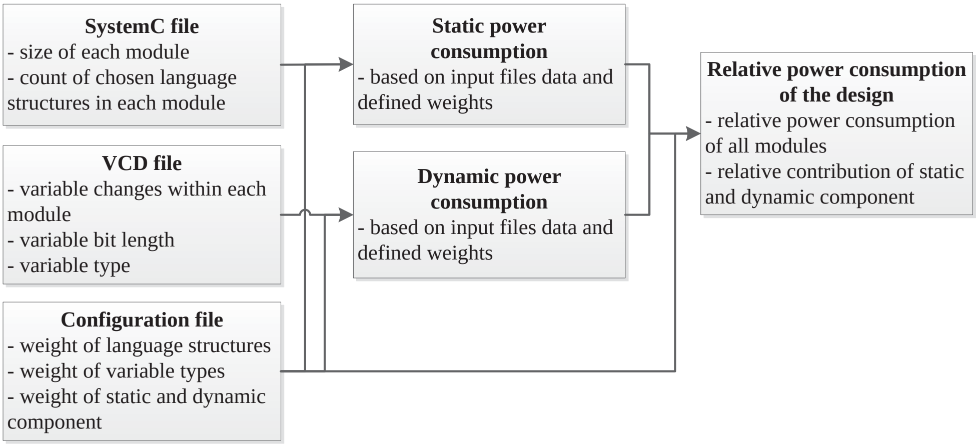

SystemC file
- size of each module
- count of chosen language structures in each module
VCD file
- variable changes within each module
- variable bit length
- variable type
Configuration file
- weight of language structures
- weight of variable types
- weight of static and dynamic component
Static power consumption
- based on input files data and defined weights
Dynamic power consumption
- based on input files data and defined weights
Relative power consumption of the design
- relative power consumption of all modules
- relative contribution of static and dynamic component

Figure 1. Overview of proposed method.

4 ESL power estimation software tool

Our goal is to create a light-weight and user-friendly tool implementing the proposed method. We focus on making it simple and easily-extensible in the future.

The software tool works in the following way. After starting the program user loads a hardware design and signal switching information from the specified files. If the user has a text file containing the supporting information, it can be loaded at this stage as well. If not, the default values will be used. This supporting information could be also changed directly in the user interface. In case of changing default values, the entered values will be saved as the new defaults.

Now the program has all the information needed to estimate power consumption of the design. User can start the estimation by pressing respective button. Analysis of the SystemC source file and VCD file will take place, and then based on the obtained information the power will be estimated accordingly to afore-mentioned method. The results will be displayed and saved automatically to be compared later with power estimation of another hardware design.

After successful power estimation user can load the next design to be estimated for power consumption in the same way as described. After power-estimating at least two hardware designs it is possible for the user to choose two of them for comparison. The tool will evaluate and display the results of analysis and comparison of both designs.

4.1 Architecture of the software tool

Functionality of the software tool will be divided into four parts, which will communicate to each other. This will help to achieve the desired transparency and simplicity of the tool as well as it will make the potential future changes easier. It is also necessary to separate the user interface from other parts of the software to maintain responsiveness of the program interface during longer computations. The proposed modules and communication among them are illustrated in Figure 2.

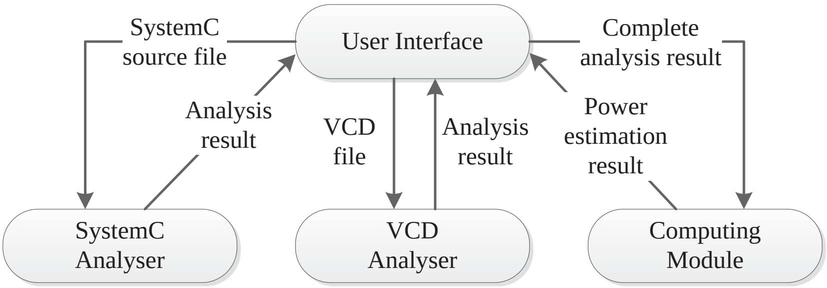


Figure 2. Architecture of the software tool.

The user controls the tool from *User Interface*. All the possible user actions, i.e. loading the input files, changing parameters of the power estimation, starting the power estimation and starting the comparison of two power-estimation results of two hardware designs are performed in this module. It also connects all the modules together and the other ones cannot communicate directly between each other. User-specified SystemC and VCD files are loaded and sent as a path to their respective *Analyser* modules for processing. The text file with user-specified parameters of simulation is processed in *User Interface*. These parameters could be entered also directly within this module. *User Interface* gathers the information from both *Analyser* modules and merges it into a single file, which will be sent to *Computing Module* for the evaluation of power consumption. The file with user-specified parameters is sent separately.

SystemC and *VCD Analysers* contain code for analysis of SystemC and VCD files respectively. We will design these modules specifically for the needs of our tool. These modules will gather the data needed for power estimation according to Section 3 and send the results back to *User Interface*.

Computing Module executes the power estimation itself, or it compares the selected two power-estimation results of two hardware designs. The results are sent back to *User Interface*, which will display them.

All of the results from modules are stored in text files and only the paths to these files are being sent. This should simplify the communication within the program. However, the modules on both sending and receiving side must follow the defined format of the text files.

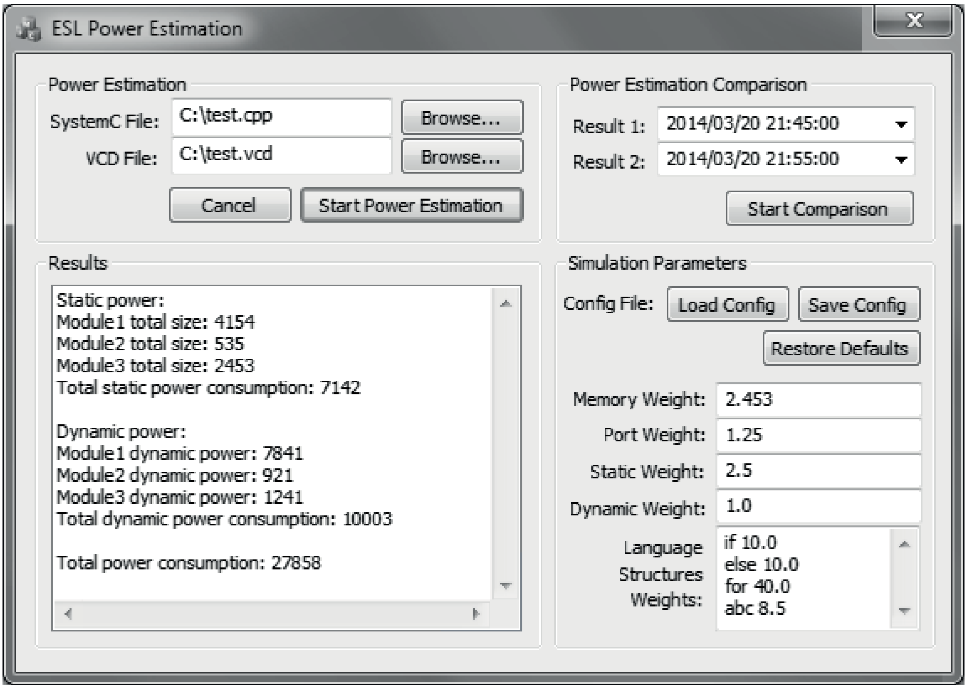
4.2 Example of the software tool results

Figure 3 illustrates the example results of power estimation of a hardware design. Please note that all the values are the relative ones.

In the *Results* section the *Static power* shows at first the size of each module of the design in number of non-whitespace characters. The size of embedded modules is included, as well as the weighted size of defined language structures. For example, in this case every occurrence of phrase *if* in the source code is counted as 10 characters instead of just 2. Other language phrases are treated similarly. Corresponding values of weights can be found in *Simulation parameters* section. *Total static power consumption* is the sum of sizes of all modules.

Dynamic power for each module is a function of total count of changes of all variables in that module, corresponding variable lengths and variable types (values for memory and port variable types can be found in *Simulation parameters* section). *Total dynamic power consumption* is the sum of the dynamic power of all modules.

Total power consumption of the design is a sum of *Static power* and *Dynamic power*, enlarged by *Static weight* and *Dynamic weight* respectively.



SystemC File: C:\test.cpp

VCD File: C:\test.vcd

Result 1: 2014/03/20 21:45:00

Result 2: 2014/03/20 21:55:00

```
Static power:
Module1 total size: 4154
Module2 total size: 535
Module3 total size: 2453
Total static power consumption: 7142

Dynamic power:
Module1 dynamic power: 7841
Module2 dynamic power: 921
Module3 dynamic power: 1241
Total dynamic power consumption: 10003

Total power consumption: 27858
```

Memory Weight: 2.453

Port Weight: 1.25

Static Weight: 2.5

Dynamic Weight: 1.0

Language Structures Weights:

```
if 10.0
else 10.0
for 40.0
abc 8.5
```

Figure 3. An example of the power estimation results.

5 Conclusion and future work

In this paper, we proposed a novel approach to the problem of estimating power of a hardware design at Electronic System Level. Our goal is to create a simple tool for a quick and easy power estimation using our method. A SystemC source file with the hardware design specification and a VCD file containing a switching data from the design simulation are provided as the tool inputs. Output is either the result of relative power consumption of the modules within the hardware design, or the comparison of power-estimations of two separate designs of the same hardware.

We believe this method could provide relatively accurate results in comparison to other available industry tools. However, finding the right values for parameters of the power estimation is crucial to make this approach viable. In the future, we will try to find such values or even additional features of design to consider in order to make our tool as accurate as possible.

Acknowledgement: This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 “Optimization of low-power design of digital and mixed integrated systems“).

References

[1] Ahuja, S.: *High Level Power Estimation and Reduction Techniques for Power Aware Hardware Design*. Dissertation, Faculty of the Virginia Polytechnic Institute and State University, (2010).

[2] Weste, N.H.E., Harris, D.M.: *CMOS VLSI Design: A Circuits and Systems Perspective*. (2011).

[3] Shauly, E.N.: CMOS Leakage and Power Reduction in Transistors and Circuits: Process and Layout Considerations. *Journal of Low Power Electronics and Applications*, (2012), vol. 2, no. 1, pp. 1-29.

[4] Cadence Design Systems: *A practical guide to low power design: User experience with CPF*, (2012). Available at: http://www.si2.org/?page=1061

[5] Benini, L. et al.: A power modeling and estimation framework for VLIW-based embedded systems. *Proceedings of International Workshop-Power And Timing Modeling, Optimization and Simulation*, PATMOS’01, (2001).

[6] Gupta, S., Najm, F.N.: Power macromodeling for high level power estimation. In: *DAC '97 Proceedings of the 34th annual Design Automation Conference*, (1997), pp. 365-370.

[7] Xilinx: *Xilinx Power Estimator User Guide*. [Online; accessed November 11, 2013]. Available at: www.xilinx.com/support/documentation/sw_manuals/xilinx2012_4/ug440-xilinx-power-estimator.pdf

[8] Altera Corporation: *PowerPlay Power Analysis*. [Online; accessed October 10, 2013]. Available at: http://www.altera.com/literature/hb/qts/qts_qii53013.pdf

[9] Synopsys: *PrimeTime*. [Online; accessed October 12, 2013]. Available at: http://www.synopsys.com/Tools/Implementation/SignOff/Pages/PrimeTime.aspx

[10] Cadence Design Systems: *Incisive Palladium III Dynamic Power Analysis*. [Online; accessed October 12, 2013]. Available at: http://www.cadence.com/rl/Resources/datasheets/incisive_palladium_dpa_DS.pdf

[11] Mentor Graphics: *Vista Architect: System Level Design Solution for Performance and Power*. [Online; accessed November 11, 2013]. Available at: http://www.mentor.com/esl/vista/upload/vista-007d1f5d-41a7-41cd-97f4-06bb92d1a2eb

A New Multiplexer-based Circuit Synthesis Method Optimized to Multiple Parameters

Marián MARUNIAK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
mar.maruniak@gmail.com

Abstract. Multiplexers provide a promising area of optimizations because of their widespread use in VLSI circuit synthesis. Thus far, countless optimization methods were proposed, focusing on various properties of multiplexers and multiplexer trees. However, optimizing only one parameter of such circuit without considering other important parameters can result in an unnecessary waste of resources. This may be prevented by using multiple parameter optimizations. In this paper we deal with the trade-offs between three important parameters of multiplexer trees, which are the area, the power and the average path length. To achieve this, we take advantage of optimization methods designed for multiplexers along with the binary decision diagram optimization methods.

1 Introduction

As the MOSFET technology reaches its fundamental physical limits, it is even more important to apply optimizations on the design level of digital circuit synthesis. Such approach allows keeping up with the ever increasing demands on electronic devices. This work focuses on multiplexers and multiplexer trees, because of their significant contribution to the total area of the circuit, which climbs up to 25% in average FPGA designs [1]. Since high fan-in multiplexers provide poor scalability and are inefficient to manufacture, it is most common to implement them as a multiplexer tree of several lower fan-in multiplexers, usually 2-to-1 multiplexers. The internal structure of a multiplexer tree can be altered in many ways whilst retaining its original function, what may result in various positive or even negative changes in characteristic parameters of given multiplexer tree.

The main goal of this paper is to achieve the best possible trade-offs between the three of the most important parameters of a multiplexer tree, which are the total circuit area, power consumption and the average path length (APL). APL represents the average length of all the paths leading from each terminal node to the root node of a BDD. Length of a path is given by the number of edges contained in this path [3]. The most crucial of these parameters is the area, since it influences the power and the APL as well. Power consumption of the multiplexer tree is primarily influenced by the dynamic power consumption caused by internal signal switching and various leakages. APL contributes to higher performance of the final circuit, or even lower power dissipation in the case of the pass transistor logic.

[*] Master degree study programme in field: Computer Engineering
Supervisor: Peter Pištek, Institute of Computer Systems and Networks, Faculty of Informatics andInformation Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 405–410.

An extensive research in the area of multiplexer tree optimization has been done so far, resulting in a number of different optimization techniques. A big contribution to multiplexer tree optimization was the idea of using binary decision diagrams (BDDs) [2] as their structural description. Most importantly, this means a BDD can be directly mapped onto a digital circuit. A simple conversion of a multiplexer tree using basic BDD reduction methods and respecting a specific input variable ordering, results in a Reduced Ordered BDD (ROBDD) [5]. Using a ROBDD was confirmed [4] to provide a minimum of 17% reduction of the circuit area in our previous work. To further optimize a ROBDD, an optimal input variable ordering has to be found. Since this is an NP-complete problem, heuristic approaches introduced in [5] are preferred over exact methods. BDDs can be used with a great advantage to reduce the APL [6] as well as power consumption [7]. Another efficient way to optimize a multiplexer tree is the use of modified truth table [8] which always results in an improvement of area, power and APL simultaneously, as confirmed in the final sections of this paper.

This paper is structured as follows. The basics of multiplexers, BDD and their mutual relations are described in Section 2. Section 3 describes the algorithm proposed in this paper. In Section 4, the experimental results are shown and the final section concludes this paper along with achieved results.

2 Preliminaries

In this section, the properties of a multiplexer tree, BDD and their mutual mapping are defined. The input of proposed algorithm is a Boolean function (further B-function) *f* of *n* input variables $x_0, x_1, \ldots, x_{n-1}$ and a binary vector *y*. This is denoted as $f(x_0, x_1, \ldots, x_{n-1}) = y$, where the order of input variables corresponds to the decreasing weights assigned to the variables from left to right, starting from the weight of 2^{n-1} for variable x_0 down to the weight of 2^0 for x_{n-1}. Such ordering can be then denoted as x_0 - x_1 - ... - x_{n-1}. Any B-function *f* specified by its binary vector *y* and a fixed variable ordering can be easily expressed as a BDD [3].

BDD is a rooted, directed acyclic graph consisting of one or two terminal nodes of out-degree zero labelled by *I* and *0* and a set of variable nodes *u* of out-degree two with two outgoing edges labelled *low* and *high*. BDD has only one node with no parent edge, called the root node. For optimization purposes we use ROBDDs (further simply referred to as BDD) which represent a canonical form of given B-function, where the identical nodes are merged, nodes with identic successors are eliminated and a fixed ordering of input variables is respected [4]. Because of the use of modified truth table, the specification of the BDD has to be altered, such that terminal nodes now gain values from *0*, *I*, *x* and $\overline{x}$ where *x* is the residual variable. In digital circuitry, BDDs are often used as a structural representation of multiplexers.

Multiplexer is a switching device with *n* control inputs $c_0, c_1, \ldots, c_{n-1}$, 2^n data inputs $d_0, d_1, \ldots, d_{2^n-1}$ and one data output *f*. Since only 2-to-1 multiplexers will be used in this work, they will be further referred to only as multiplexer and multiplexer tree composed of 2-to-1 multiplexers will be referred to as multiplexer tree. For example, B-function $f(x_1, x_2) = \overline{x_1}.x_2 + x_1.\overline{x_2}$ can be expressed by a BDD shown in Figure 1. Mapping this BDD to a multiplexer is then a simple procedure, illustrated in the same figure.

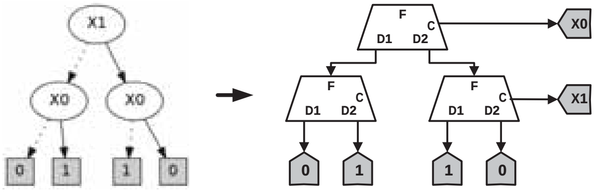


Figure 1. Mapping a BDD to a multiplexer tree.

3 Proposed Algorithm

This work combines several optimization methods into a novel multiplexer optimization algorithm. It is important to realize that optimization of only one parameter of a multiplexer tree can result in multiple results equal in values of chosen parameter but different in values of other, not considered parameters. In some cases, this difference can become non-negligible and result in an undesired waste of resources. Therefore, the aim of this approach is to find the best possible trade-off between area, power and APL. Another advantage is the possibility of optimization with configurable trade-off between these parameters. Designed algorithm is divided into three consecutive stages, each performing a different optimization method starting with the modified truth table method followed by the BDD optimization and finally the genetic algorithm. These methods were carefully chosen such that each method has an impact on all of the three specified parameters.

3.1 Modified truth table

This optimization method results in a removal of the variable with the highest weight in the ordering of given B-function by encoding the binary vector in a special way. This process is described in detail in [4, 8]. Due to the removal of the largest level of the multiplexer tree, this method provides a large improvement in the area of optimized circuit. This is also reflected into lower power consumption. In terms of BDD, the removal of one layer results in a shorter APL in most cases. The main drawback of this method is the necessity of the positive and negative residual variable signal values, which need to be connected to the data inputs of given circuit.

3.2 Genetic Algorithm

The most important part of proposed approach is the genetic algorithm (GA), where all the trade-offs can be set and evaluated. The idea of using a genetic algorithm to optimize more than one parameter was already proposed in [9] where the author focuses on the trade-off between power and area of the circuit. With several improvements, similar approach can be used for the purpose of this work. Such GA will be then used to find an appropriate input variable ordering of BDDs constructed in previous stages of this algorithm, mentioned above.

The core term of GAs is the population, which represents a set of chromosomes, in this case input variable orderings. Each chromosome consists of *n* genes which represent the order of a particular variable. For example, a chromosome of 8 variables can be represented by following vector of genes. For simplicity, the variables in a chromosome are represented by their indices.

1	5	3	2	7	0	6	4

A population of a specific stage of GA is called a generation. Population is initially created by randomly generated chromosomes, which are being selected based on their fitness value to form a new generation. Selected chromosomes have a certain probability of genetic mutation and crossover being applied to them to slightly increase the diversity of current generation and decrease the chance of getting stuck in a local optimum. To retain the best chromosomes over all of the generations, a technique called elitism is introduced. Elitism ensures that the subsequent generations will never provide a worse (and therefore useless) result than the previous generations.

After creating the population, for each chromosome a fitness value is calculated determining its probability of being selected. This is done by normalizing the values of area, power and APL of given orderings to range (0, 1). Then, the fitness values are calculated by the following formula:

$$fitness = A \times area\_normalized + P \times power\_normalized + L \times apl\_normalized \tag{1}$$

Where A, P and L are the coefficients setting percentage weights of optimized parameters. The sum $A+P+L$ should be equal to 1 at all the times. Area is represented by the count of BDD nodes.

Power is given by dynamic switching of internal signals in the multiplexer circuit, calculated by probabilistic approach [7] slightly altered to work along with the modified truth table method. APL is calculated using the BDD folding algorithm proposed in [6].

After being set, the fitness values are redistributed using a roulette wheel algorithm. This algorithm is modified to greatly increase the selection probabilities of the best chromosomes against the worst ones. At this point, GA starts to populate a new generation by selecting chromosomes from current generation and possibly applying genetic mutation and crossover. The idea of genetic mutation is to invert a random gene in chromosome. In the case of a variable ordering, this operation is implemented as a swap of the variable on a random position and the variable positioned on the complementary position. For example, a mutation in chromosome of length 8 at position 2 can be illustrated as:

1	5	3	2	7	0	6	4	➔	1	6	3	2	7	0	5	4

Another genetic operation called crossover causes two selected chromosomes to be cut at the same randomly chosen point and exchange their segments. Since a simple exchange of the parts of variable orderings can violate the uniqueness of each variable in such vectors, this operation will be interpreted in another way. Instead of exchanging parts, the chromosomes will append missing variables after the crossover point with respect to their ordering in the other chromosome. For example, let the crossover happen at position 5 for following chromosomes:

1	5	3	2	7 ‖	0	6	4		3	7	6	2	0 ‖	1	5	4
								➔								
3	7	6	2	0 ‖	4	1	5		1	5	3	2	7 ‖	6	0	4

Genetic algorithm starts with the creation of a new population (in this work we also reuse previous generations several times) and keeps applying mutation and crossover on selected chromosomes with predefined probability. This process is shown in Figure 2.

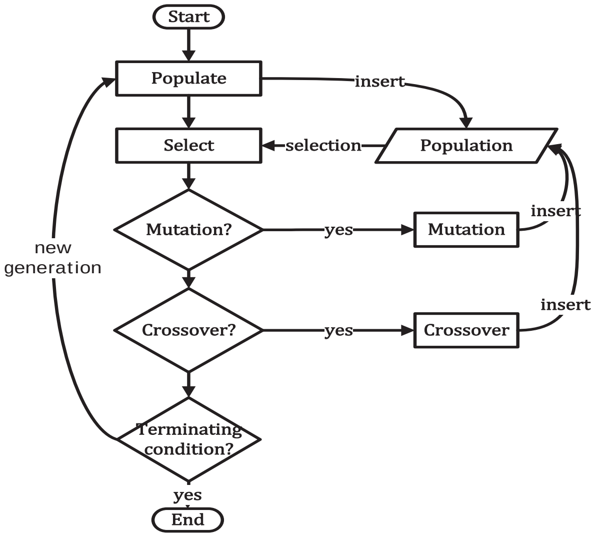


Figure 2. Genetic algorithm flowchart.

4 Experimental results

A series of exhaustive tests was performed on proposed algorithm using benchmark functions from LGSynth93 benchmark set [10]. Optimizations using the modified truth table and basic BDD reduction methods were applied to every test input. To illustrate the trade-off between parameters, two approaches for GA were used. First, the GA parameters were set to maximize the area reduction. The second part of testing was performed with equal values for all three parameters, hence to achieve their best trade-off. The probability of genetic mutation was set to 30% and the probability of crossover to 80%. As for elitism, 1% of best solutions were kept over all of the generations. The results are contained in Table 1.

Table 1. Experimental results.

Function	Non-opt nodes	Area			Power			APL			Avg T-off %
		toA	toT	T-off%	toA	toT	T-off %	toA	toT	T-off %	
majority	32	6	6	**0**	2,38	2,38	**0**	3,13	3,13	**0**	**0**
rd53	32	24	24	**0**	10,1	10,1	**0**	3,75	3,75	**0**	**0**
xor5	32	7	7	**0**	3,5	3,5	**0**	4	4	**0**	**0**
con1	96	15	17	**-13,33**	6,79	6,08	**10,58**	2,97	2,47	**16,84**	**4,7**
squar5	256	42	46	**-9,52**	18,48	17,38	**5,92**	2,09	1,84	**11,94**	**2,78**
sqrt8	340	37	40	**-8,1**	17,31	15,96	**8,45**	2,85	2,675	**6,14**	**2,16**
misex1	480	51	53	**-3,92**	21,14	19,69	**6,87**	2,86	2,55	**10,76**	**4,57**
9sym	512	31	31	**0**	10,22	10,22	**0**	7,13	7,13	**0**	**0**
5xp1	638	75	110	**-46,67**	50,93	35,03	**31,23**	3,42	2,66	**22,15**	**2,24**
inc	1024	89	98	**-10,11**	37,63	34,45	**8,44**	2,95	2,71	**8,13**	**2,15**
rd84	1024	19	19	**0**	23,69	23,69	**0**	6,79	6,79	**0**	**0**
cm152a	2048	15	15	**0**	6,5	6,5	**0**	2,66	2,66	**0**	**0**
sao2	4096	98	102	**-4,08**	27,43	22,19	**19,11**	3,21	2,6	**19,01**	**11,35**
cm151a	8192	30	32	**-6,67**	15,25	14,75	**3,28**	2,72	2,52	**7,22**	**1,28**
parity	65536	29	29	**0**	14,5	14,5	**0**	15	15	**0**	**0**
t481	65536	32	41	**-28,13**	21,31	12,86	**39,65**	4,53	4,13	**8,83**	**6,78**
cm150a	2097152	32	34	**-6,25**	12,75	12,13	**4,86**	4,13	3,53	**14,45**	**4,35**
mux	2097152	35	39	**-11,43**	14,41	13,25	**8,03**	3,54	2,85	**19,59**	**5,39**

The second column "Non-opt nodes" of Table 1 indicates the complexity of each function by showing the number of nodes of a complete, non-optimized BDD. The table of results is then is divided into three parts labelled as "Area", "Power" and "APL" containing the optimized values of these parameters. The area is expressed as the number of nodes of the final BDD. By power we understand the number of gate output transitions per global clock cycle, or simply called the switching activity. The last parameter, APL represents the average number of edges in all of the paths of the resulting BDD. Under columns labelled as "toA" are shown the results of tests focused on optimization to the best possible area. Label "toT" stands for the best trade-off optimization and "T-off" stands for the trade-off between these optimizations. A positive value of trade-off signalizes an improvement by using best trade-off optimization, whereas a negative value is in favour of the best area optimization. The rightmost column, "Avg T-off%" represents the average improvement of best trade-off optimization compared to best area optimization. Note that this value is always non-negative, therefore with better utilization of the circuit's parameters we can always achieve its overall improvement.

5 Conclusion

A novel algorithm for optimized multiplexer tree synthesis was proposed. Its main advantage is the possibility of multiplexer tree based circuit being optimized with respect to three parameters – area, power and APL with adjustable weights. Such approach provides better control of these parameters during the synthesis of given circuit. Another advantage is the possibility of shifting the trade-offs between parameters to any desired way. In previous section, an example of an evenly distributed trade-offs was shown. Moreover, from the experimental results it is evident that such trade-off shift always results in an overall circuit improvement or keeps it unchanged in the worst case.

Acknowledgement: This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 "Optimization of low-power design of digital and mixed integrated systems").

References

[1] Metzgen, P. N.: Multiplexer Restructuring for FPGA Implementation Cost Reduction. In: *Design Automation Conference, Proceedings 42nd*. Anaheim, California: IEEE, 2005, pp. 421-426.

[2] Akers, S. B.: Binary Decision Diagrams. In: *IEEE Transaction on Computers*, IEEE, 1978, C-27, no. 6, pp. 509-516.

[3] Ebendt, R., Gunther, W., Dreschler, R.: Minimization of Expected Path Length in BDDs Based on Local Changes, Proc. In: *ASP-DAC (Asia and South Pacific Design Automation Conf.),* Jan. 2004, pp. 866-871.

[4] Maruniak, M., Pištek, P.: Binary decision diagram optimization method based on multiplexer reduction methods, In: *International Conference on System Science and Engineering (ICSSE),* 2013 , pp. 395,399.

[5] Rudell, R.: Dynamic variable ordering for ordered binary decision diagrams, In: *Int'l Conf. on CAD,* 1993, pp. 42-47.

[6] Keren, O.: Reduction of Average Path Length in Binary Decision Diagrams by Spectral Methods, Computers, IEEE Transactions, 2008, vol.57, no.4, pp. 520-531.

[7] Lindgren, P.; Kerttu, M.; Thornton, M.; Drechsler, R.: Low power optimization technique for BDD mapped circuits, In: *Design Automation Conference,* 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific , pp. 615,621.

[8] Pištek, P., Kolesár, M., Jelemenská, K.: Optimization of Multiplexer Trees using Modified Truth Table. In: *Proceedings of Applied Electronics 2010 International Conference - AE 2010*. Plzeň, Česká republika: 2010, pp. 265-268.

[9] Chaudhury, S., Dutta, A.: Algorithmic Optimization of BDDs and Performance Evaluation for Multi-level Logic Circuits with Area and Power Trade-offs, Circuits and Systems, 2011, pp. 217-224.

[10] Benchmark Circuits: LGSynth93, http://www.cbl.ncsu.edu/CBL_Docs/lgs93.html, 1993.

Dynamic QoS-aware MPLS Traffic Engineering

Jakub OBETKO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
jakub.obetko@gmail.com

Abstract. Present network communication consists of multiple types of services. Some of these services have requirements for various QoS parameters such as delay or bandwidth. ISP's networks have to satisfy those requirements in order to provide service for customers. MPLS technology is widely used in present ISP's network. This paper deals with traffic engineering in MLPS L3VPN networks. We propose a new dynamic traffic engineering method, which distributes the traffic load across the network with respect to QoS requirements of a given traffic type. Further, we propose a server-based solution for implementing this method in the network.

1 Introduction

Present ISP backbone networks are expected to support various types of services e.g. real-time communication, private networks over public infrastructure, multimedia transfer. Some of these services might have various requirements for quality of service - QoS. One of the many examples is VoIP traffic with its requirements for end-to-end delay and jitter [1]. In order to provide satisfying level of QoS, resources have to be provided. Within large ISP backbone networks this can be achieved by adding more resources e.g. more high-speed links, however, network resources might not be utilized effectively. This situation can be caused by implementing simple IGP inside ISPs autonomous system - all of the traffic between two endpoints is forwarded over the same path, which often causes congestion and degradation of QoS. Traffic engineering (TE) solves this problem by diverting the traffic across the network so that network resources are effectively utilized. TE can be divided into multiple categories [2] from which we are focused on unicast MPLS-based traffic engineering methods.

2 Related work

A number of methods in various categories were proposed to effectively utilize the network potential. In our work, we focused on MPLS-based TE methods. Categorization of these methods is non-trivial task as they have different goals.

[*] Master degree study programme in field: Computer Engineering
Supervisor: Dr. Martin Hrubý, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

We have categorized them into following categories:

1. Methods focused on providing maximum number of paths in the network.
2. Methods focused on providing QoS.
3. Methods focused on path protection and restoration.

The first category deals with optimal path distribution in order to maximize the number of paths provided by the available network resources. DORA [3] and MIRA [4] are typical examples of such methods. The second category includes methods like TEAM [5] and TEQUILA [6], which are complex solutions for providing QoS by implementing TE in the network. In order to provide path protection few TE methods were proposed. These methods provide mainly 1+1 and 1:1 backup. Example of such method is Fast reroute with pre-established bypass tunnel in MPLS [7].

Present networks have to be flexible and achieve all of mentioned goals. All of analyzed methods for TE are focused only on one goal and therefore they achieve only one of necessary aspects required by present backbone networks. In our solution, we propose method, which provides maximum utilization of resources together with QoS and backup.

3 Proposal

In ISP networks, TE is important so that good resource utilization and QoS requirements can be satisfied. Providing TE in MPLS networks can facilitate the effort of ISP to implement optimal resource utilization in the network. MPLS technology is often used for providing VPN services. Therefore, we decided to design a TE method, which will be implemented in MPLS L3VPN networks.

3.1 Requirements on final system

The final system will be a traffic engineering server controlling TE in MPLS VPN networks. The server will be centralized, however the Label Switched Path (LSP) calculation can be distributed. The requirements for TE server are as follows:

- provide sufficient QoS for different types of traffic,
- effectively measure QoS parameters in the network,
- categorize traffic to specified classes according to their QoS requirements,
- provide dedicated LSPs across MPLS cloud for different customers,
- control the LSP selection so that it provides resources agreed in SLA,
- control the LSP selection so that network can satisfy as much further requests as possible,
- provide the backup, when the primary path is not available,
- provide management channels in order to communicate with routers,
- dynamically reoptimize LSP path so that it provides necessary resources and QoS, when the network is highly loaded.

There is always a trade-off between some of these requirements. Providing sufficient QoS results in selecting LSPs so that they can provide given QoS level. However, this could cause overutilization of some links, while others are underutilized. If we want to satisfy possible future LSP requests, we need to select LSPs effectively and monitor some link criticality index, so that we are noticed which links might be critical for future requests. If we select LSP based on this criticality parameter, we might not achieve sufficient QoS level on that LSP.

3.2 Design

The solution is designed as centralized TE server, which will be connected to the MPLS Label Switching Router (LSR) and control the traffic distribution in whole autonomous system's network. To implement MPLS TE, either IS-IS or OSPF has to be used so that information about link's constraints can be distributed to each LSR in MPLS network.. Our solution interacts with the OSPF routing protocol.

3.2.1 Traffic classification

In order to support QoS provisioning to different traffic types, we designed four classes of traffic, each with its specific requirements on QoS parameters. The classes with associated parameters thresholds are described in Table 1. All of the parameter's thresholds in Table 1 are given in one-way PE-to-PE (ingress Provider Edge router to egress Provider Edge router) manner.

Table 1. Proposed traffic classes.

	Delay	Jitter	Loss
Real-time class	150 ms	30 ms	1%
Low-latency class	1200 ms	N/A	1%
High-throughput class	N/A	N/A	4%
Standard class	N/A	N/A	N/A

As the table shows, classes are designed so that different types of traffic can be assigned to them. The typical example of Real-time class traffic is VoIP or videoconferencing. Client-server transactions or broadcast TV traffic might be assigned to Low-latency class and store and forward application's communication to High-throughput class. Standard class is designed to carry all of the traffic without specific QoS requirements. Bandwidth is also defined for each traffic class however this parameter can be defined individually by customer within Service Level Agreement (SLA).

Dedicated LSP is created for each one of the specified traffic classes and SLAs. SLA is defined in one-way PE-to-PE manner, which means that if customer wants to have bidirectional communication between VPN sites, he needs to define two SLAs. This allows higher granularity, as the customer might want to have e.g. high bandwidth for Low-latency class traffic only in one direction.

All of the traffic from a specific VPN site is forwarded into one of LSPs established for given customer and site.

3.2.2 QoS measurement

QoS requirements of different traffic types are usually defined in end-to-end manner. If we want to provide QoS in ISP's or backbone network, we need to think in the same way - PE-to-PE in MPLS VPN cloud. The problem occurs if we want to select the path (LSP) for e.g. Real-time class traffic. We want to guarantee one-way delay, jitter and loss to be below the defined threshold, but we do not have specific path, on which we can measure the values of these parameters. We decided to measure delay, jitter and loss only on lines in the network before the path is selected. Cisco IP SLA ICMP-JITTER operation [8] will be executed on every line and TE server will periodically collect the results of the measurements. This gives TE server the essential point of view on the level of QoS on the lines in the network and basis for path selection. The QoS measurement process is depicted in Figure 2. The arrow indicates the ICMP-JITTER operation. After the path is selected, QoS parameters are measured in PE-to-PE manner and compared with defined thresholds for specific traffic classes.

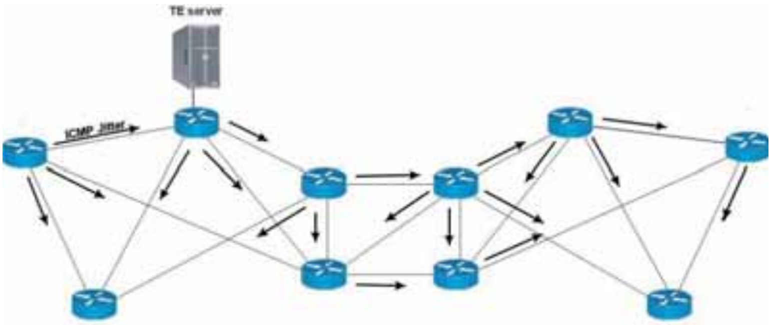


Figure 1. QoS measurement process.

3.2.3 Path selection process

Path selection process or algorithm is one of the core aspects of proposed method. In MPLS VPN networks, path is recognized as LSP. For each SLA, four LSPs are created, for each traffic class one. Traffic is then forwarded into those LSPs according to the class of service using Class-based Tunnel Selection mechanism.

Path selection algorithm is designed to effectively establish the path with respect to QoS requirements for a given LSP. The path calculation is effectively offloaded from TE server on head-end LSR (ingress PE router) of LSP. This is an advantage because TE server is not overloaded with path calculation, but the path reroute or calculation of backup path is also responsibility of given ingress PE router. Ingress PE router is calculating the path using CSPF [9] algorithm based either on IGP or TE metric of each link. However, the path has to be chosen with respect to QoS requirements, which means that TE server has to affect the process of path selection. For this purpose we defined new cost (metric) calculation equations, which reflect either the level of QoS on given link or link criticality. These two parameters are QoS metric, which reflects approximate level of QoS on given link and Link Criticality Index (LCI), which reflects link importance for future path requests. The calculation of both parameters is as follows:

$$QoS\ metric = (Delay_{SD} * Load_{DS}) * 100 \tag{1}$$

$$LCI = \sum_{SD} \left(\frac{p_{SlD}}{p_{SD}}\right) * \frac{1}{C_l - C_{used}} * 100 \tag{2}$$

QoS metric (1) reflects one-way delay [ms] on the link in direction from local router (S) to neighboring (D). This delay is multiplied by receive load [%], which is observed on the end of the link, which is connected to neighboring router.

Link Criticality Index (2) reflects completely different information about the link – its importance. This means, how important is link when considering future path requests. This metric calculation consists of two parts:

- the sum part, that reflects link criticality according to number of paths, which may cross it, where p_{SlD} is number of paths crossing the link for given PE pair and p_{SD} is number of all paths between the same two PEs,
- and the quotient, that reflects current utilization of the link, where C_l is maximum capacity of the link and C_{used} is already utilized part.

Each link has assigned both of mentioned parameters and path calculation is performed using CSPF algorithm on QoS metric or LCI. The paths for LSP which will carry Real-time class traffic or Low-latency class traffic is calculated using QoS metric while the paths for LSP which will carry High-throughput or Standard class traffic is calculated using LCI. The paths for two higher classes are chosen so that they satisfy defined QoS requirements and paths for lower two classes are built to minimally interfere with possible future requests for paths.

The responsibility of TE server is to update both of the parameters for each link periodically. The delay or load may change over the time and available bandwidth on link changes every time, new LSP is crossing given link. Therefore, TE server has to periodically monitor the actual level of QoS on each link and update both of the parameters in whole network.

After the paths are chosen, the QoS parameters can be measured in PE-to-PE way on path for 1st and 2nd class. If the paths do not satisfy the QoS requirements for given class, the path selection process is repeated.

3.2.4 Monitoring and reoptimization

After the LSPs for each class are established, level of QoS in PE-to-PE manner has to be monitored on each LSP for 1st and 2nd class. In Table 1, we defined the thresholds of QoS parameters. If the values of QoS parameters for given LSP exceeds those thresholds, the reoptimization of LSP is necessary. In our solution, we designed new algorithm for reoptimization sif such a situation occurs. The algorithm is based on "cleaning the path" for LSP, that has no longer satisfactory level of QoS. The algorithm is performed in following way:

1. Find the link *l* with highest QoS metric on the path of reoptimized LSP,
2. Identify the lowest class LSP which is crossing link *l*,
3. Reroute the LSP chosen in step 2. from the link *l*,
4. Monitor the QoS level on reoptimized LSP. If values of QoS parameters do not drop below the threshold level, repeat the reoptimization process again.

The monitoring of QoS level will be provided by ingress PE router for given LSP. TE server will be notified if values of QoS parameters exceed the defined thresholds and it initiates the reoptimization process. This can be achieved using IP SLA Proactive Threshold Monitoring [10] combined with SNMP traps. In 3rd step of algorithm, TE server just marks the chosen link as unusable for chosen LSP and rerouting is again performed by ingress PE router of rerouted LSP.

The design of path selection and reoptimization algorithm gives us advantage in the case that backup path has to be established because of some failure on primary path. We can rely on ingress PE router with calculation and establishment of backup path. TE server just needs to assure, that backup path meets the QoS requirements for given class. If not, reoptimization is performed or new path calculation is initiated.

3.3 Testing and experiments

Testing of TE server will be performed in laboratory environment. MPLS VPN network has to be configured as well as MPLS TE functionalities have to be enabled. Testing topologies will consist of 8 to 10 routers connected with exclusively point-to-point links. Each PE router will connect few simulated customer VPN sites. Proposed testing topology is depicted in Figure 2.

We will perform 4 experiments on each of proposed topology designs. The experiments will differ in number of requested SLAs and their parameters. During the experiments, we will monitor:

- the LSP rejection ratio,
- the average utilization of the links,
- the percentage of time, when the QoS parameters exceed required values,
- data loss percentage during the reroute of LSP or implementing backup.

In order to provide trustworthy experiments results, we need to generate the traffic for each traffic class. This will simulate the customer’s traffic flows. The maximum bandwidth requirements per

LSP in SLAs have to be adapted according to provided network bandwidth capacity – we cannot require 100 Mbps of bandwidth for one LSP if using proposed topology designs.

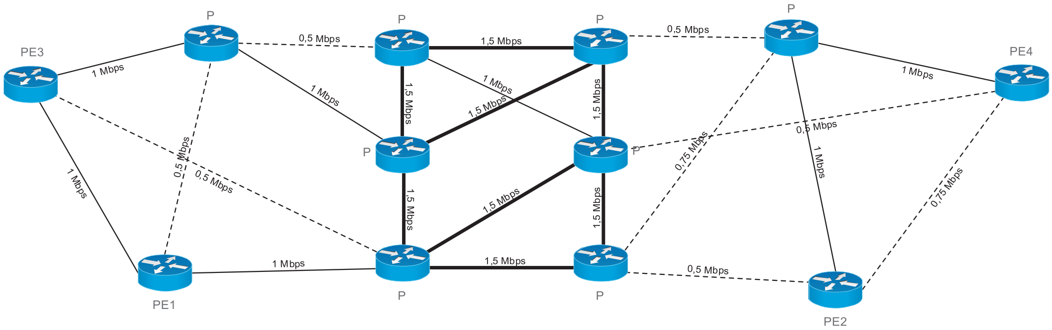


Figure 2. Testing topology.

3.4 Conclusion

In this paper we introduced new dynamic MPLS TE method which is designed to be used in present ISPs network. The proposed method takes into consideration required level of QoS for specific traffic types as well as optimal utilization of resources. Two link costs were defined to achieve mentioned goals - QoS metric and LCI. The path calculation is performed on ingress PE for VPN site of customer, TE server is only affecting path selection by updating both of proposed link costs. This design allows the distribution of calculational overhead during path selection, backup path calculation and rerouting on PE routers instead of performing all of these processes on TE server. TE server will monitor the network behavior and react when necessary.

Acknowledgement: This work was partially supported by Grant No. 1/0676/12 of the Slovak VEGA Grant Agency.

References

[1] Szigeti, T., et al.: End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs. Cisco Press, 2004, 786 p. ISBN978-1-58705-176-0

[2] Wang, N., Ho, K., Pavlou, G. et al.: *An Overview of Routing Optimization for Internet Traffic Engineering*, IEEE Communication Surveys & Tutorials, vol. 10, no. 1, 2008, 20p.

[3] Szeto, W., et al.: Dynamic Online Routing Algorithm for MPLS Traffic Engineering, Networking, 2002, 10p.

[4] Kodialam, M., et al.: Minimum Interference Routing with Applications to MPLS Traffic Engineering, IEEE INFOCOM, 2000, 10p.

[5] Scoglio, C., et al.: *TEAM: A traffic engineering automated manager for DiffServ-based MPLS networks*, IEEE Communications Magazine, vol. 42, Oct. 2004, 12p.

[6] Traffic Engineering for Quality of Service in the Internet, at Large Scale, IST Tequilahttp://www.ist-tequila.org/

[7] Lai, W., et al.: *Fast reroute with pre-established bypass tunnel in MPLS*, Computer Communications 31, 2008, 12p.

[8] Cisco Systems: *IP Service Level Agreement (IP SLA)*, White Paper, 2007, 19p.

[9] De Ghein, L.: *MPLS Fundamentals*. Cisco Press, 2006, 672 p. ISBN 978-1587051128

[10] Cisco Systems: IP SLAs Proactive Threshold Monitoring, Jul. 2008, 12p.

Efficient Design of Maximal-size Local Bitmap for Repair Algorithms for RAMs

Juraj ŠUBÍN*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
juraj.subin@gmail.com

Abstract. Built-in redundancy analysis is widely used to improve memory yield. One important feature of a class of repair algorithms for built-in redundancy analysis of RAMs is a fault bitmap. The bitmap stores, deletes and compares the stored fault data. In addition, a group of registers is usually added to the bitmap to store additional information about the data stored in the bitmap itself. Traditionally, full or compressed bitmaps were used, but recently the maximal-size local bitmap was proposed as a more efficient way to store fault information. In this paper, we propose an area and speed efficient way to design maximal-size local bitmap.

1 Introduction

According to forecast, the area occupied by memories on system-on-a-chip (SoC) designs continues to grow and will approach 70 % by 2017 [1]. SoCs are becoming memory dominant and therefore their yield is also dominated by memories. As process technology scales down into nanometer technology, memory capacity and density grows, which leads to their increased susceptibility to permanent faults. This causes memory and therefore also SoC overall yields to drop. Maintaining acceptable yields has become a very important task.

Built-in redundancy analysis (BIRA) is widely used as a solution to boost memory yield. Faulty memory cells are replaced by redundant ones according to repair assignment provided by repair algorithm (RA) embedded within BIRA. In past three decades, many RAs were proposed, e.g. [2]-[4]. One class of RA uses fault bitmaps to store fault information (FI) and to determine repair assignments. Traditionally, full or compressed [3] bitmaps were used, but recently the maximal-size local bitmap (MLB) was proposed as a more efficient way to store FI [4]. In this paper, we propose an area and speed efficient way to design MLB.

2 Fault bitmaps

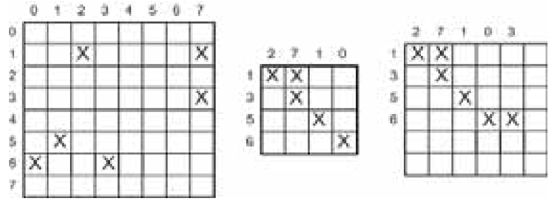
A fault bitmap (shortened as bitmap in the rest of the paper) is a memory map containing the information on the fault distribution in the memory array. Bitmaps consist of fields.

* Master degree study programme in field: Applied Informatics
Supervisor: Štefan Krištofík, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

Each field in the bitmap represents one memory cell. Based on FI stored in the bitmap, RA determines a repair assignment. There are three types of bitmaps:

1. Full. The full bitmap is of the same size as the memory. For example, 1M square bit-oriented memory block with 1024 rows and 1024 columns will be mapped to a bitmap of size 1024x1024 fields. Each field represents (stores the information about) the exact same memory cell at any time, so there is no information loss which can happen in compressed bitmaps. Full bitmaps were used by traditional software RAs implemented off-chip in automated test equipment. It is not feasible to use full bitmaps in modern BIRA since the area requirements to design those are too high [2].
2. Compressed. The compressed bitmap was proposed as a technique to reduce the area requirements of full bitmaps, making them feasible for BIRA [3]. The compressed bitmap is of a fixed, usually very small size compared to the actual size of the memory, e.g. 4x4 fields. Each field can represent various memory cells, depending on the order of fault detection. In case the bitmap is filled, i.e. no further FI can be stored in it, it needs to be emptied (partially or whole) before it can store additional FI again. Compressed bitmap size is determined by RA algorithm. If the size is set too small, the bitmap may fill up many times during repair process causing more information loss and ineffectiveness of RA. If the size is set too large, the bitmap may fill up fewer times, but area requirements increase.
3. Maximal-size local bitmap. MLB was proposed to compensate for frequent information losses in small compressed bitmaps by setting the size of the bitmap to fixed dimensions, according to (1). The size depends on the number of redundant rows (r) and columns (c) available for memory repair [4]. MLB will never be filled therefore any information loss is prevented while keeping area needed to design the bitmap relatively low.

$$MLB\_size = (\#rows) \times (\#columns) = (r.c + r) \times (r.c + c) \qquad (1)$$



(a) (b) (c)

Figure 1. Bitmap examples:(a) full, (b) compressed, (c) MLB.

Figures 1 (a), (b) and (c) show simplified examples of full, compressed and MLB bitmaps, respectively, for a memory block with 8 rows and 8 columns, 6 faults (denoted by ‘X’) detected in increasing address order and 2 redundant rows and 2 columns available for repair. It can be observed that, for example, the top left field of the full bitmap always represents the same memory cell (0,0), whereas the top left field of both the compressed bitmap and MLB can represent various memory cells. In this particular case, it is the cell (1,2).

Compressed bitmap is currently filled and cannot store the information about the 6th faulty cell (6,3). It has to be emptied before it can store additional FI, i.e. some decisions about the repair assignment for the memory have to be made according to the current information in the bitmap, which is incomplete (missing information about one fault). This type of information loss can lead to ineffectiveness of RA. On the other hand, MLB has sufficient size to contain all faults.

3 MLB description

MLB was proposed as a part of BIRA architecture in [4] and is used as an important part of RA algorithm called MLB-ORA (MLB optimal redundancy analysis). MLB-ORA comprises of two phases: the must-repair phase and the final-repair phase. The must-repair phase takes place while memory testing is in progress. When detected faults meet the must-repair condition [3], the repair is executed immediately without a need to stop the testing. Faults detected during the must-repair phase are temporarily stored in a buffer. After the repair is done, algorithm continues to process buffered faults. The final-repair phase involves an exhaustive search. During this phase all possible repair assignments are analyzed and the first one that is successful is used. FI is kept stored in the bitmap until the memory is repaired or marked as un-repairable by analyzing the contents of the bitmap registers.

MLB is a logical structure. It consists of a bitmap and several registers. These registers eliminate the need for multiple analyzers, as they can be used to analyze various repair assignments without a need to delete FI from the bitmap. There are four types of registers described below.

Address registers store row and column addresses of detected faults, as well as fault syndromes. Fault syndrome has the same bit width as the memory word. Value 1 in the syndrome denotes a faulty bit. Counters store fault counts in each row and column of the bitmap. These counts are used to check for the must-repair condition. For example, five faults in a single row with only four available spare columns would meet the must-repair condition. Strategy registers are used in the final phase of the algorithm during the exhaustive search. Their values are set according to repair strategy [4] (assignment) that is being analyzed. Value 1 denotes that spare row or column is currently used to repair the corresponding row or column of the bitmap. Valid registers are used to quickly evaluate whether the repair assignment is successful or not. Contents of these registers are determined as the outputs of certain logic functions [4]. If valid registers are in all-0 states, the repair assignment is successful. The following operations should be performed by MLB in as short time period as possible, ideally within one clock cycle:

- store FI into one bitmap field,
- set the value of one strategy register,
- store fault address to one address register,
- delete entire row or column of the bitmap,
- reset strategy registers after repair assignment analysis,
- provide information about whether the analyzed repair assignment was successful or not,
- provide the value of one address register (to write it to repair signature register, which stores the successful repair assignment),
- provide information about whether there is a fault with the same row or column address as a new detected fault (compare data stored in registers with incoming new data),
- provide information about in which row or column of the bitmap is stored the fault with the same row or column address as the new detected fault,
- provide information about the first available free position in the bitmap to store a new fault,
- provide information about meeting the must-repair condition.

4 Proposed MLB design

Figure 2 shows the proposed MLB design. *R_V* and *C_V* are row and column valid registers, respectively. *R_S* and *C_S* are row and column strategy registers, respectively. *R_counters* and

C_counters are self-explanatory. *R_AR* and *C_AR* are row and column address registers, respectively. The matrix of dimensions *m* x *n* is the bitmap.

The detailed structure of one bitmap field is depicted on the right hand side of Figure 2. D flip-flop is the key element because it indicates the detected fault. When the fault is detected, *q* output of D flip-flop is set to 1. Clock of the flip-flop is triggered by AND gate with two inputs - *adr_col* and *adr_row*. Both of these signals are outputs from the address decoder. Signal *adr_col* is active when addressing specific column of the bitmap and it is connected to all fields in the same column. Signal *adr_row* is active when addressing specific row of the bitmap and it is connected to all fields in the same row. When storing a new FI to the bitmap, only one flip-flop is addressed. Reset of the flip-flop is triggered by OR gate with two inputs – *rst_col | all* and *rst_row*. Signal *rst_col | all* is active when specific column of the bitmap is being deleted (must-repair condition for spare rows has been met) or the whole MLB including registers is being reset. This signal is connected to all fields in the same column. Signal *rst_row* is active when specific row of the bitmap is being deleted (must-repair condition for spare columns has been met) and it is connected to all fields in the same row.

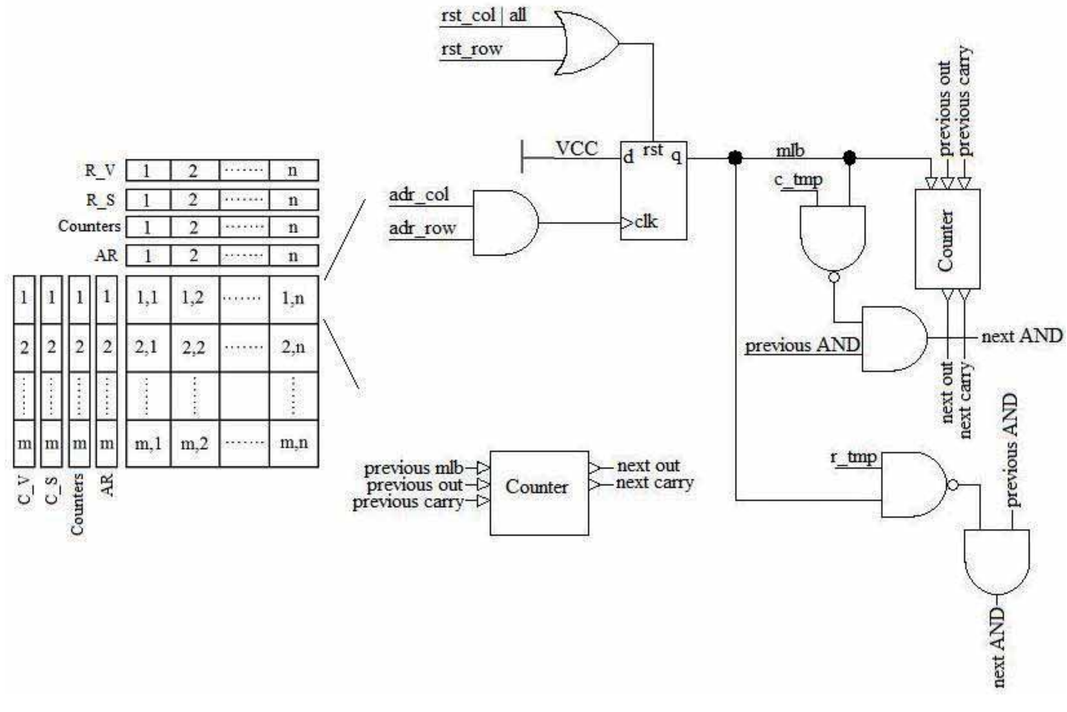


Figure 2. Proposed MLB design.

The values of row and column valid registers are determined as the outputs of logical functions (2) and (3), respectively, where $a_{i,j}$ denotes the value of the *i*th row and *j*th column of *m x n* bitmap [4].

$$R\_V[i] = \overline{R\_S[i]} \,\&\, ((a[i,1] nand \overline{C\_S[1]} \,\&\, ... \,\&\, (a[i,n] nand \overline{C\_S[n]})) \qquad (2)$$

$$C\_V[j] = \overline{C\_S[j]} \,\&\, ((a[1,j] nand \overline{R\_S[1]} \,\&\, ... \,\&\, (a[m,j] nand \overline{R\_S[m]})) \qquad (3)$$

While the memory testing is active, strategy registers are set to all-0 state. Content of valid registers is therefore generated only by the actual content of the bitmap. After testing, when the bitmap cannot be updated by detected faults anymore, the content of valid registers is generated by strategy registers. The design of these functions is shown in Figure 2.

Output from NAND gate with inputs *c_tmp* (inverted value of column strategy register) and *mlb* (1 – fault, 0 – no fault) is connected to the AND gate. The other input to this AND gate is the output from previous AND gate with the same function located in the previous column of the bitmap in the same row. These gates are concatenated through the whole bitmap, as is shown in the figure. In the last column of the bitmap there is the row valid register.

The other NAND gate (with inputs *r_tmp* and *mlb*) and the AND gate connected to its output, have similar function, but they generate content of the column valid register.

Boxes labeled as *Counter* are small logical circuits generating the counts of stored FI for each row and column of the bitmap. The design of these blocks depends on how many spare rows and columns are available. Outputs from one counter are inputs to the next counter located in the next bitmap field in the same row or column.

Address registers are designed as CAM memories in order to meet the requirement for a quick search among stored addresses.

5 Experimental results

We estimated the area requirements of the proposed MLB design in terms of transistor counts according to (4), where the symbols stand for the transistor counts of the following blocks of the proposed MLB design: F – bitmap field array, AR – address registers, V – valid registers, S – strategy registers, C – counters and AD – address decoders.

$$transistor\_count = F + AR + V + S + C + AD \tag{4}$$

Table 1 shows the transistor count of the proposed MLB design for 1 MB (1024x256x4) memory considering various amounts of spares available for repair. Table 2 shows the transistor count for 4 spare rows and columns considering various memory sizes.

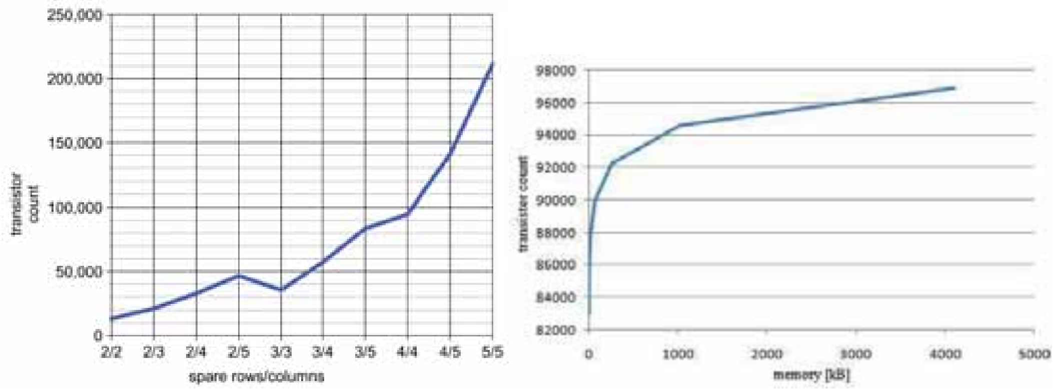
Table 1. MLB tr. count for 1 MB memory .

spare rows	spare columns	size m x n of MLB	transistor count
2	2	6 x 6	13071
2	3	8 x 9	21355
2	4	10 x 12	32995
2	5	12 x 15	46670
3	3	12 x 12	35791
3	4	15 x 16	57093
3	5	18 x 20	83587
4	4	20 x 20	94619
4	5	24 x 25	140479
5	5	30 x 30	211279

Table 2. MLB tr. count for 4 spare rows and columns.

memory size (rows*columns*word)	transistor count
32*8*4	83019
64*16*4	85339
128*32*4	87659
256*64*4	89979
512*128*4	92299
1024*256*4	94619
2048*512*4	96939

Graphical representations of the values from Tables 1 and 2 are depicted in Figure 3 (a) and (b), respectively. It can be observed from Figure 3 (a) that the MLB transistor count for a fixed memory size increases exponentially when adding more spares. With fixed number of spares, the MLB transistor count increases logarithmically in respect to memory size, as can be observed from Figure 3 (b).



(a) *(b)*

Figure 3. MLB transistor counts. (a) 1MB memory, (b) 4 spare rows and columns.

6 Conclusions and future work

We presented an area and speed efficient way to design the maximal-size local bitmap. The main goal of using this structure in BIRA architecture is to avoid having multiple repair assignment analyzers, which is area inefficient. MLB provides all the functionality needed by the control unit to perform redundancy analysis algorithm, therefore it simplifies the control plane of such architecture which may result in lower area cost. To demonstrate the functionality of the proposed MLB design, it will be used in the design of a cost-effective built-in self-repair architecture providing programmable test architecture and BIRA architecture which guarantees that the repair assignment for the memory will be found if one exists.

Acknowledgement: This work was partially supported by the Slovak Science Grant Agency under the contract no. VEGA 1/1008/12.

References

[1] Semico: System(s)-on-a-Chip A Braver New World, Semico Research Corp. (2007).
[Online; accessed February 12, 2014]. Available at:
http://www.semico.com/content/semico-systems-chip-%E2%80%93-braver-new-world

[2] Jeong, W., Kang, I., Jin, K., Kang, S.: A Fast Built-in Redundancy Analysis for Memories With Optimal Repair Rate Using a Line-Based Search Tree, IEEE Transactions on VLSI Systems, (2009), vol. 17, no. 12, pp. 1665-1678.

[3] Huang, C.-T., Wu, C.-F., Li, J.-F., Wu, C.-W.: Built-In Redundancy Analysis for Memory Yield Improvement, IEEE Transactions on Reliability, (2003), vol. 52, no. 4, pp. 386-399.

[4] Chen, T.-J., Li, J.-F., Tseng, T.-W.: Cost-Efficient Built-In Redundancy Analysis With Optimal Repair Rate for RAMs, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 6, (2012), pp. 930-940.

Synthesis of Asynchronous Sequential Circuits in High-performance Computing

František KUDLAČÁK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
fkudlacak@fiit.stuba.sk

Abstract

This paper deals with asynchronous sequential circuits synthesis based on automata representation at the logical level. A new computer-aided design (CAD) system has been developed using the automata description in VHDL format (an output format of the HDL Designer) and output is a synthesized circuit in VHDL syntax, too. Evaluation of its functionality can be done by ModelSim. The developed CAD system was implemented for a single-processor system and also for the high-performance distributed computing system. The new algorithm for composing code tables was formally described as a theoretical contribution to this topic. It allows separation of the algorithm and structures into the four main parts. Implementation for the single-processor system is written in programming language C and for the high-performance computing system also in programming language C with communication environment MPI. The service program is the third part of the system implemented in programming language C# and it interconnects all other parts of the system. Final representations of asynchronous sequential circuits were tested by the black box approach. Resulted description of automata in VHDL had the same functionality as sequential circuit described in VHDL as well. It was tested by ModelSim. Acceleration is limited by the part of algorithm which is not distributed amongst slaves. Acceleration is limited by boundary which has been defined by Amdahl law: when increasing computational nodes does not lead to increasing acceleration, due to unparallel part of program. If the number of computational nodes is increased and there are more complex circuits to compute, thus can be achieved higher acceleration, because of better unparallel code to distributed code ratio. This dependence was documented also by experimental results presented in the paper.

[*] Doctoral degree study programme in field: Applied Informatics
Supervisor: Assoc. Professor Elena Gramatová, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

Synthesis of Asynchronous Sequential Circuits in High-performance Computing

František KUDLAČÁK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
fkudlacak@fiit.stuba.sk

Abstract[1]

This paper deals with asynchronous sequential circuits synthesis based on automata representation at the logical level. A new computer-aided design (CAD) system has been developed using the automata description in VHDL format (an output format of the HDL Designer) and output is a synthesized circuit in VHDL syntax, too. Evaluation of its functionality can be done by ModelSim. The developed CAD system was implemented for a single-processor system and also for the high-performance distributed computing system. The new algorithm for composing code tables was formally described as a theoretical contribution to this topic. It allows separation of the algorithm and structures into the four main parts. Implementation for the single-processor system is written in programming language C and for the high-performance computing system also in programming language C with communication environment MPI. The service program is the third part of the system implemented in programming language C# and it interconnects all other parts of the system. Final representations of asynchronous sequential circuits were tested by the black box approach. Resulted description of automata in VHDL had the same functionality as sequential circuit described in VHDL as well. It was tested by ModelSim. Acceleration is limited by the part of algorithm which is not distributed amongst slaves. Acceleration is limited by boundary which has been defined by Amdahl law: when increasing computational nodes does not lead to increasing acceleration, due to unparallel part of program. If the number of computational nodes is increased and there are more complex circuits to compute, thus can be achieved higher acceleration, because of better unparallel code to distributed code ratio. This dependence was documented also by experimental results presented in the paper.

[*] Doctoral degree study programme in field: Applied Informatics
Supervisor: Assoc. Professor Elena Gramatová, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 423-430.

Estimation of Lithium Cell State-of-Health Using Fuzzy Logic

Ján LAŠTINEC*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
jan.lastinec@stuba.sk

Abstract. Performance decrease of the battery stack has a direct impact on the usability of an electric vehicle. The overall condition of the battery is commonly expressed in a State of Health parameter (SOH). This work deals with the design and implementation of a simple and effective method for estimating the SOH of a lithium battery cell. The proposed method uses fuzzy inference to obtain the cell's SOH based on the measurement of its direct current internal resistance and ambient temperature. The paper also presents a way how the State of Health can be further utilized to predict the remaining "lifetime" of a battery.

1 Introduction

The battery of electric vehicle (EV) generally consists of many lithium battery cells. The characteristics of the cells may vary due to e.g. internal defects or different aging rate. In order to achieve the best possible capacity and effectiveness the differences between the battery cells must be minimized. A single faulty cell decreases the performance of the whole battery stack. The battery *State of Health* (SOH) expresses the actual condition of the battery relative to a fresh one. It is a good measure that can be used for determining the probability of a cell defect and indication of imminent battery stack failure[1]. Because of the complexity of SOH parameter which depends on a number of factors, its value is often estimated using methods of artificial intelligence, machine learning or fuzzy reasoning.

In this paper a fuzzy logic approach to the battery SOH estimation is proposed. The advantage of a fuzzy logic is in the ability to simplify the modeling of relatively complex systems and its low computational requirements. This is particularly suitable for embedded systems that monitor the battery of an electric vehicle (*Battery Management Systems*).

The paper is organized as follows. Section 2 contains a description of battery characteristics and aging process, Section 3 details the proposed fuzzy model and its usage to evaluate the state and remaining cycles of the measured cell. Section 4 deals with the design of hardware that will be used to validate the correct functionality of the solution. Finally Section 5 describes simulation results.

[*] Doctoral study programme in field: Applied Informatics
Supervisor: Assoc. Professor Ladislav Hudec, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

[1] For detailed information about SOH we redirect the reader to http://www.mpoweruk.com/soh.htm.

2 Battery aging

This section describes the aging process of lithium batteries and several factors that have the biggest influence on the aging-rate. The description of the parameters that can be used to estimate a cell's State of Health as well as measurements needed for determining their values is also present.

The aging of the battery is caused by irreversible chemical reactions inside the battery that occur due to the usage, i.e. discharging and charging of the cell. The life-time is greatly influenced by the operating conditions of the battery. Operating the battery outside the limits outlined by the manufacturer (for example under-voltage, over-heating, etc.) leads to faster aging-rate.

2.1 State of health - SOH

As already mentioned, the value of State of Health expresses the performance of the battery relative to the fresh one and thus it indicates the "age" of the battery. There is no precise definition of the SOH because it is a subjective measure and the performance requirements differ with the purpose of battery.

SOH can be determined using several methods:

- *Cycle counting* - simple measure of SOH that keeps track of the number of charge/discharge cycles and compares the value with the expected cycle-life.
- *Capacity measurement* - Compares the actual capacity of fully charged cell with the capacity of a fresh piece.
- *Internal Resistance/Impedance Measurement* - the deterioration of internal cell structure increases the internal resistance and the level of increase is used to quantify the SOH.
- *Combination of parameters* - for better accuracy, several parameters may be used in estimation, applying appropriate weighting if needed.

Unlike the first two methods the internal resistance/impedance reflects the usage history of the battery cell and it is also relatively easy to measure compared to the estimation using multiple parameters. Therefore it has been chosen as a measure of SOH in the proposed model.

2.2 Internal resistance

The internal resistance (IR) determines the discharge capability of a battery and directly influences its terminal voltage. The SOH can be deduced from its value when compared to new battery[2]. Based on the method of measurement, there are two "variations" of this parameter - *direct current internal resistance (DCIR)* and *alternating current internal resistance (ACIR)*. Because of the fact that in electric vehicle a direct current is drained from the battery it is more appropriate to use the DCIR parameter.

Commonly used method for determining the DCIR is based on the *lumped parameter battery model* as described in [2]. The schematic of this model is shown in the Figure 1. The OCV refers to theoretical open-circuit voltage, $R_i + R_{ct}$ are the series resistance and charge-transfer resistance respectively and R_d and $C_d$is the diffusion resistance and capacitance. The DCIR value can be calculated by Ohm's Law from the drop of terminal voltage V_t when applying known constant discharge current I (Equation 1). The discharge process is illustrated in the Figure 2 where *Lumped resistance* is equal to DCIR.

[2] The value of internal resistance of a fresh high performance lithium battery cell is generally in the order of a few milliohms.

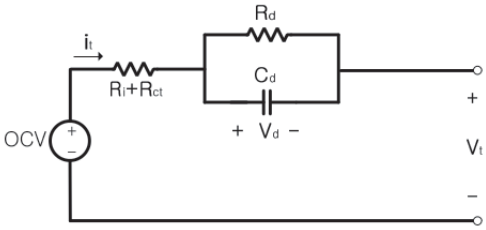


Figure 1. Lumped parameter model [2].

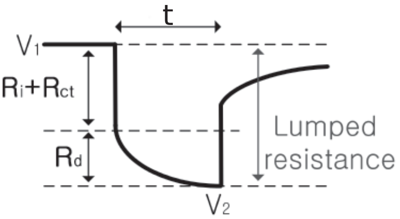


Figure 2. Discharge curve [2].

$$DCIR = \frac{V_1 - V_2}{I} \qquad (1)$$

In addition to the dependency between internal resistance and State of Health, other parameters, such as ambient temperature, *State of Charge* (amount of energy left in a battery) and discharge current, have significant influence on the internal resistance figures. In the Figure 3 values of DCIR across different temperatures disregarding the discharge current are displayed. The data are based on the technical information about several high performance lithium cells that are available on the market. Figure 4 illustrates the relationship between DCIR and discharge current and temperature at average State of Charge (limit 30%). The values were populated using the same principle as in the previous figure.

As can be seen from the displayed graphs, the DCIR value depends mainly on the actual temperature. The influence of the cell's State of Charge on DCIR is significant only when under 30 – 40%. The variance of DCIR due to different discharge currents is also negligible.

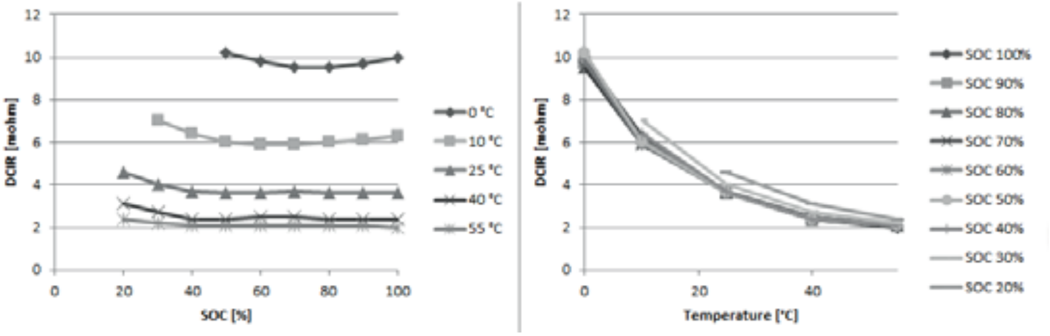


Figure 3. DCIR - temperature relationship.

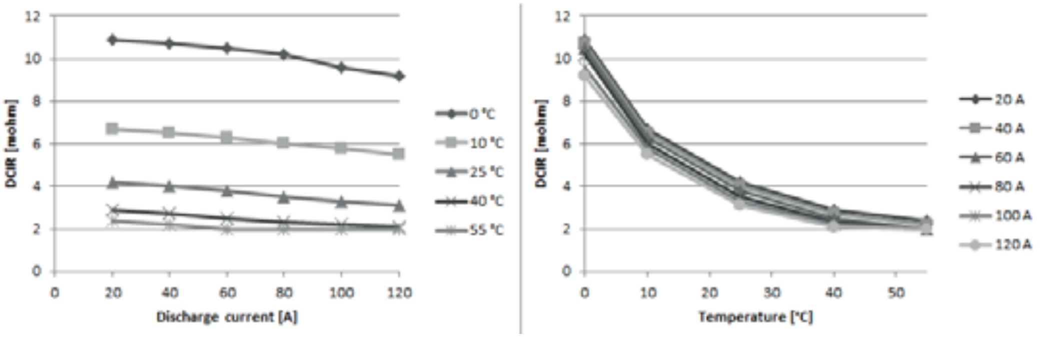


Figure 4. DCIR - current relationship.

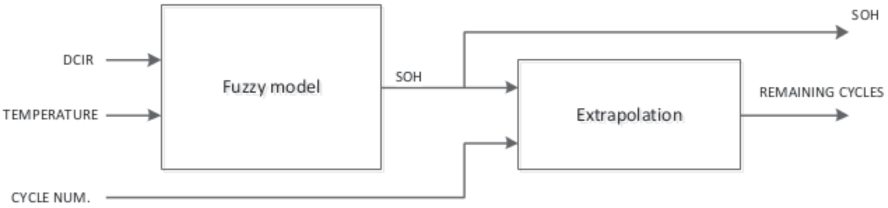


Figure 5. Algorithm schematic.

3 Fuzzy model

The visualization of relationships between internal resistance, SOC and discharge current (see previous section) reveal that the influence of SOC and discharge current can be neglected in SOH estimation without significantly affecting the accuracy. Therefore these parameters are not considered in the resulting fuzzy model.

The proposed fuzzy method has two input parameters (namely DCIR and ambient temperature) and one output (State of Health). Takagi-Sugeno-Kang (TSK) type fuzzy model[3] has been used for the SOH estimation. It had been chosen because of its potential to reduce the number of rules and also because of the time-effective weighted average defuzzification method. The estimated value is then further utilized to predict the remaining cycles of the measured cell using linear extrapolation. The block diagram of the proposed algorithm is depicted in the Figure 5.

3.1 Input

Each of the two fuzzy inputs has 5 membership functions. The values of DCIR input have been adapted from the experimental measurements of internal resistance of high performance lithium-polymer batteries conducted on an RC battery charger[4]. The membership functions of internal resistance can be seen in Figure 6 and the membership functions for temperature input are shown in Figure 7.

[3] A brief explanation of Takagi-Sugeno-Kang fuzzy model and a comparison to Mamdani model can be found at `http://www.bindichen.co.uk/post/AI/takagi-sugeno-fuzzy-model.html`.

[4] Further information on the measurements is available at `http://www.helifreak.com/showthread.php?t=233111`

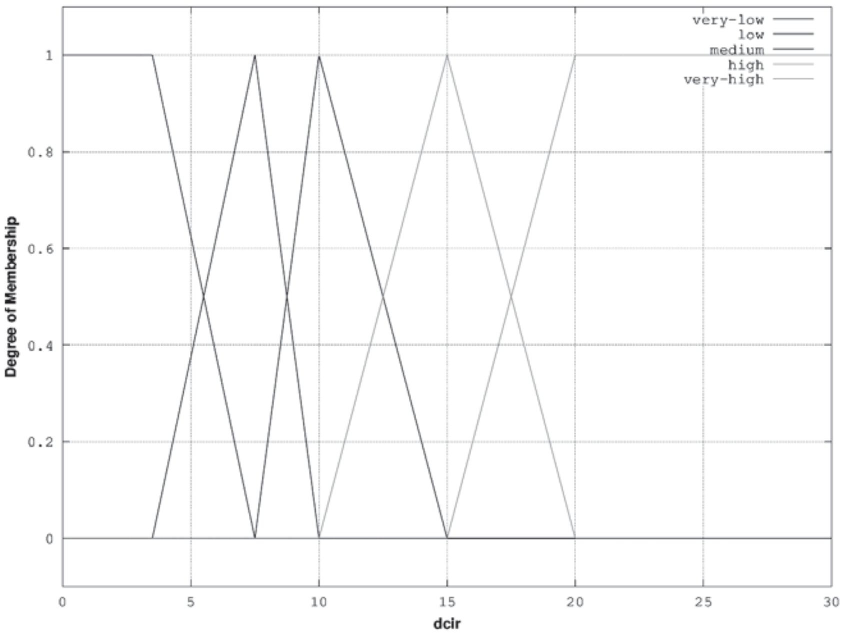


Figure 6. Fuzzy input 1 - DCIR.

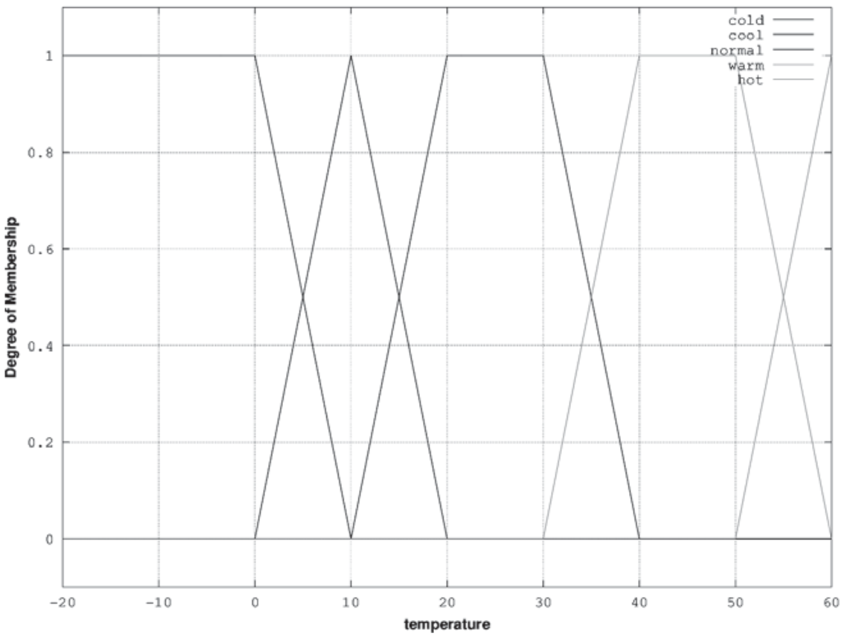


Figure 7. Fuzzy input 2 - temperature.

Table 1. Fuzzy output - SOH.

Output value	Equation
"as new"	*z = 100*
"good"	*z = 90*
"acceptable"	*z = -x - 0.5y + 100*
"poor"	*z = -x - 0.5y + 80*
"critical"	*z = -x - y + 80*

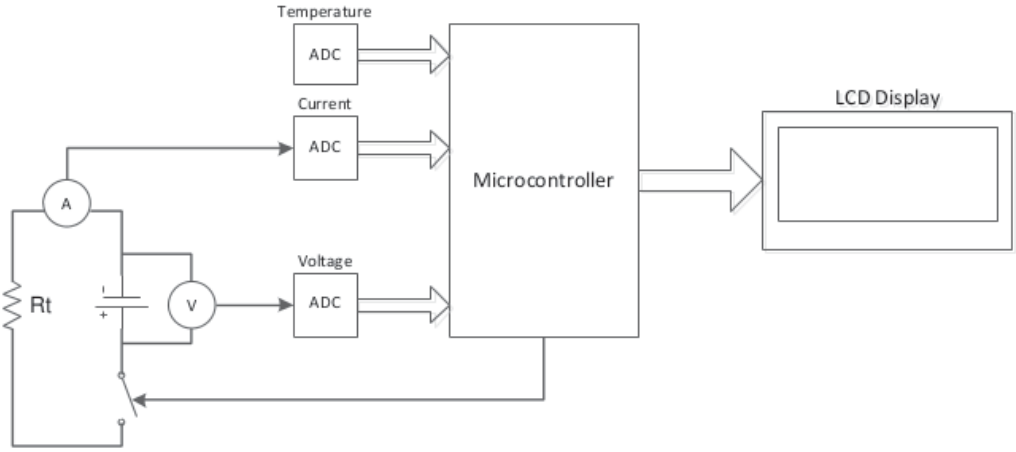


Figure 8. Hardware design.

3.2 Output and rule base

In TSK model, the fuzzy outputs typically take the form: *if x is A and y is B then z = f (x, y)*. The function *f* can be a constant (zero-order model) or a first-order polynomial function (first-order model). Some output membership functions of the designed fuzzy model are first-order polynomials to express the dependence of SOH on DCIR and temperature. The output functions are summarized in Table 1 where x, y, z correspond to DCIR, temperature and SOH values respectively.

The fuzzy model contains 17 rules in the rule database which is 32% less than an equivalent Mamdani model which needs 25 rules for the same number of membership functions per input variable.

4 Design of hardware for experiments

In this section, a design of hardware architecture for measuring the parameters that are needed for the prediction system is presented (see Figure 8). The architecture is divided into three parts. The first part consists of a test circuit for DCIR measurements and a sensor for temperature measurements. The test resistor R_t limits the discharge current flowing in the circuit and the analogue values are passed to the second part. The second part comprises of three high-precision A/D converters that convert the signals to digital form and send it to a micro-controller. The last part is a micro-controller that runs the logic described in previous section and outputs the estimates to a suitable LCD display.

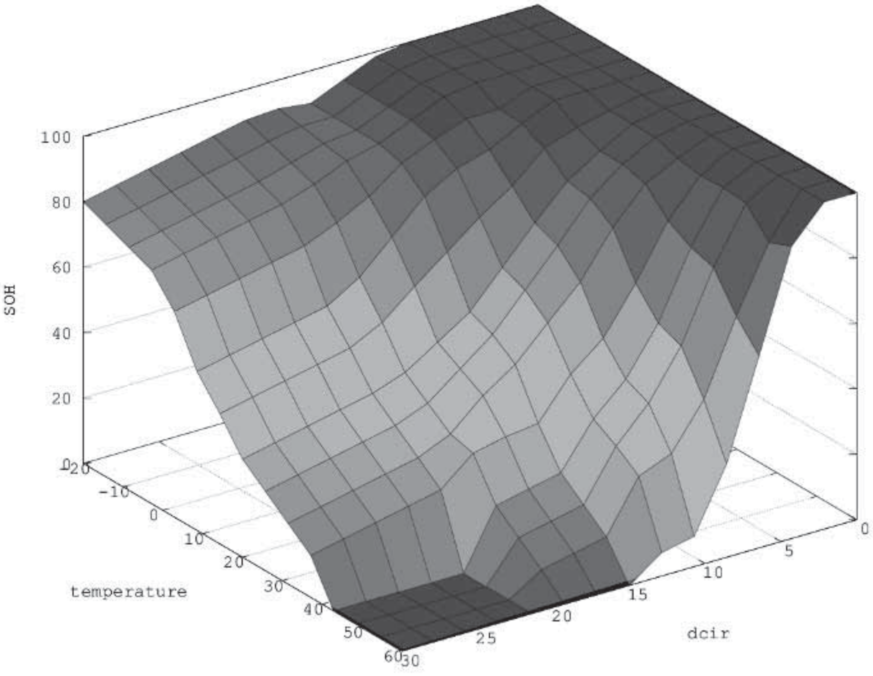


Figure 9. Model response surface.

5 Results

The proposed fuzzy model was implemented in GNU Octave [1]. Because of the long-term character of the real testing procedure[5] and also the lack of published test data concerning SOH estimation, a theoretical simulation was carried out instead. In order to study the behavior of the model a script implementing the algorithm described in Figure 5 has been developed. It reads the simulation data (DCIR, temperature and actual cycle number) from input file and estimates the state of health and expected remaining number of cycles. Table 2 lists the simulation results for selected inputs. The actual cycle number is constant here to illustrate the prediction of remaining cycles under different temperature conditions. The "zero remaining cycles" indicates that the simulated battery performance has already decreased under 80% comparing to the fresh state.

The response surface of the fuzzy model is depicted in the Figure 9 and it gives a complete overview of the estimations under all considered conditions. It can be seen that the highest SOH values are under 5 miliohms of internal resistance and the increase in DCIR causes SOH to decrease which is in accordance with the experimental results on RC models[6]. In addition, this model enhances the estimations by including the temperature dependence (internal resistance increases inversely with the ambient temperature).

[5] Standard battery cells offer cycle life of several thousand discharge cycles before reaching significant performance degradation.

[6] `http://www.helifreak.com/showthread.php?t=233111`

Table 2. Simulation results (Selected sample).

Input			Output	
DCIR [mohm]	Temp. [°C]	Act. cycle	SOH [%]	Rem. cycles
5	0	100	100	4900
10	0	100	100	4900
15	0	100	90	100
20	0	100	80	0
5	25	100	96.25	433
10	25	100	77.5	0
15	25	100	52.5	0
20	25	100	35	0
5	40	100	90.63	113
10	40	100	50	0
15	40	100	25	0
20	40	100	0	0

6 Conclusion

Degradation of battery performance greatly influences the driving performance of an electric vehicle and therefore the knowledge of the battery condition is beneficial for control systems. The fuzzy model implemented here estimates the battery State of Health and remaining number of battery cycles based on the internal resistance measurement and ambient temperature. The simulation results show that the proposed model is in accordance with the theoretical characteristics of the lithium battery cells. Thanks to the low computational requirements of the TSK model, the presented solution is suitable for use in resource constrained embedded systems such as *Raspberry Pi* or *Arduino*. The practical verification of the solution is planned in an electric race car that is being built by *STUBA Green Team* (Formula Student Electric team from Slovak University of Technology in Bratislava) for international Formula Student Electric competition. Further work will be aimed at increasing the estimation accuracy based on the experimental results.

References

[1] Eaton, J.W., Bateman, D., Hauberg, S.: *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2009, ISBN 1441413006.

[2] Kim, J.H., Lee, S.J., Lee, J.M., Cho, B.H.: A New Direct Current Internal Resistance and State of Charge Relationship for the Li-Ion Battery Pulse Power Estimation. In: *The 7th International Conference on Power Electronics*, IEEE Press, 2007, pp. 1173–1178.

Novel Power Management Unit Design

Dominik MACKO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
macko@fiit.stuba.sk

Abstract[1]

Power consumption has been a great concern in electronic systems design for a long time. As a result many advanced power-reduction techniques have been developed, such as clock gating, power gating, or dynamic voltage and frequency scaling. These advanced techniques are typically based on a power management. As we can see in Intel's processor [1], in modern systems the power management unit (PMU) is a complex circuit and therefore should also be targeted by power-efficient design techniques. We propose a novel design of self-managing PMU that can manage its own power and thus reduce the overall system power consumption.

The PMU utilizes the modified finite state machines (FSMs) in which the state logics directly corresponds to the control signals for the system power management elements, i.e. the output logics is not present in the FSM. In such a manner, the flip-flops saving the machine state can be continuously powered and thus be able to retain the control signals. The transition logics can be powered-down in inactive periods of time (the power state does not need to be changed). The isolation between transition and state logics is not needed because of the integrated controlling mechanism of the flip-flops – the data input has to be active only at the active clock edge and the clock is stopped during the idle time.

We evaluated the effectiveness of this PMU design strategy on the example, where a simple self-managing FSM has been analysed. This FSM controls only one power domain with four power states and can save approximately 70% transition logics leakage power in the idle time. Considering the complex systems have several power domains with dozens of power states, it is no surprise that the modern PMUs have more than a million of transistors. The proposed PMU design strategy alleviates the power dissipation in such complex PMUs.

Acknowledgement: This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 "Optimization of low-power design of digital and mixed integrated systems") and COST Action IC 1103 MEDIAN.

* Doctoral degree study programme in field: Applied Informatics
Supervisor: Dr. Katarína Jelemenská, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 439-446.

A Design-for-Testability Technique for Increasing Path Delay Fault Coverage

Miroslav SIEBERT*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`siebert@fiit.stuba.sk`

Abstract[1]

This paper describes a design-for-testability (DFT) technique targeted to untestable critical paths for improving quality of path delay faults testing. Path delay faults are tested via selected critical paths. But some critical paths can be found as untestable. To increase the number of testable critical paths, the circuit structure can be modified. One possibility is to use multiplexers involved inside of the circuit structure. A new technique was proposed for circuit structure modification with the goal to decrease area overhead. This approach reduces number of added logic gates used for replacement of disconnected critical paths as previously published work.

The idea of adding only basic gates into the circuit structure for increasing path delay fault coverage has gone from complexity of multiplexers designed for the same reason. The new DFT technique has been presented for increasing path delay faults coverage by transformation of untestable critical paths to testable ones. The achieved results demonstrate that 100 % fault coverage can be achieved for faults associated with critical paths even when none of these faults is testable in the original circuit. The reduction of the number of logic gates and additional inputs is evident in comparison with exist DFT techniques. In application this new DFT technique to a circuit, four logic gates, which correspond to 80 % of number of used new logic gates and one new input of circuit per every disconnected fanout branch is saved. A non-critical path can become the critical by placing this new gate *G* on it. Therefore the new gate can be placed only on non-critical path which cannot become critical after the small increase of delay.

The proposed method can be fully implemented in enhanced scan test method using skewed-load or broadside test. The future work is targeted to implementation of this DFT technique for automatic realization of path delay faults testing.

Acknowledgment: This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 "Optimization of low-power design of digital and mixed integrated systems").

[*] Doctoral degree study programme in field: Applied Informatics
Supervisor: Assoc. Professor Elena Gramatová, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 447-454.

Software Engineering

Refactoring Support Using XSLT Transformations

Lukáš MARKOVIČ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
lukass.markovic@gmail.com

Abstract. This paper presents the support for refactoring of software system source code. It analyses two main and most commonly used approaches to representation of the source code for the automatic refactoring, the representation based on abstract syntax tree and XML language. Further it describes the possibilities of source code refactoring based on the XML source code representation. Main part of the article defines the method of anti-patterns and code smells search using XPath and refactor the source code in XML representation using XSLT language. The last part describes a proposal of prototype for automated refactoring based on XML technologies.

1 Introduction

Refactoring is defined for the first time in publication of Martin Fowler as a process focused on improving the quality of source code without changing its external behaviour [4]. The main objectives of refactoring are source code parts called "code smells", which often indicates, like smell in the real world does, the deeper software system problem [4]. Besides them, there is also concept of anti-patterns, firstly used by Andrew Koenig, which represent frequently used technique, but which is in contrast to patterns undesirable [5]. Many of code smells and anti-patterns are commonly known and well described for the relative long time, for example by Martin Fowler, but nowadays there is also hundreds of new.

Because of very fast growing of software systems scopes, the research in the area of automated refactoring is very important. Many of integrated development environments has basic automated refactoring tools built yet, but so far, there is no sufficiently comprehensive tools, able to automatically improve the quality of source code with covering a lot of anti-patters and code smells. The main reason is this, that almost every anti-pattern and code smell requires a specialized approach. These approach also often requires changes right from the method of representation of source code in the process of refactoring, which in an enormous complication.

There are a plenty of works in the field of refactoring and also here in our Faculty of Informatics and Information Technologies we have created framework for the model refactoring [11] and prepared methods for the automatic identification of the code smells using

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dr. Ivan Polášek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

rule-based approach [9], aspect-oriented paradigm [8] or extended similarity scoring and bit-vector algorithms [10]. This paper briefly analyses the most common technique of code representation used during the process of refactoring, being further dedicated to the proposal of relative new technique based on XML technologies and possibilities of using it.

2 Source code representation in the refactoring process

Due to fact, that almost every anti-pattern and code smell requires its own, specific approach, the appropriate source code representation is often limiting factor during the automated refactoring of individuals problems. Nowadays, there are few integrated development environments, which have already their own, in-build, tools for automated refactoring. This tools are mostly based on abstract syntax tree, which represent very simple and clear way, how to look on source code.

The syntax tree is most common form, used for source code translation, modification and searching in source code in modern translators and integrated development environments, for example, like Eclipse [6]. The abstract syntax tree is variant of syntax tree, which unlike the concrete syntax tree does not contains all syntax details. The AST structure has many great advantages, for example, presence of the tree structure advantages. Another benefits during the AST based automated refactoring is fact, that there are standard and well debugged libraries for work with the AST, often directly, as part of the development environment. As an example of such libraries may be mentioned AST libraries, used in Eclipse. This libraries has a lot of great method for manipulation, searching and changing AST. The basis options, how to work with AST are shown in Figure 1.

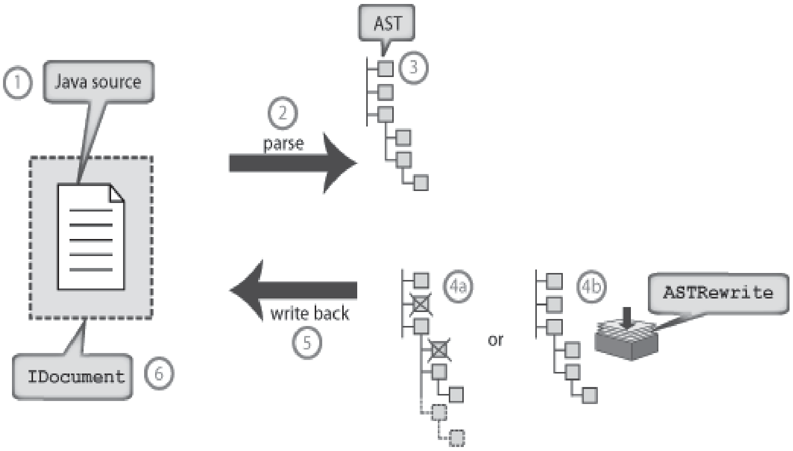


Figure 1. Typical workflow with AST in Eclipse [6].

As an opposite of abstract syntax tree, there is also possibility to represent source code using XML language [12].

XML is language designed to transmit data, regardless of the platform. XML document is, similarly to AST, based on hierarchical structure. The document contains of two parts – text and tags. The document structure is created by tags, which specified the data – the text part [1].

XML, as common way, how to clearly represents data, could be also used in the process of refactoring. This way is not yet so widespread, as AST, but has also many advantages. The main reason, why to prefer XML over AST is that, there is a lot of XML related technologies and languages, which could be used for anti-pattern and code smell search and removing. On the other hand, there is also many complications, as that, the XML, unlike the AST, is not generally used in integrated development environments to represents the source code, so the way, how to get XML source code representation is much more complicated.

2.1 XML source code representation

In contrast to AST, the XML is not commonly used for refactoring yet, therefor, there is not universal way, how to transform source code to XML representation and back. For this purpose there can be found a few freeware "convertors" on the internet. These tools are, in general, using one of two possible approaches to transformation [12]:

1. *The bottom-up approach* – in this approach is used analysis of AST. The tree is recursively searched, while the nodes are mapped to output XML document. This approached was used by few tools, for example JavaML, CppML or OOML [7].
2. *The top-down approach* – approach is very simple, The AST is not needed, for the conversion is the source code enough. The source code is browsing from top to down and every language construction is tagged. Using this approach could be also white space preserved for manipulation.

This two approaches also markedly determines options of automated refactoring, because the output XML documents are significantly differ.

The greatest advantage of XML based refactoring is possibility of using XML technologies including related languages. Before refactoring itself, there must be firstly identified anti-pattern or code smell. With source code represent using XML, this could be done very simple using XPath.

3 XML based anti-pattern and code smells search

XPath language is query language, designed for navigate and select nodes from XML document [2]. The syntax of XPath language is very simple. The language contains a set of functions and operations, used for verify, if the node satisfies the condition.

The XPath expression contains of three main parts:

1. *Node test* – define the condition, which node must satisfied,
2. *Predicate* – define the constraints, which must be truth,
3. *Axis* – define the movement based on the tree structure of XML document.

XPath expression, identifying many code smells or anti-patterns are relative simple and easy to understand. The Table 1 shows some examples of XPath expressions, which could be used for anti-pattern and code smells search.

Table 1. XPath expressions examples.

Code smell name	XPath expression
Long Method	//function[count(block//expr_stmt) + count(block//decl_stmt) > 20]
Long parameter list	//function/parameter_list[count(param) > 3]
Switch statement	//switch
Catch and ignore	//catch[count(block/child::node()) = 1]
Catch and rethrow	//try[catch/param/decl/name=catch//throw/expr/name]

4 Refactoring using XSLT

Similar to a search in XML documents, for the XML language, there exist a lot of related languages, one of them dedicated for transformations of XML documents. The XSLT language, part of XSL language family, is XML related language, serving for transformation of XML documents into new documents, which are not necessarily also XML.

Language is based on XPath, which is used by XSLT for identifying the desired nodes. The structure of XSLT is same as the structure of XML itself, with the difference that every tag has to have “xsl:” prefix. The root element also has to be specified according to the rules of language XSLT, which mean, that it must has the form “<xsl:stylesheet>” or “<xsl:transform>” [3].

Using XSLT language, is simple way, how to make a changes in XML document. Unfortunately, as opposite to XPath, XSLT is relatively complex and not so easy to learn language. Another disadvantage using XSLT for refactoring, is that, that for every anti-pattern and code smell must be written very detailed and complex script, as can be seen bellow on example of XSLT refactoring script for very simple anti-pattern called “Catch and ignore”.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

  <xsl:template match="/unit/catch/block">
    <block>{
      <expr_stmt>
        <expr>
          <call>
            <name>
              <xsl:copy-of select="/unit/catch/param/decl/name">
              </xsl:copy-of>.
              <name>printStackTrace</name>
            </name>
            <argument_list>()</argument_list>
          </call>
        </expr>;</expr_stmt>
    }</block>
  </xsl:template>

</xsl:stylesheet>
```

One of the possible ways, how to deal with complexity of an XSLT language and difficulty of creating new XSLT refactoring scripts is to design a new language, based on standard XSLT, which will provide in addition a based, simplified predefined operation, like add, remove, etc. This “Macro XSLT” could be effective way, how to use an XSLT language for refactoring of source code.

5 A design of XML based automated refactoring tool

There is a lot of ways, how to perform automated refactoring. Many of these ways are limited and able to repair only some kind of anti-patterns and code smells.

Our approach consists of using XML based source code representation as a starting point, followed by use of related XML technologies for anti-patterns and code smells search and remove. The Figure 2 shows simplified workflow, which could be used in XML based refactoring tool.

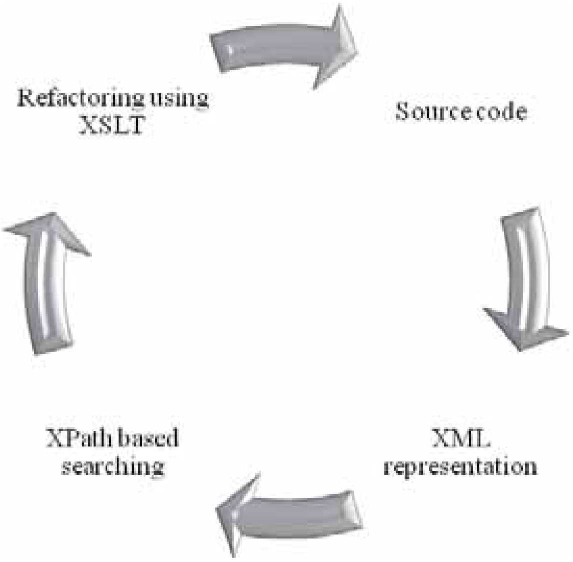


Figure 2. Main steps, need to be done in proposed refactoring tool.

Proposed tool, could be simply implemented in Java programing language. Because there is need for transforming source code into XML representation, the tool have to use an external module for transforming source code to XML representation and bact. For this module could be used one of few freeware tools available on the internet. The great choice is, for example, program SrcML, which is very quick, effective and easy to use.

For the anti-pattern and code smell identification could be effectively used XPath language. Depending on chosen tool, able to convert source code into XML representation and back, there is need to adapt presented XPath expressions for XML document.

In our approach is for the main part, the refactoring itself, used the XSLT language, able to transforming XML document. Lately, there is also possibility to use theoretically proposed “Macro XSLT” language for transformation. The output document will be then converted back into source code, and the original code part will be replaced.

Selected parts of proposed tool are being implemented now and we also proposed a few XPath expressions and XSLT transformations, capable of repairing anti-patterns and code smells, which serve for testing and tool functionality verification.

The proposed tool could also be implemented as a rule-based system designed to support users during the process of source code automated refactoring, and also at creating and editing XPath expressions and XSLT templates.

6 Conclusions

Nowadays, the most popular techniques for automated refactoring of source code are approaches based on abstract syntax tree, but using XML technologies has also great potential. The using of XPath language for anti-patterns and code smells search is easy and relative comfort way.

The most problematic part of XML based automated refactoring, are XML document transformations, using XSLT language. This part could be simplified proposing "Macro XSLT" language, which would be dedicated exclusively for use in the process of XML based source code automated refactoring.

The proposal of language dedicated for transformation of XML documents during refactoring process called "Macro XLST" represent the one possibility for future work. Another option is to design quality rule-based system aimed at support the user through the whole refactoring process.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., Yergeau, F.: *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. [Online; accessed November 26, 2008]. Available at: http://www.w3.org/TR/REC-xml

[2] Clark, J., DeRose, S.: *XML Path Language (XPath)*. [Online; accessed November 16, 1999]. Available at: http://www.w3.org/TR/xpath

[3] Clark, J.: *XSL Transformations (XSLT)*. [Online; accessed November 16, 1999]. Available at: http://www.w3.org/TR/xslt

[4] Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, (1999).

[5] Koenig, A.: Patterns and Antipatterns. In: *Journal of Object-Oriented Programming*, (1995), vol. 8, pp. 46-48.

[6] Kuhn, T., Thomann, O.: *Abstract Syntax Tree*. [Online; accessed November 20, 2006]. Available at: http://www.eclipse.org/articles/Article-JavaCodeManipulation_AST

[7] Mamas, E., Kontogiannis, K.: Towards Portable Source Code Representations Using XML, University of Waterloo, Waterloo, (2000).

[8] Pipík, R., Polášek, I.: Semi-automatic refactoring to aspect-oriented platform. In: *CINTI 2013: proceedings of the 14th IEEE International Symposium on Computational Intelligence and Informatics*, Budapest, (2013), pp. 141-145.

[9] Polášek, I., Snopko, S. Kapustík, I.: Automatic identification of the anti-patterns using the rule-based approach. In: *SISY 2012 : IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics*, Subotica (2012), pp. 283-286.

[10] Polášek, I., Líška, P., Kelemen, J., Lang, J.: On Extended Similarity Scoring and Bit-vector Algorithms for Design Smell Detection. In: *INES 2012 : IEEE 16th International Conference on Intelligent Engineering Systems Proceedings*, Lisbon, (2012), pp. 115-120.

[11] Štolc, M., Polášek, I.: A Visual based Framework for the Model Refactoring Techniques. In: *SAMI 2010, 8th IEEE International Symposium on Applied Machine Intelligence and Informatics*, Herľany, (2010), pp. 77-82.

[12] Zou, Y., Kontogiannis, K.: Towards A Portable XML-based Source Code Representation, University of Waterloo, Waterloo, (2001).

Implementing the Control Flow Pointcut in Python

Michal Bystricky[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`m@mby.sk`

Abstract. Control flow pointcut is an advanced mean of expression in a programming language, which can be used in many ways to make source code more lightweight and thus more understandable. Popular languages like Python do not implicitly or explicitly provide control flow pointcut, but some of them provide, compared to aspect oriented programming, lower level programming called metaprogramming. Therefore, aspects can be present in dynamic languages like Python. However, Python does not provide full implementation of the aspect oriented programming compared to AspectJ. We present an approach to implementation of the control flow pointcut in Python and a new library AspectPy. The approach to implementation of control flow pointcut is contained in the AspectPy library. The library also includes implementations of other pointcut types and inter-type declarations.

1 Introduction

Aspect oriented languages add additional features to object oriented programming. Users can benefit from the features during whole software development life cycle. Even in the first phase, specification of requirements, can be seen separation of concerns in form of requirements as security or logging. Moreover the aspects help with better expression of software design and design patterns. On a programmer's side this results in lightweight and more understandable source code. Later the source code is more manageable and maintainable.

One of the feature of aspect oriented programming is control flow pointcut, which we will be presenting in the work along with implementation in Python. In Python the aspect oriented programming has been developing for many years [1] and yet only a few features are present there. For example, AspectJ implements control flow pointcut types, which is not available in Python or any Python library. We can see partial implementations in Python of aspects before, but unfortunately none of them provide complex implementation of aspect like in AspectJ.

Python is dynamic language and offers means to develop aspect oriented extensions by extending itself at runtime, which is also called metaprogramming. The metaprogramming is lower level

[*] Master study programme in field: Software Engineering
Supervisor: Assoc. Professor Valentino Vranić, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

programming than the aspect oriented programming, because aspect oriented extensions can be implemented by using the metaprogramming. Unfortunately we do not know any library, that implements control flow pointcut in Python.

In this work we study approach to implementing control flow pointcut in Python. The approach was implemented, tested and included to AspectPy library.

In the second Section we define control flow pointcut and outline issues with implementing control flow pointcut specifically for Python.

The third Section is about transferring flow from one object to another object implicitly without altering original source code. The Section uncovers basic idea of implementing the control flow pointcut in Python.

In the fourth Section, we present Python specific and AspectJ inspired syntax of aspect in Python. We also compare the Python way syntax of aspect to AspectJ syntax.

The fifth Section presents a new library AspectPy—AspectJ implementation in Python. The library AspectPy includes almost all pointcut implementations, which are also contained in the AspectJ. Inter-type declarations are present in the AspectPy too. The AspectPy library is available for download [2].

2 Where Python meets control flow pointcut

The term control flow pointcut needs to be defined. AspectJ is the most advanced aspect oriented programming language, thus we will adopt AspectJ terminology and syntax in this work. Term control flow pointcut is defined in the AspectJ as following. Every join point in the control flow of each join point P picked out by Pointcut, including P itself [3].

As we mentioned before, implementing control flow pointcut as described is possible, but there are some issues that need to be solved at first. We need to know about objects and methods that were called before in a flow. These objects and methods can be called a flow list.

Secondly the flow list has to be saved in the unique place for the call. There is a difference between call and method. Method is just one and can be called multiple times, compared to call, which is unique. Thus the flow list cannot be located in the class, object or method itself. The problem is when multiple threads are accessing the same method with different flow, they would cause rewriting the flow list. The class, object or method are not unique for the call.

Finally the advice needs to be applied if the names of object and method match the regular expression pattern. Moreover the advice has to have an access to context. Context is for example the access to this object (self in Python), arguments and objects in the flow.

3 Transferring of the flow

Aspect in the AspectJ can be defined as specific mean of expression and contains pointcut and advice definition. Although Python does not have these constructions like in the AspectJ implicitly, a new definition of aspect in form of the class will have to be presented.

As was mentioned before, we need to know about the flow list. It is possible to achieve this by saving objects and methods of previously called methods and their objects in the flow.

In the AspectJ, a ThreadLocal is used to track control flow for each affected thread [3]. ThreadLocal provides thread-local variables. Thus the flow list is saved in thread-local variable.

In order to transfer the flow list from source to destination method, there has to be place, where the flow list can be saved for specific *call*. Unique identifier of call can be located in call's arguments. But using arguments it would be necessary to alter all methods in order to implement control flow pointcut and this is not acceptable.

In Python it is possible to wrap method to another method and replace the original method with a wrapped method [4]. This wrapped method is executed when the method is called. Since in Python everything is object [4], the wrapped method can be also an object, but callable. Moreover it is unique for the call, because it is created when the method is called. A programmer thinks he is calling the method, but instead he instantiates wrap object and this wrap object calls the original method or maybe can apply advice. Thus the flow list can be saved in the wrap object. This opens various questions. How to wrap only specific methods in class? How to access the object from which the method was called?

The first question can be answered by metaclasses. It is possible to alter classes after its initialization, thus all methods in the class will be wrapped. The original functionality of the method will be preserved, because the original method is only wrapped and there is no change, there is just an overhead. Then the advice can be called depending on the flow list and context in which the flow object is located.

All methods of all classes need to be wrapped. It is because if there is one method that is not wrapped, the flow list will not be transferred to destination method. So another called wrapped method in the flow does not know about previous objects and methods in th flow, because chain of wrapped methods was interrupted (none method called the destination method).

The second question can be answered by Python's `inspect` module [4]. Python provides `inspect` module through which it is possible to get object and method, which called the currently executing method.

To wrap the method with the wrap object, we will create a patch function. The patch function returns `patch_in` function, which will be called when the call occurs. The function just creates the wrap object. The example of the patch function can be seen in the following piece of source code:

```
def patch(method):
    def patch_in(self, *args, **kwargs):
        wrap = Wrap(method, self)
        return wrap(self, *args, **kwargs)
    return patch_in
```

Finally callable object wrap just gets advice from aspect and if there is any method to inject, it will apply (inject) it before, after or around. As we mentioned before the advice is applied according to the flow list and context.

4 Applying advice to control flow

Now, the last piece of the puzzle is to define aspect in Python. Python does not have constructions like aspect, pointcut or advice implicitly. Therefore, we present a new definition of an aspect:

```
class MyAspect(Aspect):

    def __init__(self):
        self.around("cflow(call(B.m1)) and not target(C)", self.advice1)

    def advice1(self, context):
        print("This will be printed before")
        result = context.proceed(*context.args, **context.kwargs)
        print("This will be printed after")
        return result
```

As can be seen the aspect is actually a class. Similar aspect definition in form of a class is present in multiple Python libraries. Pointcuts can be defined in constructor, where are mapped to methods. It is possible to use regular expressions to identify class or method names. Inherited class Aspect using metaclass initializes `MyAspect` object, which will register aspect to an `Application` object.

The Application object is singleton object and contains all pointcuts to advices mappings. Then wrap object can get advice from the Application singleton.

In the example of the aspect, the text in the advice method will be printed before every method called from object of type B and method m1 (including the method) instead of methods which target object are type C. After the text is printed we proceed execution of original method. Original method along with this object, previous objects in flow, arguments and flow list are located in a context argument of advice method.

The AspectJ provides more straightforward and more understandable syntax. There is no need to define constructor or class like in Python. Unfortunately Python does not provide means to extend its constructions to create domain specific extensions. The same example of the aspect can be written in the AspectJ as following source code:

```
public aspect MyAspect {

    void around(): cflow(call(* B.m1())) && !target(C) &&
            call(* *.*()) && !within(MyAspect) {

        System.out.println("This will be printed before");
        proceed();
        System.out.println("This will be printed after");
    }
```

The presented implementation of control flow pointcut was evaluated by multiple tests. The implementation of control flow pointcut type was tested alone and also with other pointcut types such as target, this, get, set and others. The other pointcut types can interfere with the control flow pointcut, thus it was important for example check whether variable, which we set by using set pointcut, is accessible by control flow pointcut advice and other way around.

The main factor deciding whether tests were successful, were the AspectJ results of the aspects compared to results of the implementation of the same aspects. We created same aspects with our approach and with AspectJ. The results were compared and when they matched, the test passed. The tests can be seen in AspectPy library web site [2]. We will describe the library in next Section.

5 AspectPy library

The proposed control flow pointcut in Python is a part of a bigger endeavor called AspectPy [2]. AspectPy is a new library that extends Python with the AspectJ-style aspect-oriented programming constructs. The library also contains all AspectJ pointcut types except execution, staticinitialization, adviceexecution, within and if pointcut types.

Pointcuts can be composed like in the AspectJ with and and or expressions. The negation of the pointcut is possible by not expression. The AspectPy includes inter-type declarations too.

We used metaprogramming to develop aspect oriented support in the AspectPy. The library uses wrappers and special access to object in order to prevent loops. There was no need to alter the source code by metaclasses.

The implementation of the AspectPy library has serious performance issues. Before every call, the list of pointcuts needs to be evaluated whether a pointcut match to the current call or not. Actually, the aspects are not weaved to source code, but only logic is weaved to source code. The logic then decides what aspect should be applied. The better approach would be to weave specific source code of advice into the defined classes like in the AspectJ.

The difference between the AspectPy and AspectJ weaving is that the AspectJ weaves source code of advice into the defined classes [5], thus minimum overhead is present at runtime. The AspectPy weaves only logic and a program needs to decide at runtime, which aspect will be applied. Thus there is much larger overhead compared to the AspectJ. In the case of the control flow pointcut, the logic is needed to be weaved also in the AspectJ. The logic then uses thread-local variable, that keeps track of the flow list, as we mentioned before.

6 Related work

According to Matusiak the implementation of aspect oriented extension for Python can be divided into metaclass as hook, dynamic mutation and program transformation [6]. Metaclass as hook is modification using metaclasses, e.g. Pythius library. It simply adds `__metaclass__` to original source code and it will modify the class. Dynamic mutation uses for example the AspectPy library or Logilab aspects. The source code is dynamically changed using e.g. weaver (in order to made changes using technique external dynamic mutation), which is also known as monkeypatching. External dynamic mutation is wrapper that we are using in the AspectPy described before. By using program transformations the source code can be injected (compiled) to the original source code.

Matusiak also writes about external dynamic mutation as partial reach. Meaning that not all goals can be achieved by this technique, but the AspectPy library clearly shows that the aspect oriented programming can be fully implemented using external dynamic mutation.

Cachmann, Bergmeyer and Scheriber performed analysis and according to them the Aspyct library is solid, well-designed and light-weight implementation providing AO functionality for Python developers [1] it is using monkeypatching, like the AspectPy. Unfortunately the Aspyct library provides only wrapper to functions with custom behaviour and ability to cut through modules and classes to apply these behaviours. No pointcuts are present in the Aspyct.

The SpringPython allows to define pointcut using regular expressions [7]. The advice is applied only on methods defined in this regular expression. But if we wanted to compound exact rule like in the AspectPy or the AspectJ, it would not be possible.

7 Conclusion

Although there is no presence of control flow pointcut in Python, we presented an approach to implement the control flow pointcut in Python and a new library AspectPy. Moreover the approach was implemented in the AspectPy library. The AspectPy also provides possibility of composition of pointcuts and inter-type declarations, thus a programmer can use almost full featured aspect oriented programming experience in Python.

Future work will be focused on the optimizing performance of the AspectPy library. We can see significant improvements that can be made in the library to increase performance.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Bachmann, A., Bergmeyer, H., Schreiber, A.: Evaluation of aspect-oriented frameworks in Python for extending a project with provenance documentation feature. *The Python Papers*, 2011.

[2] Bystricky, M.: AspectPy Library. `https://bitbucket.org/m\_by/aspectpy`, 2013.

[3] Colyer, A., Clement, A., Isberg, W., Kersten, M., Vasseur, A., Webster, M., Hawkins, H., Bodkin, R.: The AspectJ Programming Guide, 2002.

[4] Drake, F.L., Lundh, F.: Python v2.7.5 documentation, 2013.

[5] Hilsdale, E., Hugunin, J.: Advice Weaving in AspectJ. *AOSD Conference 2004*, 2004.

[6] Matusiak, M.: Strategies for aspect oriented programming in Python, 2009.

[7] Turnquist, G., Suchojad, D.: Spring Python documentation, 2011.

Detection of Developer's Task Context Boundaries in Programming

Tomáš CABAN[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`t.caban@gmail.com`

Abstract. In recent years, recommendation systems for software engineering have emerged to assist software developers with software development. Most of these systems extend Integrated Development Environment (IDE). Recommendations of these systems are quite limited as developer has often to define his task context and request the recommendations. Implicitly obtained context are limited, e.g. to actual working file, class or method. However, developer's task context in IDE often consists of set of actions and files. Developer also works on more tasks during some time interval. Tasks can even overlap as developer switches between them. This paper proposes method for detecting programmer's task context boundaries based on actual task context model in IDE.

1 Introduction

Software engineering is ever evolving and very challenging expert area. Developers are continually introduced to new technologies, ideas or patterns. Systems they work on grow larger in code, use more technologies and depend on more libraries. Mastering a programming language is no longer sufficient to be able to develop software systems on high level of proficiency and efficiency. Developers must also be able to navigate large code bases, class libraries and external resources such as Web resources, which are essential to complete actual task. Without help, developers can easily be flooded by large amount of irrelevant or even useless information.

We can see a parallel in other areas, where users struggle with flood of large amount of information when searching for some specific piece of information. In such cases, various recommendation systems come in, which help people find information and make decisions. These systems combine many computer science and engineering methods to make suggestions that meet users' particular information needs and preferences. To date, most recommendation systems have been deployed on web.

Recommendation systems for software engineering (RSSEs) are emerging to assist software developers in various areas of software development process such as reusing code, writing effective bug reports or navigation in code.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Tomáš Kramár, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Key factors which allowed rise of these recommendation systems include large public stores of code for recommendations analyzes and mass adoption of common software development interfaces, including Web interfaces like Bugzilla, integration development environments such as Eclipse and source control tools like SVN or Git.

Most existing RSSEs analyzed in our work are integrated into these developers' tools and assist the developers by providing recommendations considering his actual task. Here arises the problem of defining the actual task of developer to be able to provide the recommendations properly. Analyzed systems struggle with this issue in several ways. Most of existing systems require developer to define his task context by himself and request recommendations when needed. This approach is often annoying for users and they often refuse to use it. Even if developers are willing to use such system, they may not always know when they need a recommendation.

Research of recommendation systems [1] shows that future RSSEs should obtain developers' task context implicitly without any interaction with developers and recommendations should be provided proactively based on detection of current needs of developers.

There were a few RSSEs among the analyzed tools that try to obtain programmer's task context implicitly and provide recommendations proactively. However, these task contexts are quite limited, e.g. to actual working file or fragment of code. Such limited task context can be used only for limited subset of recommendations.

Typically, developers work on several tasks during some time period (day, week), they can work on them in sequence or even switch between them. Each task consists of multiple files and actions on them. Obtaining of such task context implicitly from development tools such as IDE can be quite straightforward by monitoring all developer actions in IDE. Problem arises when we consider that developers can work on several tasks in a row or even switch between multiple tasks. RSSE should be able to detect that the developer has started a new task and provide recommendations relevant for this new task.

Our paper focuses on implicit detection of developers' task boundaries based on actions in integrated development environment. We propose a method capable of detecting a start of a new task in IDE based on developers' past actions. This method is partly based on research in Web domain, where researchers detect users' session boundaries during Web search.

2 Related work on recommendation systems and task context

There have been several researches done concerning recommendation systems for software engineering. *Suade* [2] is an Eclipse plug-in providing recommendations for code navigation, where developer explicitly defines a set of relevant attributes and methods and thus creates his context. Suade evaluates static relations of code elements and provides most relevant related code elements.

Eclipse plug-in *eRose* [3] provides recommendations of files that should be changed together. If developer commits together two or more files into CVS, these files probably belong to one transaction. Later, when one of these files is modified, eRose recommends change of the other files too. Recommendations are provided proactively after each save action.

CodeBroker [4] is a recommendation system integrated into Emacs IDE, which analyzes the comments in actual Java class and builds search queries based on them. These queries are used in special search engine to search Javadoc documentation of third party libraries for components, classes or methods that could implement desired functionality.

Strathcona [5] is a recommendation system that assists developers in using frameworks and external libraries. Developer can highlight a partly finished code in Eclipse and Strathcona searches for similar examples of code usage.

All analyzed recommendation systems for software engineering use some form of actual task context of developer to provide recommendations. Definition and representation of the task context differs for each system. However, for most of these systems, we can state that the obtained task context is quite limited. Developer task often consists of more files and actions. However, most of analyzed systems consider the task context an actual file, class or even method.

Most systems also require an explicit definition of task context by developer himself, which can be annoying, time consuming and developers may not realize when they really need a recommendation.

Our paper focuses on obtaining the task context of developer implicitly, without any need to interact with the system, based on his actions in IDE. In early stage of our work, we tried to propose a general model of task context, which could be applied to provide a wide variety of recommendations like code navigation, code references search or web search.

However, we came to conclusion, that long-term task context built from all actions in IDE is not applicable for such recommendations. Developers often work on multiple tasks in a row or even simultaneously and for each task the recommendations can differ considerably. Therefore we came to conclusion, that we need to be able to detect, when the task context of developer changes.

Our paper proposes a method for implicit detecting the *task context boundaries*. Based on our research of recommendation system, we propose a method able to implicitly detect a change of task context during developers' work in IDE. We have not encountered any similar research in this area.

However, many researchers study analogical problem in Web domain, where they aim to detect users' session boundaries to optimize search engines. There have been several researches. One of first and still most used metric is a session timeout first presented by Catledge and Pitkow in 1994 [6]. Based on this work, a generally acknowledged value of 30 minutes is used as session timeout. Later, there were several researches that suggested timeout values from 5 to 120 minutes. Several researches using real data stated, that session timeouts are not sufficient to detect if queries belong to the same session. They aimed to analyze the content of queries to detect if they belong to the same session. Several approaches were proposed, like mutual words in queries [7], detection of added, deleted words in queries [8], mutual words in queries and documents retrieved by queries [9]. Our method is inspired by the research in Web domain, we adopted several approaches and attempted to combine them and apply them to the task context of developer working in IDE.

3 Task context

Task context represents the actual developers' task. As stated in section 2, most existing systems use the actual working file as task context or force the developer to explicitly define his task context. Only eRose system is trying to build task context implicitly from all files used by developers during some time intervals. However these intervals are explicitly defined and don't always correspond with actual task context.

We define the task context as set of all files developers use and work with and propose a method able to detect when the task context changes. This method should ensure that all files used to generate recommendations are relevant to the actual task.

For purpose of our method, we use extended task context, which contains all actions executed in IDE by developers and all files associated with these actions. Extended task context is essential in order to detect when the task context changes.

However, detecting of task context boundaries is only first step in obtaining an accurate and useful actual task context. For purpose of recommendations we propose a task context as ordered list of files, where files are ordered based on their relevancy (working time, display time, number of returns to file, etc.).

4 Method for detecting task context boundaries

Our proposed method evaluates each action performed by developer in IDE and states if this action belongs to actual task context or if this action is a start of a new task. It is based on task context model, which is created from all actions performed in IDE and updated after each new action. This task context model is then used for each action to compute a difference between the action and actual task context.

Difference between action and actual task context considers the specifics of object oriented software development:

- Tasks are often localized to one or more packages, which have some relations between them
- Various tasks require a specific set of actions in order to complete the task, e.g.:
  - New functionality requires adding packages, files, adding large amount of new code
  - Refactoring requires replacing and removing files, packages, changing code
  - Bug fixes require code review, adding debugging code (traces, logs)

We consider all these specifics during creating the actual task context and computing the difference between performed action and actual task context.

4.1 Task context model

We proposed a task context model, which contains information about activities of developers during their work in IDE. We defined a closed set of actions, which are used to build the task context and can also represent the boundary of task context. These actions include:

- Actions with files (add, move, delete, rename, open, switch to, close)
- Actions with packages (add, delete, rename)
- Actions with source control tool (commit, check out)

For each action, we retrieve a set of features and compare it with the actual task context. If the action is classified to belong to the actual task context, we update the task context with the set of obtained features for performed action. Actual task context is represented as a vector of features (currently 23).

This vector represents the actual task of developer and is updated with each performed action that is evaluated as part of actual task context. In case the performed action is classified as a start of new task, building of task context is restarted and this action is the first action of a new task context.

4.2 Difference between actual task context and action

After one of the defined actions is performed by a developer, the first step of our method is to compute a difference vector of performed action and actual task context. This vector contains several features that represent how much the action differs from the actual task. The difference vector contains currently 18 features.

4.3 Evaluation of action

Our method uses machine learning methods for classification of difference vector described in section 4.3 to one of two pre-defined classes: *is* or *is not* task context change. This problem can be expressed formally as function (see Equation 1).

$$f(x): \{a_1, a_2, a_3, \dots, a_n\} \rightarrow \{0,1\} \qquad (1)$$

Where $a_1 - a_n$ are features of the difference vector and {0,1} are two pre-defined classes. In order to create and train model capable of classifications of developers' actions to one of the classes, dataset containing information about this class is required. We use a logistic regression machine learning method in our first experiments.

5 Experiment

In order to evaluate our proposed method, we implemented an Eclipse plug-in, which monitors all actions performed in Eclipse IDE, builds task context based on model described in chapter 4.1, computes the difference vector for each performed action (chapter 4.2) and collects explicit feedback from developers concerning the changes of their task context.

In first stage of our research, we conducted an experiment on one developer, who worked during several weeks on several Java projects of different size and focus. We collected 815 actions, out of which 79 represented a task context change.

Using WEKA [10] tool and logistic regression method, we created a classifier for the action difference vector. We used two methods of evaluating the classifier:

1. N-fold cross-validation
2. Split the dataset into testing and training sets (80:20)

Table 1 shows obtained results of classifier evaluation using n-fold cross-validation. We experimented with parameters of classifier. Results differences are due to small dataset statistically insignificant.

Table 1. N-fold cross-validation classifier evaluation.

	Success rate	Precision	Recall
1	90,87%	0,68	0,34
2	90,76%	0,67	0,34
3	90,91%	0,67	0,34
AVG	90,85%	0.67	0,34

Table 2 shows results obtained by evaluating the classifier using training and testing sets. Table shows 3 runs of evaluation as experiments with different parameters were conducted. Deviations in results are statistically insignificant due to small dataset.

Table 2. Training and testing sets classifier evaluation.

	Success rate	Precision	Recall
1	90,80%	0,66	0,36
2	90,66%	0,65	0,33
3	90,71%	0,66	0,34
AVG	90,72%	0,657	0.343

Our first experiments show quite promising results. However, we cannot state that proposed method achieves good results until we collect more data and create a more representative dataset. We also plan to apply other methods used in Web domain to our dataset so we can evaluate the proposed method in comparison with other methods used to solve this problem.

6 Conclusions and future work

In this paper we presented recommendation systems for software engineering. Our research in this are showed that despite their great potential of becoming a part of standard tools of software developers, most of existing RSSEs are limited in obtaining the task context of developers and most rely entirely on developer to define his task context used for recommendations generation.

Based on this conclusion, we presented a method for detecting developers task context boundaries during their work in IDE. This method is motivated on research in Web domain concerning detecting of users' session boundaries during Web search. Proposed method is based on task context model built from actions performed in IDE and computing of difference vector for each performed action and actual task context.

Our future work involves first of all distribution of our logging system to more Java developers and collecting data.

With sufficient dataset, the next step is to improve classifier for actions difference vectors. We plan to use and evaluate several machine learning methods.

At last, we plan to apply several metrics used in Web domain for session boundaries detection (session timeout, mean session length) to our dataset and compare results of these metrics with our proposed method.

Acknowledgement: This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] Robillard, M.P.; Walker, R.J.; Zimmermann, T., "Recommendation Systems for Software Engineering," *Software, IEEE* , vol.27, no.4, pp.80,86, July-Aug. 2010

[2] Robillard, M.P., "Topology Analysis of Software Dependencies," *ACM Trans. Software Eng. and Methodology*, vol. 17, no. 4, 2008, article no. 18.

[3] Zimmermann, T.; et al., "Mining Version Histories to Guide Software Changes," *IEEE Trans. Software Eng*., vol. 31, no. 6, 2005, pp. 429–445..

[4] Ye, Y. and Fischer, G. "Reuse-Conducive Development Environments," *Automated Software Eng.*, vol. 12, no. 2, 2005, pp. 199–235.

[5] Holmes, R.; Walker, R.J. and Murphy, G.C. , "Approximate Structural Context Matching: An Approach for Recommending Relevant Examples," *IEEE Trans. Software Eng.*, vol. 32, no. 1, 2006, pp. 952–970.

[6] Pitkow, L.; Catledge, J., "Characterizing browsing". Proceedings of the Third International World-Wide Web Conference on Technology, tools and applications. 1995.

[7] Jansen, B. J.; Spink, A.; Blakely, C. and Koshman, S., "Defining a session on web search engines." Proceedings of International Joint Conference on Artificial Intelligence. 2000.

[8] He, D.; Goker, A. and Harper, D. J., "Combining evidence for automatic web session identification." Information Processing and Management. 2002, s. 727-742.

[9] Joachims, T.; Radlinski, F., "Query chains: learning to rank from implicit feedback." In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD '05). ACM, New York, NY, USA, 239-248.

[10] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B. Reutemann, P.; Witten, I. H., "The WEKA Data Mining Software: An Update." SIGKDD Explorations. 11, 2009.

Identifying Hidden Source Code Dependencies from Developer's Activity

Martin Konôpka*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xkonopkam@stuba.sk

Abstract. Traditional software metrics evaluate software product as a result of development process, we use them to identify problems in source code while not taking into account source of identified problems. Software and its source code is the result of developer's work and so its attributes depend on the developer's activity and context during the development process. In this paper we use developer's activity and interactions with source code files during development to identify hidden implicit dependencies in the source code. In our method we extend dependency graph of software components with implicit dependencies, which can be used for navigation in source code. We identify application of our method in the phase of source code development and review.

1 Introduction

Software project comprises of many phases with their own inputs and outputs which are important to monitor. The main output of software project is the resulting software product. Development of this product requires developers to understand requirements, design and work with source code. We may evaluate various attributes of the resulting software which are meaningful to both users and developers. Many software metrics have been proposed to evaluate these attributes, mostly based on syntactic analysis of the source code [7].

Developers of software project work on various tasks during the development process. Both types, i.e., adding new functionality and changing existing functionality, require developers to use or perform changes in dependencies in source code, e.g., existing structures, compositions or inheritance hierarchies. It is helpful for developers to know about these dependencies before making the change because it is the way to learn about how the existing source code works and how it is organized. As a dependency we understand oriented connection between two source code components of selected granularity, e.g., reference, inheritance or call.

Traditionally, dependencies reflect explicit statements in source code and are identified with syntactic analysis. Identified dependencies form dependency graph of software components which helps developers in study of software components and their attributes, in identifying problematic places and complexity of the web of software components.

[*] Master degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 474–479.

Developers' activities and their work environment affect resulting attributes of created source code [6]. We may look for connections between the attributes of software [3] and activities done

by the developers together with the context which they reside in. In our work we propose method for identification of dependencies between software components from developer's activity to broaden the space of dependencies for analysis in dependency graph.

This paper is structured as follows: in section 2 we discuss related work in our research area, in section 3 we introduce implicit dependencies in software source code, following with section 4 about integrating existing dependency graph with implicit dependencies and concluding with section 5 discussing application of our method, describing its evaluation and identifying further work.

2 Related work

Tools for navigation and techniques for discovery of dependencies in the source code are of interest in research because of the various scenarios developers may encounter during the development,

e.g., disruption of work, switching between multiple tasks to complete or taking over task after different developer. Most common tools are software product metrics [7] and dependency graph of software components [5]. Dependency graph can be used not only for studying the source code, but also for code smells detection [5]. In [10] authors applied network algorithms on dependency graph to predict problems in software design. Differently, in [4] authors propose method for inferring programming sessions from monitored activities during the development. Identified sessions are then used to describe tasks which developers had worked on.

In [1] authors shown that developers visit places in source code relevant for the task completion more often during the work on that task. Information about developers' tasks can then be used to recommend developers for next tasks, based on their knowledge about the source code parts [8]. Authors in [9] proposed method for identification of usage contexts from interactions with development environment. When compared to our method, they create and provide contexts to the developer to speed up navigation during development. Because of that, identification of relevant source code places is helpful for recommendation in source code search or for studying the solutions in the existing code base.

3 Dependencies between software components

Measuring attributes of software is of high interest in software engineering. Several software metrics have been proposed to measure the attributes in software development, e.g., complexity, modularity [3,6]. Additionally to the evaluation perspective of finished development, software metrics are also applicable during the development for developers. Example situation is identification of complex places in the source code by looking for dependencies of software component and how much possible change in it would affect the rest of the code.

Dependency between two software components is *oriented connection* of selected type, traditionally reference, call, inheritance or hierarchy membership [5]. These types of connections are explicitly stated in the source code, so we understand them as explicit dependencies. Explicit dependencies are identified from the contents of source code and are visualized in form of dependency matrix or oriented graph of vertices for software components (of selected granularity) and edges for explicit dependencies between components. Developers and code reviewers may use this graph to navigate in the source code space to understand existing connections.

3.1 Implicit dependencies in source code

Given the graph of explicit dependencies in source code we are not able to infer whether the developer got inspiration from other component during the development, whether the source code was copy-pasted or whether it follows some kind of good-practice solution for particular problem used elsewhere in the source code. At the same time, developers move in the source code space during the development, read it, edit it, they think about solutions for problems and bugs. This motivates us to identify implicit dependencies to underline connections in source code which are not explicitly stated, but still present in developer's activity, intents and decisions during development, or even during the runtime of the software.

Identification process of implicit dependencies relies on user's activities which we are able to record during the development process. In our work, we use these low-level logs of user activities with source code files:

- *Open*, *close* and *switch-to* operations on source code file – time-based activities of navigation in source code space of software project, result is change in currently opened file.
- *Copy-paste* content from one source code file into another.
- *Check-in* (or *commit*) a collection of source code files to source code revision control system.

These activities are recorded during the work within integrated development environment, e.g., Microsoft Visual Studio or Eclipse, with custom extension [2]. In case of *check-in* activities, we gather collections of files with wrapper around used revision control system [2], e.g., Microsoft Team Foundation Server or Git.

3.2 Identification of implicit dependencies from activity logs

Time-based activities are described with operation on source code file, file identifier, and timestamp when the activity occurred. When the developer switches to different file, we do not know the source and target of the *switch-to* operation, only the target file. Because of that, we use algorithm with state machine to emit new dependencies when switching between states (see Figure 1).

We distinguish between state when developer has opened file and state when we are not aware of the current file (unknown or none or none).

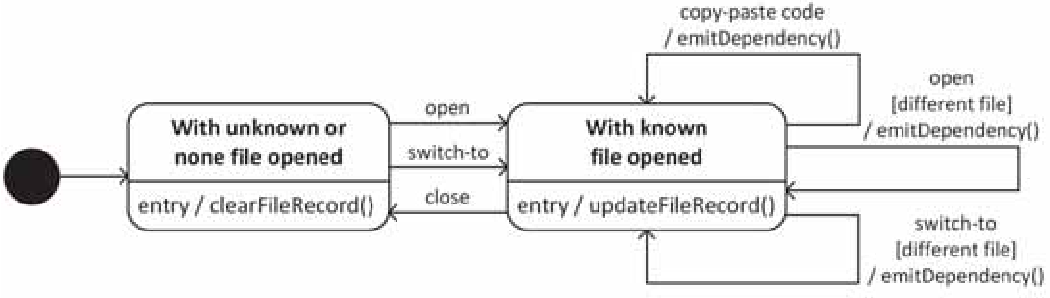


Figure 1. State machine of currently opened file in development environment used by algorithm for identification of implicit dependencies from activity logs.

We are aware of few downsides of our algorithm, although we see it as only option how to model developer's interactions with source code files because of the nature of activity logs. As an example of situation which we are not able to identify is when developer was viewing two source code files at once, i.e., docked side by side.

3.3 Weighing implicit dependencies

Explicit dependencies are weighed according to the number of existing connections of selected types. Weights describe significance of identified connections in the source code, so implicit dependencies have to be weighed as well, but differently according to the source which they were identified from.

Implicit dependencies identified from time-based activities describe developer's navigation in the source code space, when developer changes currently opened source code file (with open or switch-to action). These dependencies are weighed according to the time spent in opened file, i.e., significance of opening that file for developer on the closed interval from *0* to *1*.

Weight of dependencies identified from copy-pasting may correspond to amount of code copied. However, we chose to weigh every copy-paste operation with constant significance of *1*. Dependencies between files from one check-in action are identified between each pair of files in collection and are proportionally weighed, always summing to constant of *1*.

4 Dependency graph

Dependency graph with implicit dependencies is constructed with aggregation of identified dependencies into edges. We enhance definition of graph *G(V,E)* by adding new set of implicit edges E_{imp} to the explicit ones E_{exp}, so the set of edges equals to:

$$E = E_{exp} \cup E_{imp}$$

When constructing implicit edge for pair of selected components, we determine its weight by aggregating tuples *(timestamp, weight)* of all identified dependencies between these two components into single weighted edge. We identified these two possible versions of determining the weight:

- Sum of weights of dependencies regardless of timestamps,
- Sum of weights validated to selected point in time, i.e., timestamp.

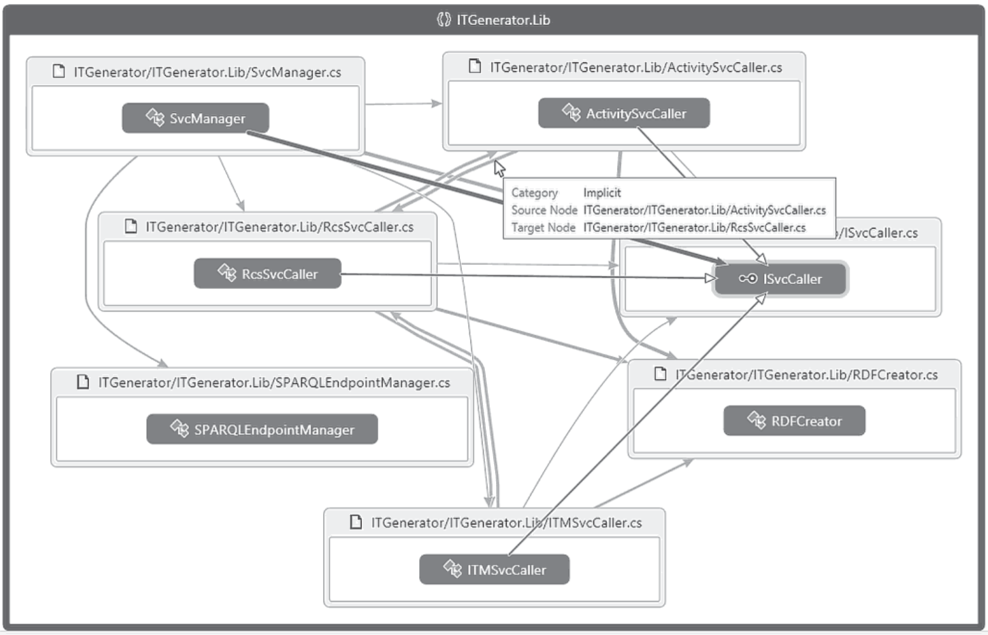
Construction of explicit edges is based on aggregation of existing single explicit dependencies in the source code relevant to the selected time. It is important to differentiate between these two types of edges in the graph, not to aggregate them, because of the difference in their meaning:

- Explicit edges describe real connections in the source code, e.g., inheritance, references, calls.
- Implicit edges describe how developer interacted with source code during the development.

Implicit edges may reflect explicit edges in the graph, e.g., in scenario when developer references other class during writing his method, switches to the source code of that class, studies it and possibly implements the functionality. However, note that developers with knowledge about the source code do not need to study every class they reference to.

See Figure 2 for example of resulting dependency graph with implicit dependencies in environment of Microsoft Visual Studio. Existing *Code Map* functionality[1] provides generation of graph with explicit dependencies which can be later changed using its source file. We introduce implicit dependencies identified with our method into that source file, add new nodes for source code files and map their contained classes and interfaces. The result is that developers may interact with dependency graph directly in Microsoft Visual Studio, switch to the real source code files from the graph and see their contents. While this integration of our method with existing tool is very effective, it is currently available for projects in *C#* programming language only.

[1] Map dependencies in specific code using code maps in Visual Studio, Microsoft Developer Network, `http://msdn.microsoft.com/en-us/library/jj739835.aspx`


ITGenerator.Lib
ITGenerator/ITGenerator.Lib/SvcManager.cs
SvcManager
ITGenerator/ITGenerator.Lib/ActivitySvcCaller.cs
ActivitySvcCaller
ITGenerator/ITGenerator.Lib/RcsSvcCaller.cs
RcsSvcCaller
/ISvcCaller.cs
ISvcCaller
ITGenerator/ITGenerator.Lib/SPARQLEndpointManager.cs
SPARQLEndpointManager
ITGenerator/ITGenerator.Lib/RDFCreator.cs
RDFCreator
ITGenerator/ITGenerator.Lib/ITMSvcCaller.cs
ITMSvcCaller

Category Implicit
Source Node ITGenerator/ITGenerator.Lib/ActivitySvcCaller.cs
Target Node ITGenerator/ITGenerator.Lib/RcsSvcCaller.cs

Figure 2. Visualization of dependency graph with explicit and implicit edges in Microsoft Visual Studio.

5 Conclusions and discussion

In this paper we introduced our method for identification of implicit dependencies in source code from developer's activity to enhance existing dependency graph, as we see its application in processes of development, code review or search. During the software development, developers may be affected by their context, e.g., personal, their knowledge, environment or context of tasks which they were working on [6]. Dependency graph with implicit edges may be used to discover relevant places in source code for particular task. We identify these scenarios to illustrate task context:

- Developer returning to work on his own source code searches for components which were relevant to the previous task. This includes ability to find relevant components which are loosely coupled or possibly not explicitly connected in the source code, e.g., when connections are defined in configuration files.
- Developer taking over another developer's task, or fixing bug, searches for related components that may be source of problem to be solved or may be affected by the solution.
- Senior developer searching for bugs and spread of problems across the source code. If we assume that developers were influenced by their context or environment during the development, we may discover bugs in time-related components during development as well as just locating one single problem.

Our work is part of research project PerConIK[2] – Personalized Conveying of Information and Knowledge, which applies Web engineering methods in software development domain [2]. We use PerConIK for provided tools and services, as well as a source for experimental evaluation of our method while it provides logs of activities on number of students' individual or team projects.

[2] PerConIK, `http://perconik.fiit.stuba.sk/`

To evaluate our method, we introductorily compared sets of explicit edges with identified implicit edges for selected projects. For 2 individual and 2 team projects we achieved average *55.7%* precision on of implicit dependencies (while not considering their orientation) reflecting explicit ones. Secondly, in evaluation of our method we look onto subset of implicit dependencies that were not matched with explicit ones whether they describe real hidden dependencies in the source code. This involves manual run through identified dependencies by developers of the selected projects.

Acknowledgement: This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] Antunes, B., Cordeiro, J., Gomez, P.: An Approach to Context-based Recommendation in Software Development. In: *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys '12)*, ACM, (2012), pp. 171-178.

[2] Bieliková, M., Polášek, I., Barla, M., et al.: Platform Independent Software Development Monitoring: Design of an Architecture. In: *Proceedings of the 40th International Conference on Current Threads in Theory and Practice of Computer Society (SOFSEM '14)*, LNCS 8327, Springer Verlaag, (2014), pp. 126-137.

[3] Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative Evaluation of Software Quality. In: *Proceedings of the 2nd International Conference on Software Engineering (ICSE'06)*, IEEE Computer Society Press, (1976), pp. 592-605.

[4] Coman, I.D., Sillitti, A.: Automated Identification of Tasks in Development Sessions. In: *Proceedings of the 16th IEEE International Conference on Program Comprehension (ICPC'08),* IEEE Computer Society Press, (2008), pp. 212-217.

[5] Counsell, S., Hassoun, Y, Loizou, G, et al.: Common Refactorings, a Dependency Graph and some Code Smells: An Empirical Study of Java OSS. In: *Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering (ISESE '06)*, ACM, (2006), pp. 288-296.

[6] Eyolfson, J., Tan, L., Lam, P.: Do Time of Day and Developer Experience Affect Commit Bugginess?. In: *Proceedings of the 8th Working Conference on Mining Software Repositories (MSR '11)*, ACM, (2011), pp. 153-162.

[7] Fenton, N.E., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach. 2nd Edition, PWS Pub. Co., Boston, MA, USA, (1998).

[8] Fritz, T., Murphy, G.C., Hill, E.: Does a Programmer's Activity Indicate Knowledge of Code?. In: *Proceedings of 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT symposium on The Foundations of Software Engineering (ESEC-FSE '07)*, ACM, (2007), pp. 341-350.

[9] Robillard, M.P., Murphy, G.C.: Automatically Inferring Concern Code from Program Investigation Activities. In: *Proceedings of 18th IEEE International Conference on Automated Software Engineering*, IEEE Computer Society Press, (2003), pp. 225-234.

[10] Zimmermann, T., Nagappan, N.: Predicting Defects Using Network Analysis on Dependency Graphs. In: *Proceedings of 30th International Conference on Software Engineering (ICSE '08)*, ACM, (2008), pp. 531-540.

Aspect-oriented Solution to Platform-specific Implementations in Cross-Platform Application Development

Martin KONÔPKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xkonopkam@stuba.sk

Abstract. In hand with the increasing number of smartphones and tablets running on different platforms grows the effort required to target multiple platforms while keeping the resources required for development low. Several solutions have been proposed to lower this effort, mostly frameworks able to target multiple platforms. We see solution for this problem with proper architecture design and selection of design patterns, although still with some limitations. In this paper we propose aspect-oriented solution to extend the shared components with platform-specific implementation while keeping them intact for other platforms. We show our ideas on example application using the Model-View-ViewModel pattern, compare it to the existing solutions but still identify further work because our concept is not yet possible to realize.

1 Introduction

In recent years we register growth in penetration of smartphones and tablets which comes in hand with development of native mobile applications [3]. Even though the Web and its technologies may head to eliminate the need for development of native applications and seamlessly bridge differences in various platforms, it is still required to develop separate native applications to cover each targeted platform and keep consistent user experience across the platform. If we want to target multiple platforms with the same or similar applications, just to reach most users because of the market share of the platforms, we have two options how to develop them. First, more resource demanding approach is to develop separate applications of the same design and functionality but with different platform-specific technologies (application programming interface, integrated development environment, toolkits, etc.) [7]. Second and more prolific approach is to develop applications using technologies that allow us to reuse components (at various levels) and thus allow us to apply key principles of software engineering in development, such as modularization, reusability, or *do not repeat yourself* principle [1].

[*] Master degree study programme in field: Software Engineering
Supervisor: Assoc. Professor Valentino Vranić, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 480–485.

2 Cross-platform application development

The most popular mobile platforms currently are Android, iOS, Windows Phone and Windows 8 [2]. Development of cross-platform applications attracts developers mostly because of the benefits of vast code reuse and faster development process with goal to cover multiple platforms. The main goal is to separate selected portions of functionality to the shared part which is then reused in all platform-specific implementations. Because of the differences in user interface layer and other platform-specific technologies (mostly platform APIs, programming language and frameworks) [6], we are not able to write and run totally the same application on two different platforms. Cross-platform development is then possible at these levels of source code reuse:

- Having multiple copies of the same source code file for each platform,
- Linking to the same source code files and using compiler directives to differentiate between platform-specific parts,
- Linking to the shared platform-independent library.

The most advantageous approach seems to be on the component level and thus to create platform-independent libraries, e.g., for the business logic with generic API. The component is monolithic, thus interchangeable and testable. In spite of these advantages (separating business or even the presentation logic), high decomposition often leads to more complex architecture design.

3 Selection of Model-View-ViewModel as architectural pattern

Several architectural patterns exist to separate the application into the layers based on their functionality and responsibilities. Common approach is to separate the View (for user interface logic) and the Model (application and business logic) with following architectural patterns [2]:

- Model-View-Controller,
- Model-View-Presenter,
- Model-View-ViewModel.

Separating the Model from View and reusing it may be sufficient in certain situations. However, if we develop the same application for multiple platforms, they do differ only in details, in platform-specific user interface design, but provided functionality to the user through the user interface remains the same. To avoid repetitive development of the UI logic (not the layout because of the platform-related differences), the Model-View-ViewModel [6] pattern was introduced which separates user interface into two parts:

- Layout – the View layer, defined in XAML language allowing loose coupling with source code through declarative bindings, it specifies UI components and their layout.
- Logic – the ViewModel layer, notifying about changes in data and providing actions on data through implementations of Command pattern to be initiated from the View layer.

Model-View-ViewModel (MVVM) is mostly used on platforms running .NET framework, e.g., Windows 8 and Windows Phone, though it may be used on Android and iOS as well. With MVVM pattern we are able to separate the ViewModel layer from View to the shared components, thus reuse even more source code than if we separated the Model layer only with other architecture patterns.

The MVVM pattern provides separation of the layers into independent components (code libraries) and thus to swap them for stub implementations when unit testing or real-time UI designing. To wire all the components we apply *Dependency Injection* and *Inversion of Control* patterns [2]. With combining all mentioned patterns we apply some kind of *brute force* which increases complexity of even the simplest applications, though with mentioned advantages.

4 Problems of cross-platform application development

Model-View-ViewModel gives us ability to separate Model and ViewModel layers and create only the View layer per platform (Figure 1). Unfortunately, this is not very common because the Model and ViewModel layers may also rely on platform-specific features. This ends in situation shown in Figure 2, when we still separate Model and ViewModel layers to the shared components, but also create additional extensions to them for each platform because of platform dependencies and requirements, e.g., to provide more functionality on one of the platforms.

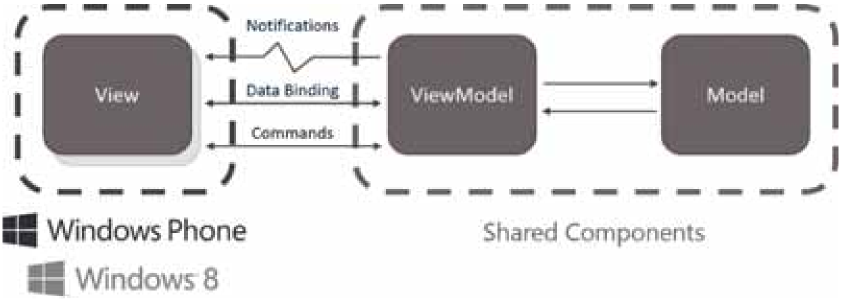


Figure 1. Ideal situation of separation layers with Model-View-ViewModel pattern in cross-platform application development – only the View layer is specific for the targeted platforms.

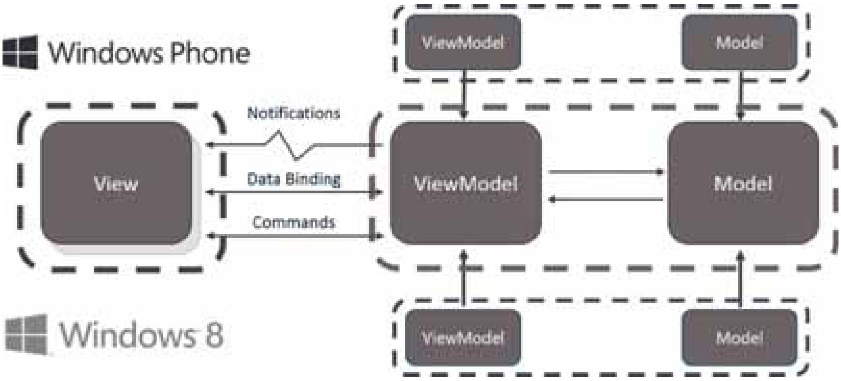


Figure 2. Common situation in cross-platform application development when using Model-View-ViewModel pattern and creating platform specific extensions to the shared components.

To add the platform-specific features we may derive Model or ViewModel classes from the shared base classes or provide interfaces of these classes in shared components. This solution brings the problem of instantiation of these classes in the shared code. There are two options available, based on what we are instantiating:

- Leverage already used Dependency Injection in Model as well – suited for instantiation of services – data producers and consumers, e.g., web service client, database interface client.
- Apply another design pattern, e.g., *Abstract Factory*, for instantiation of data classes [2].

If we decide to apply the Abstract Factory pattern to produce simple data classes, the data producer classes in the Model layer require the instance of general factory described by an interface. Using Dependency Injection we then inject concrete instances. On the other side, this solution introduces additional problem because we code *against interfaces*: data-producing classes return and use interfaces of data classes, ViewModel layer also operates on interfaces of data classes only.

5 Aspects as a solution for platform-specific implementations

Aspect-oriented software development uses aspects to change the behaviour or to extend already implemented functionality. We propose aspects as a solution for problem of platform-specific implementations in cross-platform application development. As stated in [5], we can understand aspects as a mechanism to extend the functionality, e.g., like the extend relationship in UML use case diagrams. If we understand the shared code library as main use cases available to the user, the extending use cases are then realized by the aspects, which in fact are the platform-specific implementations. To fully use aspects to enhance already shared implementation, they have to be defined in the final application for selected platform. Aspects weaving happens during the build process or before the execution of the application. Aspects also allow us to modify implementation not modifiable during the runtime (e.g., using reflection).

5.1 Example problem scenario

We will present our ideas on example Windows 8 and Windows Phone applications built using the MVVM pattern and sharing the same core libraries of Model and ViewModel layers. The applications download new data from the web service and cache it in the local database. Web service calls, data classes and database fall into the Model layer. The ViewModel layer contains classes for displaying the data, UI logic and `ViewModelLocator` class to wire ViewModel classes with Model ones using Dependency Injection and to provide single access point for View. Lastly, the View layer consists of the pages which user interacts with. All platform-independent functionality (Model and ViewModel layers) are separated, except for `ViewModelLocator` class.

Unfortunately, because of different Object-relational Mapping (ORM) engines for both platforms, namely SQLite[1] and LINQ-to-SQL[2], we are not able to use the same data classes in shared libraries. Both engines are based on applying static attributes to the definitions of mapped classes, though different as shown in Listing 1. The problem is that we are not able to provide these attributes in source code of shared library. The data classes are used not only throughout the Model layer but in the ViewModel layer as well, optionally in View layer.

The traditional solution is to use Abstract Factory pattern. In the Model layer we define interface for access to database and define data classes and database access in the final application using particular technology. That leads to unwanted complication in architecture, while the database access class does have to be different for each application but the data classes are better to be not.

Listing 1. Example of Customer data class definition in C# using SQLite (left) and LINQ-to-SQL (right).

```
using SQLite;                       using System.Data.Linq.Mapping;
[Table(“Customers”)]                [Table(Name = “Customers”)]
public class Customer               public class Customer
{                                   {
 [AutoIncrement]
 [PrimaryKey]                        [Column(IsPrimaryKey = true)]
 public int Id { get; set; }         public int Id { get; set; }

                                     [Column]
 public string Name { get; set; }    public string Name { get; set;}

 [Ignore]
 public bool Active { get; set;}     public bool Active {get; set;}
}                                   }
```

[1] SQLite. http://www.sqlite.org/

[2] LINQ-to-SQL, Local database for Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202860(v=vs.105).aspx

5.2 Solution using aspects

To solve the example problem we use aspects to inject new attributes to the general implementation of data classes based on the platform we are targeting. PostSharp Aspect Framework[3] provides the `CustomAttributeIntroductionAspect` aspect for custom attribute introduction to inject new attributes to already defined classes. We apply this aspect with `DataContractAspect` to modify class definitions before runtime. Note that by design it is not possible to modify attributes during the runtime. Listing 2 show example of modified source code from Listing 1, where we defined general attributes to the shared data class. Then each platform-specific application will apply its own aspect implementation to change the data classes (Listing 3) and we can have data classes defined in the shared library for Model layer, letting other classes and ViewModel layer use classes, not just interfaces.

Listing 2. Persistence defined in data classes with custom general attributes.

```
[Persistence.Table("Customers")]
public class Customer
{
  [Persistence.AutoIncrement]
  [Persistence.PrimaryKey]
  public int Id { get; set; }

  [Persistence.Column]
  public string Name { get; set; }

  public bool Active { get; set; }
}
```

Listing 3. Example of aspect to inject SQLite attributes using PostSharp Aspect Framework in C#.

```
[MulticastAttributeUsage(MulticastTargets.Class)]
public sealed class DataContractAspect
  : TypeLevelAspect, IAspectProvider
{
 public IEnumerable<AspectInstance> ProvideAspects(object target)
 {
  Type targetType = (Type)target;
  var typeInfo = targetType.GetTypeInfo();
  var introduceTableAttributeAspect =
    new CustomAttributeIntroductionAspect(
      new ObjectConstruction(typeof(SQLite.TableAttribute)));
     // ... other aspects instantiation ...

  if (targetInfo.IsDefined(typeof(Persistence.TableAttribute)))
  {
    yield return new AspectInstance(targetType,
      introduceTableAttributeAspect);
  }

  foreach (var property in targetInfo.DeclaredProperties)
  {
    // ... weaving aspects to other properties...
  }
 }
}
```

[3] PostSharp Aspect Framework, `http://www.postsharp.net/`

6 Related work

We may understand cross-platform development as the development of a software product line [4] because we create shared core components and then select features (even platform-specific) to create resulting application. Currently, the stated problem of platform-specific implementations in cross-platform application development have been solved using object-oriented design patterns with mentioned downsides in section 4. The MVVM pattern was firstly introduced [6] to separate logics in the architecture design, however we see new dimension of its application in our solution.

7 Conclusions and further work

Cross-platform mobile applications development brings advantages of source code reuse and size reduction. Proper architectural pattern selection gives us ability to test and design user interface in real-time. We stated favourite design patterns in mobile application development and looked further on Model-View-ViewModel pattern with the ViewModel layer as a bridge between the Model and View layers. However, based on the targeted platforms we may still need to provide changes in implementations in shared libraries which may end in cascade effect well-known for layered architectures. Our solution is in application of aspects which will modify or extend the implementations in shared components.

Although we see our solution being effective to the stated problem, we still see further work. Aspect frameworks have to allow to weave custom aspects to the referenced libraries, because aspects are platform-specific and thus cannot be woven in the shared libraries. For instance, latest version of PostSharp Aspect Framework at the time of writing this paper allows to weave custom aspects only within the library because definition of weaving is placed in source code of the targeted library. Because of that, concept mentioned in this paper is currently not possible to be realized.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative Evaluation of Software Quality. In: *2nd Int. Conf. on Software Engineering (ICSE '76)*. IEEE, (1976), pp. 592-605.

[2] Buschmann, F., Schmidt, D.C., et al.: *Books on Pattern-Oriented Software Architecture* (POSA), vol. 1-5. Wiley and Sons Ltd., (1996-2007).

[3] Gartner: *Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013*. [Online; accessed February 16, 2014]. Available at: http://www.gartner.com/newsroom/id/2623415

[4] Kohut, J., Vranić, V.: Guidelines for Using Aspects in Product Lines. In: *8th Int. Symp. on Applied Machine Intelligence and Informatics (SAMI)*, Slovakia, IEEE, (2010), pp. 138-180.

[5] Jacobson, I: Use Cases and Aspects – Working Seamlessly Together. In: *Journal of Object Technology* (2003), vol. 2, no. 4, pp. 7-28.

[6] Smith, J.: *The Model-View-ViewModel (MVVM) Design Pattern for WPF*. MSDN Magazine, Microsoft, (2009). [Online; accessed February 16, 2014]. Available at: http://msdn.microsoft.com/en-us/magazine/dd419663.aspx

[7] Xanthopoulos, S., Xinogalos, S.: A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. In: *Proceedings of the 6th Balkan Conference in Informatics (BCI '13)*. ACM, (2013), pp. 213-220.

Analysis of Source Code Evolution Using Abstract Syntax Tree

Juraj KOSTOLANSKÝ[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`juraj@kostolansky.sk`

Abstract. Source code repositories provide a large amount of information containing the changes introduced in the software system throughout its evolution. However, we lack effective tools to mine repositories for key facts about the software evolution process. In this paper, we analyze existing approaches and propose our own solution for detection of source code changes by comparing two versions of the same code represented by abstract syntax trees. We discuss the strengths and weaknesses of our tool and the direction of our future work.

1 Introduction

More and more effort is devoted to mining various information from source code repositories, as well as from other supporting tools for software development. This information includes those related to source code changes over time. A deep knowledge about software evolution can be useful for bug prediction, project management, risk management, human resource allocation, project monitoring, and so on.

In this paper we analyze existing solutions and approaches in this research area and we propose our own solution. It allows the detection of source code changes by comparing two versions of the same code of a software system. To hide the specific representation of a source code we decided to use abstract syntax trees (AST). Our solution is based on three different approaches to match nodes between trees – subtree matching, leaf matching and bottom-up matching of inner nodes. We show where it works well and where it fails and outline the direction of our future work.

2 Related work

A number of tools for identifying source code changes have been developed. Commonly used tool for code changes detection is *GNU diff*. However, it is a text-based tool which deals with flat information by comparing two files line by line and ignores the hierarchical structure of a software system.

[*] Master study programme in field: Software Engineering
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 486–491.

Chawathe et al. [2] studied the problem of detecting and representing changes in hierarchically structured documents. They implemented *LaDiff* – a system to detect, mark and display changes in LaTex documents. Their algorithm is based on a bottom-up traversal and the assumption that when comparing two labeled trees T_1 and T_2, any leaf in T_1 is at most matched by one leaf in T_2. However, it does not perform well for documents with duplicated parts like source code files.

Fluri et al. [3] enhanced the existing tree differencing algorithm by Chawathe et al. to classify source code changes with some improvements (dynamic tresholds, bigrams instead of largest common subsequences, inner node similarity weightening, etc.) and created the *Change Distiller*.

Raghavan et al. [5] developed *Dex* for analyzing syntactic and semantic changes in C-language code bases. They decided to use abstract semantic graphs as a representation of source codes. Their algorithm combines a top-down matching for establishing a rough correspondence between trees, to reduce the number of pairwise node comparisons that must be done in the next bottom-up matching phase.

Nguyen et al. [4] proposed the algorithm *Treed*, which is another example of a combined node matching. The bottom-up process of their algorithm might be unable to map some nodes, so they introduced the top-down part of mapping. For this purpose, they use *Exas* characteristic vectors to measure structural similarity between subtrees.

If we take a closer look at the ways existing approaches traverse trees, they can be divided into two groups: top-down and bottom-up methods (even those that tries to combine them: *Dex* uses a top-down matching only to establish a rough correspondence, *Treed* uses it as a post-processing step). Both of them have some pros and cons. In the top-down approach, inner nodes are matched before the matching of leaves is known. This conflicts with the observation by Nguyen et al. [4]: two inner nodes should not be matched if they do not have any two matched descendant nodes in corresponding subtrees. The majority of analyzed algorithms uses the bottom-up method, which generally gives better results. It starts with a matching leaf and continues with a matching of inner nodes. Hovewer, the internal structure of an AST is ignored in the leaf matching part.

3 Approach overview

In this Section we describe our method for fine-grained source code changes detection. Our approach is based on a comparsion of two versions of the same code represented by two abstract syntax trees. The whole process can be devided into three steps: (1) ASTs building, (2) nodes matching and (3) edit script generating.

As an powerful and universal AST generator we decided to use *ANTLR*[1]. From a grammar, it generates a parser that can build and walk trees. A collection of grammars for many different programming languages is available online[2].

The core of the process consists of comparing and matching nodes between two generated ASTs. Our matching approach is composed from three parts: (1) subtree matching, (2) leaf matching and (3) bottom-up inner node matching. The subtree matching is our way how to combine top-down and bottom-up methods. When some parts of an AST are significantly changed and we can not match them as subtrees, we try to do it in the fine-grained leaf and inner node matching. We describe these methods deeper in the next Sections.

The last part of the overall process is the edit script generation. An edit script is a sequence of edit operations (inserting, deleting, moving and updating nodes) turning one tree into another [1]. Assume that each edit operation has a cost. The main goal of the previous node matching part is to keep the sum of the costs of the operations in the edit script as small as possible. This sum represents tree edit distance.

[1] `http://www.antlr.org`

[2] `https://github.com/antlr/grammars-v4`

If the matching of nodes between two trees is known, it is relatively simple and straightforward to create the appropiate edit script by traversing the ASTs. If there are matched nodes with different values, we can generate an update operation. If two matched nodes have different (unamtched) parents, the node was moved. If there is an unmatched node in first tree, it was deleted and if there is an unmatched node in the second tree, it was newly insterted.

3.1 Subtree matching

Firstly, the ASTs are needed to be sliced into multiple subtrees. Our slicing method creates subtrees with a given maximal depth, so they do not have to contain all of the corresponding leaf nodes from the original AST. However, each of these subtrees has to contain at least one of the leaf nodes from the original tree. This constraint ensures the observation by Nguyen et al. [4] – we should not match two inner nodes if they do not have any two matched descendant nodes.

When these subtrees are created, they are assigned to sets of subtrees with the same structure. After that, there are two corresponding sets of similar subtrees for a given subtree: one set for each of the original ASTs. This step is necessary for choosing the right direction of most similar subtree searching in the next part of the approach.

For a subtree from the smaller set of similar subtrees we are looking for the most similar subtree from the bigger set. The similarity between these subtrees is calculated from predecessors of the subtree root in the original AST. The existing node matching is taken into account in this process.

If we can identify the two most similar subtrees, they are matched together in the last step. As we have already mentioned above, two inner nodes should not be matched if they do not have any two matched descendant nodes [4]. For this reason only the nodes representing leaves in the original ASTs and their predecessors in the subtrees are matched.

3.2 Bottom-up inner node matching

In this step of the overall matching process we got inspired by Chawathe et al. [2] and Fluri et al. [3]. For inner tree nodes we use a measure of how many leaves the subtrees have in common:

$$\frac{|common(x,y)|}{max(|x|,|y|)} \geq t \tag{1}$$

where x, y are compared inner nodes, $|x|$ denotes the number of leaves contained by x, $common(x, y)$ represents the number of common (matched) leaves of appropiate subtrees rooted at nodes x, y and t is a similarity treshlod usually between 0.5 and 1. We set this treshold to 0.5. In case of multiple such node pairs we create a matching from the most similar one.

3.3 Leaf matching

Matching leaves is similar to matching subtrees. They are assigned to sets of leaves with similar values. The value of a leaf node can be, for example, a variable or a method name. In this case, the similarity is based on the Jaccard index of two bigram sets created from the values of leaves. If the similarity coeficient is greater than a givent value, they are considered similar. This way we can match nodes with changed values (e.g. renamed variables).

Then, for a leaf node from the smaller set we are looking for the most similar leaf from the bigger set. The similarity is calculated in the same way as for subtrees – from predecessors of the leaves – and the existing matching is taken into account, too. Finally, if we can identify the two most similar leaves, they are matched together.

When we talk about leaves in the context of comparing and matching nodes, we mean these leaves together with their parents. The reason for this is that a parent of a leaf node determines the type of this leaf in the AST generated by ANTLR. If there is a method with the same name as the name of a variable, the nodes which represents these names will not be matched together.

3.4 Overall algorithm

Each of the mentioned methods can build a new node matching on an existing one. For this reason, it is useful to run them in multiple iterations until there are no more nodes that can be matched together.

The overall process is shown in Figure 1. It starts with iterations of subtree matching with a decreasing maximum subtree depth. Each of these iterations is followed by the bottom-up inner node matching. Then, the algorithm continues with the leaf matching followed by the inner node matching. At first, the treshold for the leaf value similarity is set to 1, so we are looking for leaves with the same value. This step iterates until there is no change in the node matching. After that, the leaf matching iterations repeat with the treshold set to 0.5, so leaves with changed values are matched.

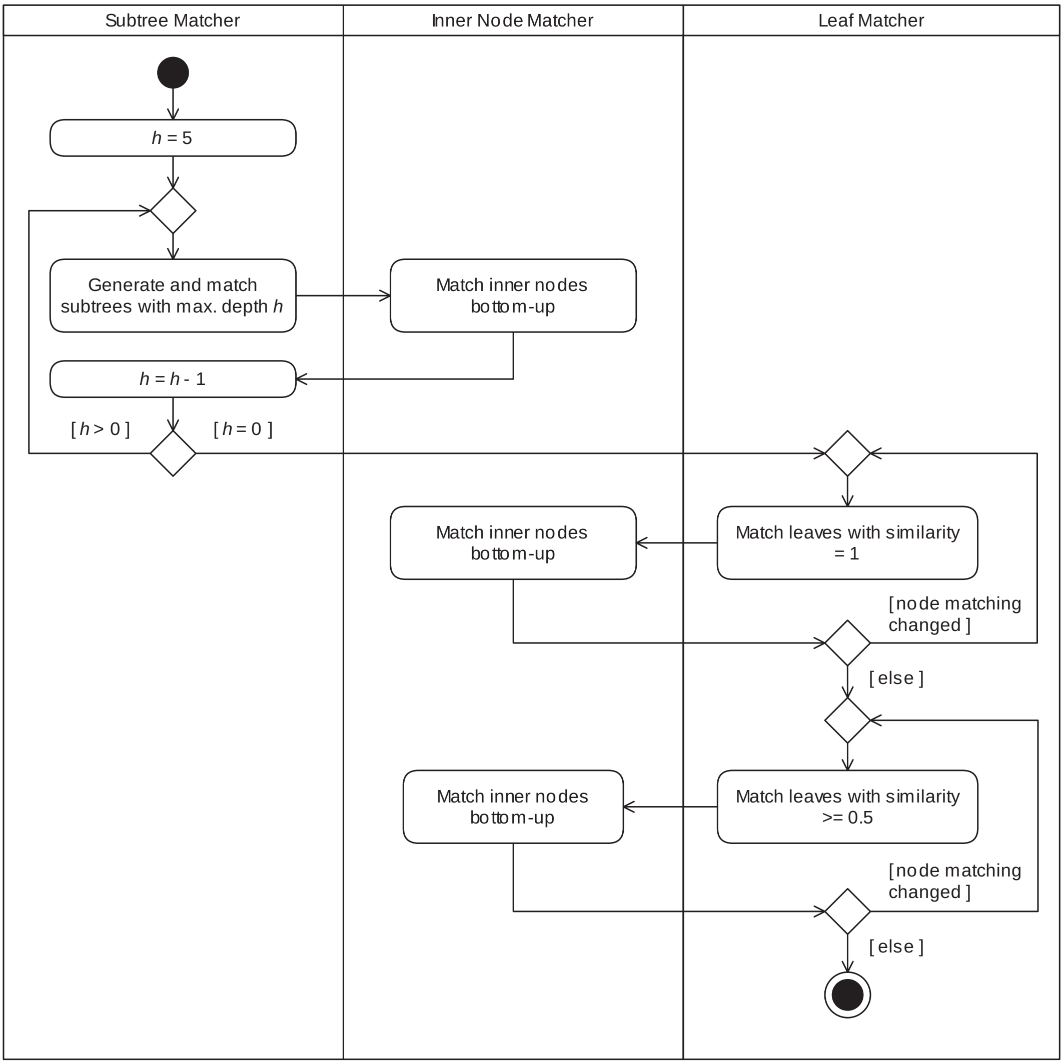

Subtree Matcher
Inner Node Matcher
Leaf Matcher
h = 5
Generate and match subtrees with max. depth h
Match inner nodes bottom-up
h = h - 1
[h > 0]
[h = 0]
Match leaves with similarity = 1
Match inner nodes bottom-up
[node matching changed]
[else]
Match leaves with similarity >= 0.5
Match inner nodes bottom-up
[node matching changed]
[else]

Figure 1. Activity diagram of the overall algorithm.

4 Evaluation

To evaluate our approach we developed a prototype to detect changes in Java language source codes. We tested our tool on a dataset consisting of 25 manually created pairs of source code files. These pairs represented state before and after a code change and covered many different changes which can be seen in real-world software projects. We manually evaluated the output of our prototype for this dataset and summarized the recognized weaknesses of this approach. For each of them we propose a possible solution in this Section.

Based on this evaluation, we can say that about 99% of all matched nodes were matched in the subtree matching part. The precision of our algorithm in this testing was 100%. There were not any false positives, what means that there was not a case of bad matched nodes. All of the observed errors were false negatives – unmatched nodes, that were supposed to be matched. This introduced pairs of deletions and insertions instead of moves in the final edit script. The computed recall varied between 91% and 100% with the average value of 99.3%. The lowest value was caused by an unmatched variable, which was renamed and used in multiple places. The overall f-measure was 99.6%.

However, these values are based on the synthetic dataset and real-world percentages may differ. We are going to perform a deeper evaluation on a dataset based on source codes from real software projects after we implement improvements for the recognized weaknesses and optimize the algorithm.

4.1 Renaming

A common change type in real-world projects is a class, method or variable rename. After renaming, multiple changes occure typically in many places in a source code. In a case of a bigger name change, when the calculated similarity between old and new name is lower than a given treshold, many nodes stay unmatched.

A possible solution for this weakness is a postprocessing step, in wich the ASTs are traversed again. If there are multiple pairs of unmatched leaf nodes with the same "before and after" values, it is probably a rename change in the source code and we can match these nodes together.

4.2 Small subtrees

If there are, for example, small methods in the source code (like setters and getters), into which more lines of code are inserted, the subtree root representing this method can stay unmatched. This happens due to our approach of inner node matching, wich is based on the number of common leaves.

Fluri et al. [3] proposed a solution for these errors – a dynamic treshold for inner nodes similarity. They experienced adequate results for $t = 0.6$ if $n > 4$ and $t = 0.4$ if $n \leq 4$, where n is the number of leaf descendants of the inner node.

4.3 ANTLR trees related weaknesses

Most ASTs built by other tools save a variable-related information (e.g. value, type, modifier) as node attributes. In the case of ANTLR, it is represented by separate leaves. Tree nodes generated by ANTLR do not have attributes, they have only values. For this reason, there is a couple of unmatched nodes. For example, if someone change a variable type from `int` to `double`, corresponding nodes stay unmatched, because their values (strings representing types) are not sufficiently similar.

In this case, the correct matching is uncertain. Should the nodes representing `int` and `double` be matched? Do they represent the same part of a code? Do we want to generate a change operations of removing a variable type and adding another one instead of a change operation of updating a node?

If we want to solve it, we can manually create sets of related node values (e.g. a set for variable types, such as `int`, `double`, `String`) and match node pairs, which values are from the same set. However, this solution is language-specific.

4.4 Leaves with the same similarity

In some cases, two or more leaf nodes in one AST can have the same value, type (specified by their parents) and also the same predecessors. For example, in Java language it can be leaves with value `String` in a subtree representing the code snippet `Map<String, String>`. When these nodes are compared to the nodes in the second AST representing the same source code, they have the same similarity. And because we can not tell which one of them is more similar, they stay unmatched.

We can eliminate this weakness in the leaf matching part. If there are two or more leaves with the same value, type and predecessors, so we can not indentify the most similar one, they will be matched in the order of their occurences in the original source code.

5 Conclusion

In this paper, we propose a method for source code changes detection by comparing two versions of the same code. We use abstract syntax trees as a code representation. The core of our work is comparing and matching nodes between two ASTs. Our method combines three matching approaches: (1) subtree matching, (2) leaf matching and (3) bottom-up inner node matching.

To evaluate this approach, we implemented a prototype and created a dataset, which covered many different code changes. Based on the obtained results, we can say that it is possible to correctly identify the majority of code changes with this method. We describe some recognized weaknesses of our prototype and propose a solution for each of them.

In the future we are planning to implement the mentioned solutions and optimize the parameters of the algorithm (e.g. maximal subtree depth, tresholds for leaf similarity). We are going to perform deeper experiments with our tool on a dataset based on codes from real open-source software projects and compare the results with other available tools.

Acknowledgement: This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] Bille, P.: A Survey on Tree Edit Distance and Related Problems. *Theor. Comput. Sci.*, 2005, vol. 337, no. 1-3, pp. 217–239.

[2] Chawathe, S.S., Rajaraman, A., Garcia-Molina, H., Widom, J.: Change Detection in Hierarchically Structured Information. In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. SIGMOD '96, New York, NY, USA, ACM, 1996, pp. 493–504.

[3] Fluri, B., Wuersch, M., PInzger, M., Gall, H.: Change Distilling: Tree Differencing for Fine-Grained Source Code Change Extraction. *IEEE Trans. Softw. Eng.*, 2007, vol. 33, no. 11, pp. 725–743.

[4] Nguyen, T.T., Nguyen, H.A., Pham, N.H., Al-Kofahi, J.M., Nguyen, T.N.: Clone-Aware Configuration Management. In: *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering*. ASE '09, Washington, DC, USA, IEEE Computer Society, 2009, pp. 123–134.

[5] Raghavan, S., Rohana, R., Leon, D., Podgurski, A., Augustine, V.: Dex: A Semantic-Graph Differencing Tool for Studying Changes in Large Code Bases. In: *Proceedings of the 20th IEEE International Conference on Software Maintenance*. ICSM '04, Washington, DC, USA, IEEE Computer Society, 2004, pp. 188–197.

Extracting Contextual Metadata in Programming Domain

Jakub KŘÍŽ[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
jacob.kriz@gmail.com

Abstract. When programming the programmer does various things apart from writing the source code and he does all these activities in a particular context. By identifying this context and understanding the programmer's activities we can make the results of the programmer's web searches better. In this work we propose a method for metadata extraction from the source code the programmer is working on and his activity in order to build a context model. This model represents his current state and intentions. It can be applied in various ways, in this work we focus on reranking the search results in order to make them more relevant to the current context.

1 Introduction

When programming the programmer faces a great deal of many different problems related to his work. He often uses web search engines to try and solve them. This means that in any given programming session he can make lots of search requests.

When searching the web the users usually enter only a few words to form a search query, which may often lead to unsatisfactory results. Programmers are expected to be usually better at putting together a search query, however, because they make the searches much more often than regular users it is likely that they get inattentive and make searches with unsatisfactory results as well.

Programmers do many things during a typical programming session apart from just writing the source code. They do all these things with a certain intention, in a certain context. By understanding this context we can make the results of the programmers' web searches more accurate and relevant.

In this work we propose a method for extracting context metadata in the programming domain, from the sources specific to programming, in order to build a programmer's context model. We also discuss its application, and propose a method to make the search results better via reranking of the results.

[*] Master study programme in field: Software Engineering
Supervisor: Dr. Tomáš Kramár, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 492–497.

2 Related work

Context extraction and application in general is a common problem, especially in a web search session. However, there are not many works which deal with context in the programming domain specifically.

Previous works, which deal with context during web search, often consider the last few web searches and documents visited from results to be the source of the user's context, e.g. [1,8]. This is probably not ideal for programming domain, because programmers may try to solve many different problems in a fast pace and the context of the search might be different after a short while. Many of these works extract metadata from the documents, e.g. [7], which are then used to improve the search query via query expansion [5] or reranking of the search results [8].

As mentioned, the user's context is often mined from the sources by extracting metadata. Metadata extraction from source codes is not a well researched problem either.

Ohba et al. [6] successfully altered the TF-IDF algorithm in order to mine conceptual keywords from source codes. This method might be altered and used to mine other types of keywords from source codes as well. The TF-IDF algorithm can be successfully altered further, for example to give different importance to different parts of the document [2], for example to give more weight to the part with which the user is currently working.

3 Context extraction and application

Based on our experience with the programming domain we group the problems programmers usually face into three categories:

Conceptual problem - The programmer is trying to understand the idea of an algorithm or come up with a conceptual solution to a problem.

Technical problem - The programmer solving a problem associated with using the methods of the language or library or API methods which he is currently using.

Error - The programmer is solving an error which occurred.

The searches the programmers make when facing a problem from each of these groups can be quite different, which is why we use these groups as the basis for our method. We consider the programmer's activity to be the most important source of context, which greatly outweighs other possible sources of context. The method we propose for context extraction in programming domain uses two specific sources of context: the source code the programmer is currently working with and his activity in the development environment. The method extracts metadata from these sources and creates a context model, which can be further used to improve the web search results.

3.1 Context model

The context model should represent the programmer's intentions in any given point in time. It should include information about his activity and about the problems he is trying to solve. The context model which we propose is designed to be very versatile and has the following structure:

- Conceptual part
- Technical part
    - Language
    - Framework / libraries

    - ○ Key identifiers

- Error part
- Programmer’s state

The *conceptual part* says about the conceptual intentions of the programmer. It should answer questions like what problem is he trying to solve, what algorithm is he trying to design. It is composed of a set of ranked keywords. The *technical part* includes information about the technologies the programmer is currently working with and it is divided into three parts. *Language* is the identifier of the language he is currently using, such as *java*. *Framework / libraries* is a set of identifiers of frameworks and libraries the programmer is currently working with, such as *android*. *Key identifiers* is a set of ranked identifiers of core, library or API methods and objects, which are currently important to the programmer. An example of such identifier is the name of a class from Android API *ContactsContract*.

The *error part* contains the last run-time error. This part is separated from the technical part because we consider it very important for our method. An example for this part is an identifier of an error `j`ava.lang.NullPointerException. The *programmer’s state* part says in which state is the programmer currently in, specifically on which type of problem he is currently working. This part can have three values: *conceptual*, *technical* or *error*.

3.2 Metadata extraction from source codes

The metadata for the context model we described in the previous section are extracted from the source code the programmer currently works with. When designing this extraction method we focused on the programming language Java and some parts of the method are specific for this language. However, it should not be problematic to replicate this method for other programming languages.

3.2.1 Conceptual keyword extraction

The method we use for extraction of keywords and identifiers is inspired by a method successfully used to extract conceptual keywords from source codes in previous work [6]. We changed and extended this method for our purposes. The algorithm used for extraction works in the following way:

1. extract the conceptual parts of the source code,
2. split the conceptual parts to words,
3. filter out stop words,
4. rank the words using the weighted TF-IDF,
5. sort the words by their rating and pick the best rated.

Conceptual parts of the source code are parts where their content is fully chosen by the programmer and, therefore, we assume that they are representative of his intentions:

- comments and JavaDoc,
- variable names,
- own method, object and class names.

Table 1. Description of constants and variables used by weighted TF-IDF.

al	number of active line	sk	value for non-important lines
tl	number of word line	n	occurrences of word in document
k	linear dependence of important lines	l_limit	number of important lines

Stop words are very short words - two characters or less. These can be found in many source codes quite often, because many programmers tend to use very short names for their unimportant variables. This method uses TF-IDF as a method to rank and sort keywords. We chose TF-IDF because, despite its simplicity it is widely considered to be an accurate method for keyword extraction from natural language and the words, which we extract from conceptual parts of the code are similar to natural language. As a corpus for extraction we use the set of documents of the programmer in the particular language.

Our method uses altered, weighted TF-IDF to account for the position of word in a document relative to the active line on which the programmer is working, which is calculated using the following formula for the *tf* part, the *idf* part is calculated in regular way:

$$f_i = \begin{cases} \frac{k-abs(al-tl)}{k} & \text{if } abs(al-tl) < l\_limit \\ sk & \text{else} \end{cases}$$

$$tf = \sum_{i=1}^{n} f_i$$

Table 1 contains the description of constants and variables used by these formulas. When calculating the line numbers, the blank lines of the source code are ignored. By using these formulas we ensure that the words which are close to the active line are much more important than the words from the rest of the document, but we also ensure that some words will always be extracted.

3.2.2 Key technical identifiers extraction

The method we use to extract technical identifiers is very similar to extraction of conceptual keywords described in the previous section. The method works in the following way:

1. extract the technical parts of the source code - identifiers,
2. rank the words using the weighted TF-IDF,
3. sort the words by their rating and pick the best rated.

The technical parts of the code are the parts where the programmers do not pick the content, these are:

- reserved words of the language,
- identifiers of externally defined method, objects and classes.

For ranking of the identifiers we use the same weighted TF-IDF as described in the previous section. In this case, however, we use the TF-IDF algorithm for slightly different reasons. Since we use the programmer's own collection of source code documents as the corpus, the identifiers, which are used less frequently throughout this collection of documents are going to have higher rating. We consider this behavior as desired, because we assume that the programmer will need more help with the identifiers which he has used less often.

3.3 Programmer's state detection

Our method divides the programmer's state based on the type of problem he is currently solving into three possible outcomes: conceptual, technical and error. The programmer's state is decided by analyzing his activity in the IDE. The method we use considers the following factors:

- time since last writing,
- type of lastly written expression,
- time since last selection,
- type of lastly selected expression,
- time since last error,
- time since last compilation.

The method classifies the programmer's state based on these factors. We consider using a couple of different techniques: a custom designed decision tree and supervised learning. The decision tree is based on our own experience with the programming domain. The algorithms for supervised learning we use are Naive Bayes and kNN [4].

3.4 Context model application

The context model which the method extracts is quite versatile and could possibly be used for many different purposes. For example, if we considered other users we could use the model to create a recommendation system based on collaborative filtering.

In our work we use the context model in order to make the programmer's web search results better. We do this by reranking the search results. Based on the detected programmer's state the method picks the relevant part of the content model. When ranking the search results the method boosts the rating for documents, which contain words or identifiers which were also found in the context model.

4 Experimental evaluation

We evaluate the extraction method in parts. So far we have focused on evaluation of the parts of the method which deal with the extraction of conceptual keywords and key technical identifiers as described in Section 3.2. The experiment used for evaluation of metadata extraction is based on comparing the data our method extracts with explicit feedback from users-programmers. The user was asked to pick a part of his own source code by selecting the active line. The method extracted the metadata from this part of the code. This metadata, two groups of keywords and identifiers were shown to the user in random order and he was asked to order them based on their relevance to the part of the code and their usability in a search query. Additionally, the user was also asked to rate each word or identifier as relevant or not.

The experiment is evaluated by comparing the user's order of the terms and the order which was created by the method. To compare these two orders we use Kendall's tau correlation coefficient [3]. We also evaluated how many of the extracted words and identifiers were considered as relevant. Five users participated in the experiment, completing 50 scenarios in total. The results of the experiment are shown in Table 2. We consider the results of the experiment to be promising and showing that the method manages to extract relevant keywords and key identifiers from active parts of the source code.

Table 2. Results of the metadata extraction experiment.

type	average τ	positive τ ratio	number of key
conceptual	0, 36	80%	3, 2
technical	0, 29	74%	2, 4

5 Conclusions and future work

In this work we propose a method for extracting a context model from programmer's activity and source codes. We also discuss a method for application of this context model in order to improve programmer's web search results. Part of this extraction method has been evaluated and shows promising results. In the future we intend to evaluate the rest of the extraction method. We believe the context model, as proposed, is very versatile and can be used in many ways. In our future work we intend to implement and evaluate the context model application method based on reranking of search results which we discuss in this work.

Acknowledgement: This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., Cui, X.: Modeling the Impact of Short- and Long-term Behavior on Search Personalization. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '12, New York, NY, USA, ACM, 2012, pp. 185–194.

[2] Kříž, J., Kramár, T.: Keyword Extraction Based on Implicit Feedback. *Bulletin of the ACM Slovakia*, 2012, vol. 4, no. 2, pp. 43–46.

[3] Kendall, M.G.: A New Measure of Rank Correlation. *Biometrika*, 1938, vol. 30, no. 1/2, pp. 81–93.

[4] Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. In: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, Amsterdam, The Netherlands, The Netherlands, IOS Press, 2007, pp. 3–24.

[5] Kramár, T., Barla, M., Bieliková, M.: Personalizing Search Using Socially Enhanced Interest Model Built from the Stream of User's Activity. *Journal of Web Engineering*, 2013, vol. Vol. 12, no. 1&2, pp. 65–92.

[6] Ohba, M., Gondow, K.: Toward mining ”concept keywords” from identifiers in large software projects. In: *Proceedings of the 2005 international workshop on Mining software repositories*. MSR '05, New York, NY, USA, ACM, 2005, pp. 1–5.

[7] Shen, X., Tan, B., Zhai, C.: Context-sensitive information retrieval using implicit feedback. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. SIGIR '05, New York, NY, USA, ACM, 2005, pp. 43–50.

[8] White, R.W., Bennett, P.N., Dumais, S.T.: Predicting short-term interests using activity-based search context. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. CIKM '10, New York, NY, USA, ACM, 2010, pp. 1009–1018.

Estimation of Programmer’s Karma Based on Programming Tasks

Eduard KURIC*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
eduard.kuric@stuba.sk

Abstract[1]

Every day, programmers need to answer several questions to find solutions and make decisions. It requires the integration of different kinds of project (software system) information, as well as, it depends on the programmers' knowledge and skills.

In a company, estimation of programmers' expertise allows managers and team leaders, e.g., to look for specialists with desired abilities, form working teams or compare candidates for certain positions. Last but not least, it is suitable for search-driven development to rank source code results not only based on relevance but also authors' (programmers’) expertise. Relevance of software components is of course paramount, however, trustability is just as important. When a programmer reuses a software component from an external source he has to trust the work of an external programmer who is unknown to him. If a target programmer would easily see that a programmer with a good level of expertise has participated in writing the software component, then the target programmer will be more likely to think about reusing.

In academic environment, estimation of students' expertise allows a teacher to evaluate students' knowledge and skills. Based on it, e.g., the teacher can adapt and modify his teaching practices. On the contrary of a software company, where software is created by professionals, in academic environment, students learn how to design and develop software (programs). Therefore, the estimation of programmer's expertise requires a different approach.

We present a novel approach to automatic estimation of programmer's expertise (karma) based on programming tasks in academic environment. We have applied and evaluated our method in a course called *Data structures and algorithms*. On the contrary of a software company, where software is created by professionals, in academic environment, students learn how to design and develop software (programs). Therefore, the estimation of programmer's expertise requires a different approach. Estimation of students' expertise allows a teacher to evaluate students' knowledge and skills. Based on it, e.g., the teacher can adapt and modify his teaching practices.

Acknowledgement: This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

* Doctoral degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 498-504.

Building Distributed Transactional Memory using CRDT

Aurel PAULOVIČ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
aurel.paulovic@stuba.sk

Abstract[1]

Concurrency and parallelism in distributed systems is due to the need to deal with node failures, high latency, data replicas and network partitions often very difficult to get right. Researchers try to simplify the development of distributed systems by designing of new methods of concurrency control. One of such proposed concurrency control mechanisms is transactional memory that aims to bring transactions, which can wrap multiple individual operations into a single failure-atomic indivisible operation, into the computing model at the level of a basic programming construct.

Building an effective transactional memory runtime for distributed systems requires us to focus on minimizing the amount of transaction conflicts and the lowering of the needed amount of synchronization between system nodes. In our work, we try to achieve these goals by designing a distributed transactional memory that makes use of convergent and commutative replicated data types (CRDTs) invented by Shapiro *et. al.* [1], which we enrich by the possibility of using them in transactions.

Since the consistency model based on sharing data via CRDTs uses eventual consistency, we can synchronize the data replicas employing asynchronous messaging without the need of a total ordering which allows us to achieve scalability and deal more readily with node failures. However, because eventual consistency and the use of CRDTs can be restrictive and reduce the expressibility of our model, we propose also the use of a special locking service. The locking service can be optionally used by a transaction to temporarily impose some global invariant of the system on the shared data that could otherwise not be provided solely by the local information available in CRDTs.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M.: Convergent and Commutative Replicated Data Types. *Bulletin of the European Assoc. for Theoretical CS*, 2011, no. 104, pp. 67–88.

* Doctoral study programme in field: Software Engineering
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 505-512.

Empirical Metadata Maintenance in Source Code Development Process

Karol RÁSTOČNÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`karol.rastocny@stuba.sk`

Abstract[1]

Analysing and understanding history of a software project development is important for project managers and developers. They spend a lot of time by analysis why some events have occurred and by proposing measures to prevent negative events and also to repeat positive ones. Managers and developers often find these actions more important than forecasting next project evolution. During this process they utilize software metrics based on source code (e.g., LLOC) and behaviour of developers (empirical software metrics). Nowadays, especially code-based software metrics are often used. It is because of number of approaches and availability of source code, which allows calculation of metrics in the time, when they are required. But even if we calculate code-based metrics across change-sets (commits) we have still only static information about source code with results of development decisions and we are not able to find an answer to the question "Why?". This question can be answered after a consideration in a wider context which can be brought by empirical software metrics. Even though empirical software metrics are not widely used. Most of software project managers used only basic information from systems for development aid or in occasional cases collaboration of developers extracted from discussions. This is caused by expensiveness of empirical data collection, quality of collected data and lack of empirical metrics.

The solution of the problem of lack of metrics and utilities can lay in utilizing approaches and methods from web engineering that analyse and use empirical data. If we look on the information space of a software house as on a "spider-web" of software artefacts, in which relations between artefacts are their dependencies at different levels of an abstraction, we can modify and reuse web engineering methods and approaches in a domain of software houses. We utilize this principle in the project PerConIK (http://perconik.fiit.stuba.sk). In the project PerConIK we collect empirical data via software tools and extensions for integrated development environments and web browser installed in developers' working environments that collect activities as open/add/edit source code file, copy/paste and visited webpages and biometrics.

We store developer-oriented empirical data and software metrics in form of information tags that are directly related to software artefacts. But empirical data are error prone, while their validity can be affected by various sources. In this work we proposed basics of information tags and analysed their dependencies and we proposed and evaluated basics of proposed approaches for maintenance of empirical software metrics anchored to software artefacts via information tags.

[*] Doctoral degree study programme in field: Software Engineering
Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering Faculty of Informatics and Information Technologies STU in Bratislava

[1] Full paper available in printed proceedings, pages 513-520.

Innovative Designs and Applications

Eye-blink Detection Using Gradient Orientations

Tomáš DRUTAROVSKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xdrutarovsky@is.stuba.sk

Abstract. State-of-the-art algorithms offer a real-time human face and eye detection which can be used to detect eye-blink. In this paper we present a feature descriptor. Gradient orientations and magnitudes are used to build the feature descriptor. We use the difference between samples of open and closed eyes. Gradient descriptors are further used to train the SVM. SVM can predict the open or closed eye state from the given input data. Due to the knowledge of the state of the eye (open or closed), eye-blink frequency and duration can be computed. These parameters are used to establish the level of sleepiness what can be used to prevent the driver from incoming microsleep.

1 Introduction

Today's technologies offer us the opportunity to capture video of a driver in the car. Information about eye openness can contribute to the estimation of blink frequency and duration. These two values are considered to be the main signs of sleepiness [10]. Further analysis of these parameters can contribute in microsleep prevention.

Webcams and smartphones are available and provide a real-time image acquiring with sufficient frame resolution. World of computer vision knows algorithms for human's face and eye detection and tracking what can be used for real-time eye sample processing. This leads to ideas of blink detection and eye state determination. Several works are devoted to the distinction between *open* and *closed* eye, but all of them have gaps in the robustness of the solution. That is because of different environment conditions, illuminations changes and various eye appearances.

Our goal is to construct a gradient orientation descriptor, which could describe open and closed eyes differently so it could be used to train the Support Vector Machine (SVM) [2]. We propose a method of gradient information calculation from the frame and processing them so the gradient features are more distinguishable. Knowing the difference between open and closed eye provides several advantages. We can use the information to detect eye-blinks or percentage of eye closure.

[*] Bachelor study programme in field: Informatics
Supervisor: Andrej Fogelton, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 523–528.

2 Related work

The automatic drowsy driver monitoring and accident prevention system was presented in [3]. The system analyzes pupils using *Horizontal Symmetry Calculation* (HSC). The system receives input colored frames from a video camera and measures the eye-blink duration of a driver constantly. After face detection the neural network-based detector is used for the precise eye pupils position. The head rotation angle is calculated using the vertical position of both pupils. If eye detection in the next frame fails, the angle helps to determine the right face and eye position. Detected pupils are analyzed using HSC to determine whether the eyes are *Open* or *Closed*. HSC uses the fact that folded closed eye samples have more difference pixels than open eye samples. Mentioned algorithm was tested on ZJU database and it achieved 94.8% accuracy for eye-blink detection. This method is dependent on samples' quality and precise eye alignment and cropping.

Work presented in [7] solves the problem of blink detection using the frame pixel difference. Authors estimate blink when the pixel intensity difference of consecutive frames is higher than the preset threshold. Method also uses thresholds to consider a non-uniform change of the illumination for each eye in consecutive frames and also an exclusion of voluntary blinks from the further analysis. Similar method was proposed in [6] where authors localize eye positions according to the significant change of the intensity in the consecutive frames. After successful eye localization, the system learns how an open eye appears so it can be used in the next phase of blink detection. Eye closure is estimated due to correlation score between current eye template and learned template examples. Mentioned algorithms require video samples with no bigger face moves and uses relatively many thresholds.

In the paper [5] authors proposed a vision-based drowsiness detector which can detect a driver in a realistic driving simulator. This detector is based on the infrared stereo camera. The detector constantly tracks eyes and estimates the percentage of closure also known as *PERCLOS*. For the best estimation, PERCLOS values were compared to results of several psychological experiments. First, face and eyes are detected using Viola – Jones detector [12] and detection failures are then corrected using *Kalman filter*. Subsequently, the sequence of filters is used to improve the frame quality so the PERCLOS can be estimated more accurately. PERCLOS is calculated from the ratio between the iris height in the frame and the nominal value which is assigned during a ten-second calibration at the start of tracking. This detector reaches high recall (90.68%) and low false positive rates using their own database consisting of 25 hours of driving. However, the system uses an infrared stereo camera what makes it an expensive solution with high hardware requirements.

Another method of eye state determination is measurement of pixel amount in eye regions. Authors in [4] presented a method of eye-blink detection using *intensity horizontal projection (IHP)*. The method uses the fact that iris has lower IHP value than other regions around the eye. This means that closing of the eyelid causes noticeable changes in the histogram of IVP values. On the other hand, algorithm presented in [8] measures eye-blink using *intensity vertical projection (IVP)*. After applying the median filter, the IVP of eye regions without eyebrows is measured. It is considered that the ratio of maximal IVP to minimal IVP for the open eye is higher than for the closed eye. Open eye has also higher maximal IVP value than closed eye. Although these methods seems obvious, they work accurately only for specific eye types. Therefore we can not consider them robust and reliable enough.

3 Gradient orientation descriptor

One of the most significant characteristics for the human discrimination of eye states is an ability to recognize shape. Shape is often described by gradient orientations. We want to use that fact and build a feature descriptor which could help the computer to discriminate different eye states using the gradient orientations. We propose the gradient calculation and orientation sorting for each eye sample. We use weight map to adjust weights of values which are added to the orientation bins.

3.1 Gradient calculation and sorting

First step of descriptor construction is the gradient calculation from the image. In this paper we do not discuss face and eye detection and tracking. We assume that the input eye sample contains nothing but eye aligned in the center of the sample. In our work we used Viola – Jones detector to obtain eye rectangles. After that, we align rectangles according to the pupils using the gradient pupil locator. In the case of the closed eye sample, we assume that eyelashes or eyelids' link is located in the center of the sample.

Our method uses *Sobel* operator for the edge detection, because it is partly rotation invariant. Sobel gives better results in computing diagonal edges than the edge detector which uses $[-1, 0, 1]$ kernel and image preprocessed with Gaussian filter.

Input image is decomposed into derivatives – horizontal image gradients dx and vertical image gradients dy. Using these two images we can compute gradient magnitude m for each pixel as maximum of absolute values from both images (Equation 1). We can also compute gradient orientation α using the tangent function (Equation 2). This approach was inspired by the work presented in [1].

$$m_{x,y} = max(abs(dx_{x,y}), abs(dy_{x,y})) \tag{1}$$

$$\alpha_{x,y} = \tan(dx_{x,y}, dy_{x,y}) \tag{2}$$

Another important operation is constructing of the weight map. This map is used to control the weight of the image pixels so the pixels in the center of the image have higher weights than pixels at the image border. Each point in the eye sample has own weight $w_{x,y} = e^{n_{x,y}}$. In our algorithm, $n_{x,y}$ is linear interpolation between 0 for border pixels and 12 for center pixels, according to the pixels position in the weight map. Exponent values were chosen due to the empirical test results. This approach is inspired by the FREAK descriptor [11] which used knowledge about human's retina and vision sharpness.

After these operations, we have two essential values for each image pixel so we can sort gradient orientations into 360 bins. Each gradient orientation is dispersed from -10 to 10 degrees and classified into bins to avoid errors in orientation calculation. Appropriate bin is increased with value from weighted map. However, we consider only gradients with magnitude higher than 10 (we consider magnitude values from 0 to 255) to avoid non-significant gradients.

3.2 Orientations function

Following step of the gradient descriptor construction is the processing of the 360-binned orientation function. Input orientation function has a lot of local minima and maxima, therefore we smooth the function. Each bin is smoothed using average value from the four closest bins. We smooth the function until it has 4 extremes exactly – two minima and two maxima. As we consider aligned eye samples, these extremes are often around 0, 90, 180, 270 degrees. We use the fact that open eyes (Figure 1) have maximum with highest value around 90 degrees, but closed eyes have higher maximum around 270 degrees bin (Figure 2). That is because an open eye has more gradients oriented vertically down near the eye center. Moreover, open eye sample has bigger disperse of the maximum peak what is caused by higher amount of horizontal orientations around the iris.

Smoothed function is then aligned to the first local maximum. Rotation invariance can be guaranteed by shifting bin values according to the rotation angle. This angle can be computed as the angle between line connecting both irises and horizontal line.

Orientations function with 4 extremes is used to construct the final gradient descriptor. Our proposed descriptor consists of 8 numbers – four pairs. Each pair represents one extreme or peak and consists of two numbers $rate_{x,y}$ and $distance_{x,y}$. $rate_{x,y}$ represents the rate of the peak height among other peaks. This means that summed size of all four peaks gives 1. $distance_{x,y}$ represents the

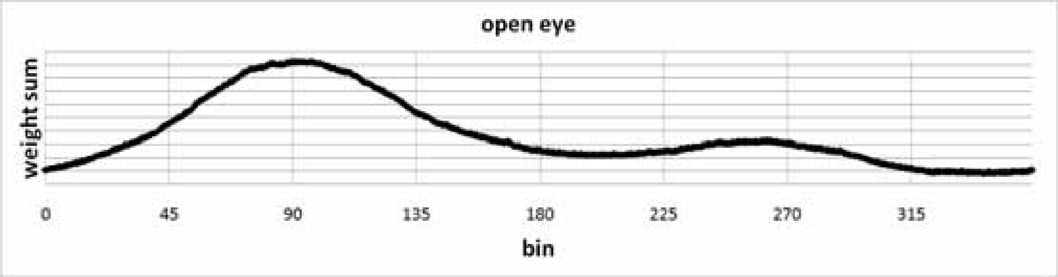


Figure 1. Orientations function of the open eye samples. Function is averaged from 200 eye samples. As we can see, function has two local maxima and first maximum has higher value.

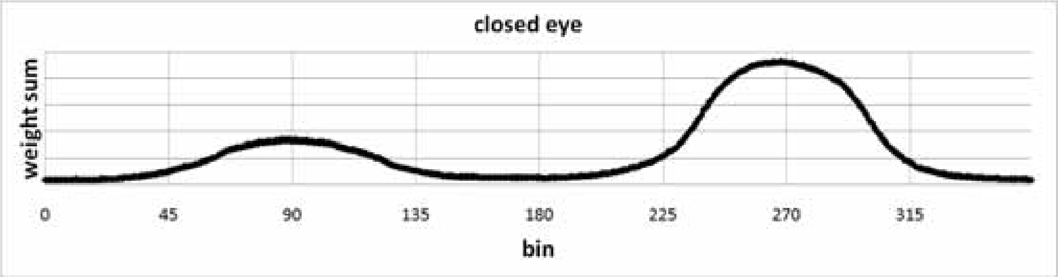


Figure 2. Orientations function of the closed eye samples. Function is averaged from 200 eye samples. Function has also two local maxima, but second maximum has higher value.

number of bins located between considered peak and the closest right peak. Descriptor $\bar{O}$ represents descriptor of averaged vector from 200 open eye samples (Equation 3).

$$\bar{O} = [0.59, 0.30, 0.10, 0.18, 0.24, 0.22, 0.07, 0.30] \tag{3}$$

4 Evaluation

Both types of eye states (open and closed) are used to train the SVM classifier. In this paper we construct the SVM from 200 open eye and 200 closed eye samples. Trained SVM is used for further state prediction. Testing was performed on other 200 open and 200 closed eyes and results are summarized in the Table 1.

Evaluation of the descriptors' accuracy was performed on our own eye dataset, which consists of the six individuals sitting in front of the camera. Eye samples were acquired using Viola – Jones eye detector and gradient pupil locator at the average resolution of 24×24 pixels.

Our method based on the gradient orientations and image weighting achieved accuracy of 87.8%. We have compared our final gradient descriptor to two other methods.

First of the compared methods is the gradient descriptor without weighting. Absence of weighting supports the fact that the strong links like eyelashes or eyebrows can cause loss of accuracy. This eye features have strong gradients which are not ignored as in the weighting method. Therefore the method achieved accuracy of 71.0%.

Another compared method is SIFT descriptor [9]. Our implementation of the descriptor has one point of interest located in the center of the sample with interest area stretched on the whole sample. Final descriptor has 128 dimensions – quantized gradient directions, as usual. This descriptor achieved accuracy of 94.3%.

Table 1. Results of all tested methods. TP represents true positive rate (correctly identified closed eye) and TN represents true negative rate (correctly identified open eye).

Method	TP	TN	overall
No weighted	67.5%	74.5%	71.0%
Weighted	93.5%	82.0%	87.8%
Sift	95.0%	93.5%	94.3%

5 Conclusion

In this paper we proposed the method of constructing feature descriptor based on gradient orientations. Our solution involves gradient orientations calculation and sorting, weighting and descriptor building. Descriptors of open and closed eye samples were used to train the SVM, which is used to predict results. Proposed method achieves accuracy of 87.8%.

Although we do not achieve the accuracy of SIFT descriptor, our descriptor consists of 8 numbers only which might result in the potential performance increase. Our method depends on the eye alignment what can affect the gradient orientations and subsequently the feature descriptor. We consider the result encouraging and see the potential of the method in the further enhancement.

We want to focus on the location of significant eye regions and use them to adjust the weighting. Moreover, we want to use the improved descriptor to implement eye-blink detector. Knowing the information about the current eye closure can lead to the estimation of the blink duration and blink frequency. Our future aim is to build a detector with fair trade-off and acceptable solution robustness.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, vol. 33, no. 5, pp. 898–916.

[2] Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning*, 1995, vol. 20, no. 3, pp. 273–297.

[3] Danisman, T., Bilasco, I., Djeraba, C., Ihaddadene, N.: Drowsy driver detection system using eye blink patterns. In: *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, 2010, pp. 230–233.

[4] Dinh, H., Jovanov, E., Adhami, R.: Eye Blink Detection Using Intensity Vertical Projection. In: *Proc. of the 5th International Multi-Conference on Engineering and Technological Innovation (IMETI)*, Huntsville, Alabama, USA, Dept. Electrical and Computer Engineering, University of Alabama in Huntsville, 2012.

[5] Garcia, I., Bronte, S., Bergasa, L., Hernandez, N., Delgado, B., Sevillano, M.: Vision-based drowsiness detector for a realistic driving simulator. In: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 887–894.

[6] Grauman, K., Betke, M., Gips, J., Bradski, G.: Communication via eye blinks - detection and duration analysis in real time. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Volume 1., 2001, pp. I–1010–I–1017 vol.1.

[7] Kurylyak, Y., Lamonaca, F., Mirabelli, G.: Detection of the eye blinks for human's fatigue monitoring. In: *Medical Measurements and Applications Proceedings (MeMeA), 2012 IEEE International Symposium on*, 2012, pp. 1–4.

[8] Lee, W.O., Lee, E.C., Park, K.R.: Blink detection robust to various facial poses. *Journal of Neuroscience Methods*, 2010, vol. 193, no. 2, pp. 356 – 372.

[9] Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 2004, vol. 60, no. 2, pp. 91–110.

[10] Schleicher, R., Galley, N., Briest, S., Galley, L.: Blinks and saccades as indicators of fatigue in sleepiness warnings: looking tired? *Ergonomics*, 2008, vol. 51, no. 7, pp. 982 – 1010.

[11] Vandergheynst, P., Ortiz, R., Alahi, A.: FREAK: Fast Retina Keypoint. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, vol. 0, pp. 510–517.

[12] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Volume 1., 2001, pp. I–511–I–518 vol.1.

Ensuring QoS in SIP Single Port VoIP Networks with MPLS using SLAMCA

Marek Galiński[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
galinski.marek@gmail.com

Abstract. In this article we brought a unified solution for monitoring and measuring the Quality of Service in networks based on SIP Single Port architecture. SIP Single Port enhances the multimedia session management, however it does not evaluate the quality of the network below. We provide the MPLS-TE network to define several paths in the network and together with our solution we allow the SIP Single Port architecture to utilize them more efficiently than using default network mechanisms. To monitor quality of these defined paths, we brought a system SLAMCA (SIP single port Line Availability Monitoring Capable Agent) that generates pseudo-traffic and evaluates media session quality by calculating its Mean Opinion Score and providing this information to the SIP Single Port architecture controller used in this network.

1 Introduction

Multimedia sessions based on SIP use generally three protocols - SIP, RTP and RTCP, while each of these has to communicate via its pair of (in most cases) UDP ports. This makes session management in core networks inconvenient, as each stream is transmitted independently. Furthermore, if NAPT is implemented, sessions can not be initiated without using additional methods to avoid NAPT refusing connection. By using SIP Single Port [8] that multiplexes SIP, RTP and RTCP streams into one stream transmitted between one pair of UDP ports only, session management is much more convenient, as each session is identified by one single data stream. NAPT traversal issue is solved as well, as the only needed port is opened at the time of session initiation - i.e. SIP `INVITE`, as shown in [8]. This fact allows us to implement additional technologies in provider's core network to ensure Quality of Service, which is the main added value of this work. We developed complex system which monitors line availability and allows us to avoid congestion that results in unexpected session drops. The article is organized as follows: In Section 2 we described parts of our system and how we use them. Section 3 describes actual solution.

[*] Bachelor study programme in field: Computer Engineering
Supervisor: Ján Murányi, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

Table 1. MOS values table.

5	Perfect, approaching to face-to-face communication
4	Fair quality. Certain imperfections can be perceived
3	Low quality, sometimes impossible to understand the other side
2	Very low quality, hard to understand the other side generally
1	Impossible to communicate

2 State of the art

Once all streams are multiplexed into one SIP Single Port stream that contains both signalization and media belonging to one call, now to make the network management more convenient, we have to look up a possible way how to shape call traffic in providers network.

2.1 MPLS traffic engineering

Each service provider has its own network topology. These topologies are mostly redundant, meaning that there are at least two possible paths between source and destination. We can leave call routing to one of the routing protocols configured in our network, but this solution does not guarantee any quality of service, or load balancing. Data will always be routed via one route, considered by routing protocol as the best one. This does not have to be the solution we would like to use in our network, as it can result in congesting one line while other line will be unused.

The most commonly used technology is Multi Protocol Label Switching (MPLS) with its “Traffic Engineering” application (abbreviated MPLS-TE) [2]. MPLS is a technology to deliver IP services. Data packets in MPLS networks are not forwarded using IP lookup, but by using labels. MPLS enabled router does not look into IP header to forward packets [7]. MPLS Traffic Engineering (MPLS TE) is one of the MPLS application, that allows us to shape traffic in our network. MPLS TE uses MPLS forwarding, but in addition determines the routes for traffic flows across the network [2]. This means we are able to configure dynamic or static tunnels in our network, where each tunnel has its own bandwidth, and by using access lists (ACLs) we can decide which packet will be sent via which tunnel.

This technology allows us to configure tunnels manually, by configuring it hop-by-hop. It is important to be aware of the fact that tunnels are always one-directional - it means it is fully up to us, whether both directions of data stream will be sent over the same path, or via two independent, different paths.

2.2 Call quality evaluation

All above described technologies are meant to be used in provider networks to ensure quality of service. Now we need to have certain control mechanism to determine, whether call quality is acceptable or not, as well as detection of line congestion. Mean Opinion Score (MOS) is a numerical indication expressing voice and video quality. MOS is expressed in one number from 1 to 5, 1 being the worst and 5 the best [6]. The values can be interpreted as shown in Table 1.

There are several methods to calculate MOS automatically by analyzing the data stream. ITU-T in its recommendation G.107 recommends the E-model [5]. Method of calculating MOS according to E-model is given by Equation 1:

$$R = R_o - I_s - I_d - I_{e,eff} + A, \tag{1}$$

Table 2. Codecs parameters table.

Codec	a	b	c	d	e	f	g	h	i	j
G.729	3.61	-0.13	1.22e-03	3.76e-03	-2.29e-05	4.71e-06	-5.16e-05	2.54e-08	1.28e-07	-4.43e-08
iLBC	3.64	-5.25e-02	2.45e-03	1.34e-03	-2.71e-05	-2.07e-05	-1.76e-05	2.95e-08	6.23e-08	1.12e-07

where R express MOS, R_o express the basic signal to noise ratio, I_s express all impairments that occur simultaneously with the voice signal, I_d represents sum of all impairments due to delay and echo effects, $I_{e,eff}$ is an effective equipment impairment factor and A is an advantage factor that allows for certain systems trading voice quality for convenience [4].

To simulate data stream with pseudo data instead of real human voice or video we need to simplify the described model to meet our needs. The most important variables are Jitter, Round-Trip Time and Packet Loss.

- Jitter - According to [3] "Interarrival jitter is an estimate of the statistical variance of the RTP data packet interarrival time measured in timestamp units and expressed as and unsigned integer". Exact jitter calculation is defined in [3] by Equation (2):

  $$J(i) = J(i-1) + \frac{(|D(i-1,i)| - J(i-1))}{16}, \tag{2}$$

  where J (i) is Jitter for packet i, D is the difference in packet spacing at the receiver compared to the sender for a pair of packet given in [3] by equation (3):

  $$D(i,j) = (Rj - Ri) - (Sj - Si) = (Rj - Sj) - (Ri - Si), \tag{3}$$

  where *Si* is the RTP timestamp of packet *i* and *Ri* is the time of arrival of packet *i* in RTP timestamp [3].

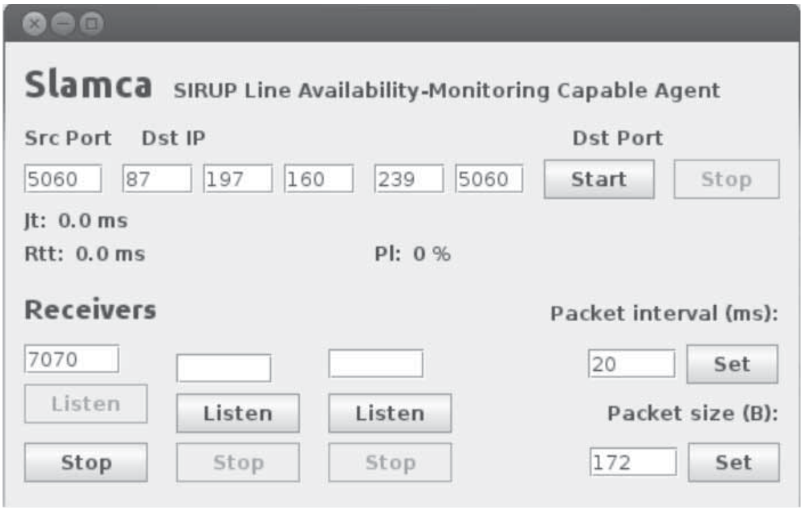
- Round-Trip Time - Similar to latency determined by using `ping` - measures the time between sending packet and receiving confirmation. Evaluation of Round-Trip Time has to be either implemented manually, or by using one of existing tools and protocols - we decided to implement it manually as other tools might be too complex for our needs.

- Packet Loss Ratio - expressed in percent of how many packets were sent but not received by the second party.

Now when we are able to measure the variables mentioned above, we can calculate MOS easily by using the Equation 4.

$$MOS = a + bx + cy + dx^2 + ey^2 + fxy + gx^3 + hy^3 + ixy^2 + jx^2y, \tag{4}$$

where *x* is the packet loss ratio and *y* is jitter. The parameters are dependent for each codec - two examples are given in table below [1]. It is important to be aware of the fact that MOS strongly depends on codec.



Slamca SIRUP Line Availability-Monitoring Capable Agent

Src Port 5060
Dst IP 87 197 160 239
Dst Port 5060

Jt: 0.0 ms
Rtt: 0.0 ms
Pl: 0 %

Receivers
7070

Packet interval (ms): 20
Packet size (B): 172

Figure 1. Graphical user interface of SLAMCA agent with settings fields and statistics.

3 Solution description

To manage Quality of Service, we need to monitor all multimedia lines' (MPLS tunnels) quality in real time, where we measure the variables explained above. Therefore we have developed a system that is responsible for monitoring lines and is purposed to notify proxy servers about line outages and congestions.

The system consist of three main parts. SIP Single Port UDP proxy named SIRUP; core network topology with MPLS Traffic Engineering implemented; and software responsible for measuring session quality named SIP Single Port Line Availability Monitoring Capable Agent (SLAMCA). As SIRUP is being developed by another team we cooperate with, in this article we will focus on MPLS network and SLAMCA.

3.1 SLAMCA

One SLAMCA agent generates pseudo traffic similar to desired codec - we can explicitly set the size of the packet as well as the interval between sending individual packets. SLAMCA agents run on both sides of lines, which is needed to confirm received packets since UDP packets are unconfirmed. Now agents are able to to calculate Jitter, Packet Loss and Round-Trip Time in real time. The design of SLAMCA agents allows to monitor multiple multimedia lines simultaneously.

As depicted in Figure 1, user interface is meant to be as simple as possible, showing results live while sending test samples. One instance of SLAMCA can be both sender and receiver at the same time. SLAMCA is implemented in Java. We use standard Java libraries such as `DatagramPacket` and `DatagramSocket` among others. As shown in Test-Beds, sending packets using these libraries brought results that satisfied our needs. SLAMCA is meant to be cross platform, no installation is required. SLAMCA is runnable Java archive `*.jar`.

4 Test-beds

To test whether shaping traffic in MPLS tunnels can bring improvement in call quality, we simulated the providers network in GNS3, where we created the topology depicted in Figure 2. In this topology

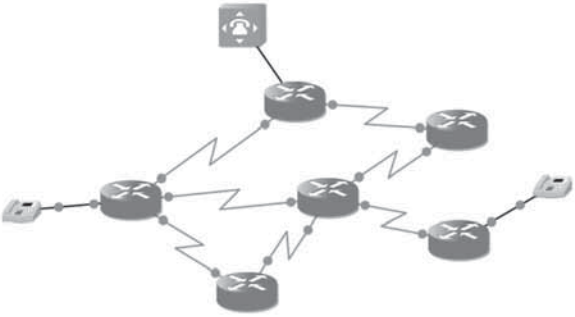


Figure 2. Simulation network topology with two participants with configured MPLS-TE tunnels.

Table 3. Tests results - one line for both calls.

	1 call	2 parallel calls	Difference in %
Average maximal jitter	16.877 ms	25.074 ms	48.56
Average mean jitter	10.854 ms	12.574 ms	15.847

we created multiple tunnels with limited bandwidth to simulate line congestion. Each tunnel's bandwidth was slightly more than needed for one voice call using standard codecs such as iLBC or G.711. In the first scenario we were transmitting two parallel calls in separate tunnels, in the second scenario we transmitted both calls via one congested tunnel simultaneously. Each scenario consisted of 5 tests where only one session was active at the time, and 5 tests with two simultaneous sessions at the same time. Results for the first scenario are shown in Table 3.

Results of the second test scenario where each call was transmitted via its own uncongested line are in Table 4. As can be seen from the results, even though we used the same topology in both cases, thanks to traffic shaping we are able to avoid the risk of congestions just by load balancing multimedia sessions within the tunnels configured in our network. This difference can be seen in the charts below (Figure 3), where we can see the difference between average mean jitter in both simulation scenarios.

5 Conclusion

Test results confirmed the added value of implementing advanced traffic shaping in provider networks, as it allows us to use the network more efficiently. Our system is unique because we generate our own traffic similar to the one generated by any of the existing codecs and measure defined variables

Table 4. Tests results - each call transmitted via separate line.

	1 call	2 parallel calls	Difference in %
Average maximal jitter	18.156 ms	22.2 ms	22.274
Average mean jitter	12.527 ms	12.822 ms	2.355

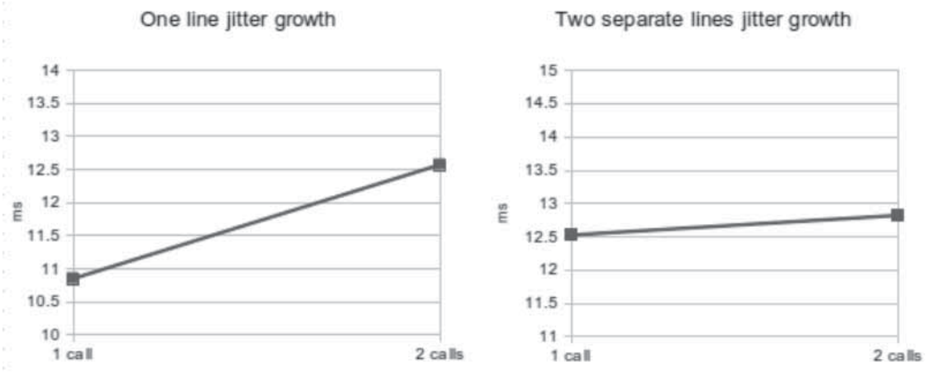


Figure 3. Charts showing difference of average mean jitter on congested and uncongested line.

to determine session quality. As a future work we would like to implement advanced cooperation with SIRUP proxy servers, so that the network will be independent and will be able to react on quality changes itself, which may improve quality of service provided to end customer.

References

[1] Direct estimation of MOS based on the packet loss rate and delays. `http://netserver.ics.forth.gr/wiki/images/3/3a/VOIP_MOS_models.pdf`, [Online].

[2] MPLS Traffic Engineering - Cisco Systems. `http://www.cisco.com/en/US/docs/ios/12%200s/feature/guide/TE%201208S.html`, [Online].

[3] RFC 3550 - RTP: A Transport Protocol for Real-Time Applications. `http://www.ietf.org/rfc/rfc3550.txt`, [Online].

[4] The E-model - E-model Tutorial. `http://www.itu.int/ITU-T/studygroups/com12/emodelv1/tut.htm`, [Online].

[5] ITU-T - G.107: The E-model, a computational model for use in transmission planning. `ttps://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.107-199812-S!!PDF-E&type=items`, [Online].

[6] ITU-T - P.800.1: Mean Opinion Score (MOS) terminology. `https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.800.1-200303-S!!PDF-E&type=items`, [Online].

[7] Mesjar, I.T.K.I.P.: CCIE R&S Techtorial - MPLS. `http://www.cisco.com/web/SK/expo2011/pdfs/CCIE%20Boothcamp%20MPLS%20Peter%20Mesjar.pdf`, [Online].

[8] Murányi Ján, Kotuliak Ivan, N.J.: Simplifying the Session Management using SIP Single Port. *NGMAST 2013 : the Seventh International Conference on Next Generation Mobile Applications, Services, and Technologies, 25-27 September 2013, Prague. - Los Alamitos, IEEE Computer Society, 2013. - ISBN 978-0-7695-5090-9. - S. 148-152.*

Security in Mobile Ad Hoc Networks

Jozef FILIPEK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xfilipekj1@is.stuba.sk`

Abstract. In this paper, we define distributed firewall architecture that is designed specifically for MANET networks. Our design is using the concept of network capabilities and is especially suited for environment which lacks centralized structure and is composed of different devices. Our model denies all communication by default and nodes can access only services and other nodes that they are authorized to. Every node contains a firewall mechanism which includes intrusion prevention system and compromised node will not necessarily compromise whole secured network. Our approach should add security features for MANETs and help them withstand security threats which would otherwise damage, if not shutdown unsecured MANET network.

1 Introduction

MANETs are well known networks, which do not rely on fixed infrastructure. Their popularity lies in the decentralized nature and flexibility. They can be deployed without any costs spent on the infrastructure, like access points and antennas. Nowadays we can find these networks in various conferences, military tactical situations and emergency rescue operations. MANET's greatest advantage, flexible and decentralized nature, also proposes many security problems. MANET is vulnerable to both insider and outsider attacks [3], because it lacks well-defined perimeter in which firewall, intrusion detection system and other security systems would be deployed.

Basic concept of the firewall is to deploy it on the entry/exit point of the network. Since MANET does not have fixed infrastructure, every node acts as a router, therefore every node has to have some kind of firewall mechanism. The question is how to distribute firewall mechanism to every other node of the network. In the last years, several approaches have been introduced, which try to provide MANET with distributed firewall:

- Routing as the Firewall Layer [1] uses deployed routing protocols in MANET (AODV, OLSR) which were modified to act as a firewall. This approach does not consider how to propagate firewall mechanism to unsecured nodes, nor how to deal with compromised nodes. Those are main drawbacks of this solution
- Distributed Firewall for MANETs [2] uses Central Authority (CA) to propagate firewall mechanism across whole network. Network capabilities are used as said mechanism. They limit communication radius and services nodes can use. This approach uses RSA

[*] Master degree study programme in field: Computer Engineering
Supervisor: Assoc. Professor. Ladislav Hudec, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

cryptography for secure communication between nodes. Drawbacks of this solution are unsecured CA and missing PKI.

In this paper we introduce our concept of Distributed Firewall for MANETs.

2 Related work

As mentioned before, there are two completely different approaches how to implement firewall mechanism into MANET environment. First is Routing as the Firewall Layer [1]. This solution uses modified routing protocol as firewall mechanism. It is fast and effective solution, since filtering of traffic and firewall advertisements are happening on routing layer. However, this concept does not deal with compromised nodes or distributing firewall mechanism to unsecured nodes. One compromised node could immediately diminish security of the whole network. Second approach, Distributed Firewall for MANETs [2], uses CA for distributing firewall mechanism and network capabilities as means for limiting communication radius and services nodes can use. On top of that, this concept uses asymmetric RSA cryptography for encrypting communication and signing capabilities. Compromised nodes can affect only nodes within its own communication radius, so they will not be able to compromise security of the whole network. Additionally, this approach does not protect against compromising of CA, which can affect every secured node. Based on the analysis of these two solutions, we decided to base our firewall concept on the second [2] approach.

3 Proposed solution

After the analysis we were able to identify critical parts of our firewall concept for MANETs:

1. Distribution of firewall mechanism – for this purpose we decided to use CA, which would distribute network capabilities to all secured nodes. CA knows beforehand which nodes to distribute capabilities to.
2. Asymmetric cryptography – initial handshake between nodes, where they exchange capabilities is achieved with RSA algorithm. With this we achieve secured communication against eavesdropping and modification attacks.
3. Symmetric cryptography – during initial handshake devices exchange shared password that will be used for encrypting subsequent data sending. Using symmetric cryptography is less power and processing consuming than asymmetric cryptography.
4. Network capabilities – capabilities serve as firewall mechanisms to allow communication between secured nodes. They define who can nodes communicate with, how much bandwidth they can use and what services they can use. Capabilities are issued by CA (forward capability) and by individual nodes (secondary capability), each with different impact on the network.
5. Databases – our distributed firewall needs three databases to work as intended. Database of network capabilities, transactions and compromised nodes. Each has its own purpose and these databases will be described in later chapter.
6. Secured communication – each network capability is signed with RSA algorithm and during initial handshake is encrypted with RSA algorithm. Subsequent data communication is encrypted using symmetric cryptography.
7. Any communication which is not allowed is by default denied.

Every node in a network has to be able to use asymmetric and symmetric cryptography. It even has to store information about network capabilities and captured traffic in its databases.

3.1 Network capabilities

Network capabilities are key feature of our proposed firewall solution. They define communication restrictions on the network. We use two types of network capability:

1. Forward capability – Issued by CA. Signed with CAs private key (ideally 1024 bits long). Has long term validity, up to one hour.
2. Secondary capability – Issued by individual nodes. Signed with nodes' private key (128 bits long). Short term validity. Valid for few minutes or one session only. Contains transaction ID, which is used in the communication session for faster packet processing and lower overhead.

   Following is the example of said forward capability:

serial: 130745	*/ capability ID*
owner: unit01.nj.army.mil (public key)	*/ capability owner*
*destination: *.nj.army.mil*	*/ allowed communication radius*
service: https	*/ allowed communication service*
bandwidth: 50kbps	*/ maximum allowed bandwidth*
expiration: 2013-10-30 23:59:59	*/ duration of capability validity*
issuer: captain.nj.army.mil	*/ issuer of capability*
signature: sig-rsa 23455656767543566678	*/ capability signature*

3.2 Databases

We are using three different databases in our distributed firewall:

1. Capability database – this database is used for storing forward and secondary capabilities. Secondary capabilities are stored only for the duration of session, but forward capabilities are stored until validity of the capability expires.
2. Transaction database –this database stores traffic sessions flowing through node and traffic sessions node can capture. Here is where our IDS takes place. Node evaluates traffic and if any node is violating rules given by capability, it can add node to the Compromised nodes database. Since nodes do not trust each other, identities of compromised nodes are not shared. Following is the example of transaction entry:

ID	*/ entry ID*
Source node	*/ capability owner*
Destination node	*/ communication destination node*
Transaction id	*/ communication transaction ID*
Capability	*/ capability copy*
Usage	*/ bandwidth, session duration*

3. Compromised nodes database – list of nodes, which IDS evaluated as compromised. Based on the severity of violation, node can be removed from this database. If node violates bandwidth usage or allowed service, this is most severe case, than is permanently added to database. If some node happen to abuse his bandwidth constraints, i.e. using it often to the limit, then is added to the database temporarily, because IDS mechanism cannot be sure if node is compromised or not.

3.3 Communication between secured nodes

Secure communication consists of two steps:

1. Forward capabilities allocation
2. Connection establishment

3.3.1 Forward capabilities allocation

This is procedure, which is done before forming the network. CA has beforehand knowledge of nodes to which send capabilities. CA sends forward capabilities signed with its own private key. Along with capabilities CA sends its public key, so individual nodes can verify that capability has not been tampered with. After this allocation, CA goes into offline state. Communication between CA and nodes is one way. CA may go into online state later to add new nodes into secured network. With this, CA will renew capabilities and with it public keys. This renew procedure will happen at least once an hour to keep increased security of used 1024 bit encryption of forward capabilities.

3.3.2 Connection establishment

Consider 2 communicating nodes, Transmitter and Receiver (Figure 1). If Transmitter wants to communicate with Receiver, it sends request. First request always fails, because nodes exchange public keys, which will be used for encrypting communication. Any subsequent communication is encrypted with nodes' respective public keys. Request consists of forward capability and secondary capability. Secondary capability defines different bandwidth, it can be more than forward capability defines, destination can be only one node and duration of validity is much shorter. Intermediate nodes verify capabilities and add them to their respective databases. After verification, they forward them. On receiving capabilities Receiver verifies them. If successful, Receiver sends its own forward capability, Transmitters' secondary capability and his own capability, which can match Transmitters' requested bandwidth or can be different. If packets travel back using different route, intermediate nodes verify capabilities and forward them. Upon receiving, Transmitter verifies capabilities and sends back to Receiver his own capability, consulting Receiver's secondary capability. Intermediate nodes and Receiver verify and add this capability to their respective databases.

During this handshake, communication nodes exchange their shared keys. Shared keys are used for encryption of any subsequent data transfer.

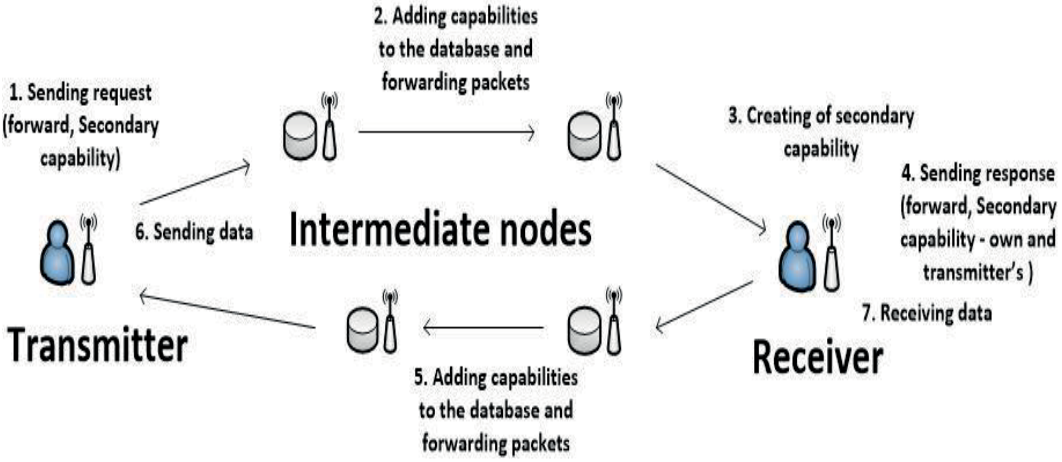

1. Sending request (forward, Secondary capability)
2. Adding capabilities to the database and forwarding packets
3. Creating of secondary capability
4. Sending response (forward, Secondary capability - own and transmitter's)
5. Adding capabilities to the database and forwarding packets
6. Sending data
7. Receiving data
Intermediate nodes
Transmitter
Receiver

Figure 1. Communication between secured nodes.

3.4 Control packets

Control packets are used for exchanging information about network capabilities. Data packets can be piggybacked onto control packets.

CAP-REQ message is used for establishing an entry in the capability database of the nodes along the path from a source to a destination. The message contains the source node address (IDi), transaction id (TXir), destination node address (IDr), flags and the capability (C). This message is signed with the private key of the sender.

SEC-CAP message is used by a node to send its secondary capability.

CAP-ERROR message is used for sending error messages.

PUB-KEY message is used for sending public key.

SYM-KEY message is used for sending shared symmetric key.

DATA message is used to piggyback data on control messages.

CAP-INFO message is used for requesting information about unknown capabilities (during route change).

Figure 2 shows database changes in intermediate nodes and connection establishment between communicating nodes.

3.5 Data transfer and routing changes

The data packets are of following format:
IDi ,IDr ,TXi , < data >, < signature >
An intermediate node verifies the packet against the associated capability before forwarding it. It also probabilistically verifies the packet signature to prevent spoofing attacks.
If any node accepts data packets and do not have transaction identifier associated with capability database entry, it send message to the node it received packet from requesting missing capability.

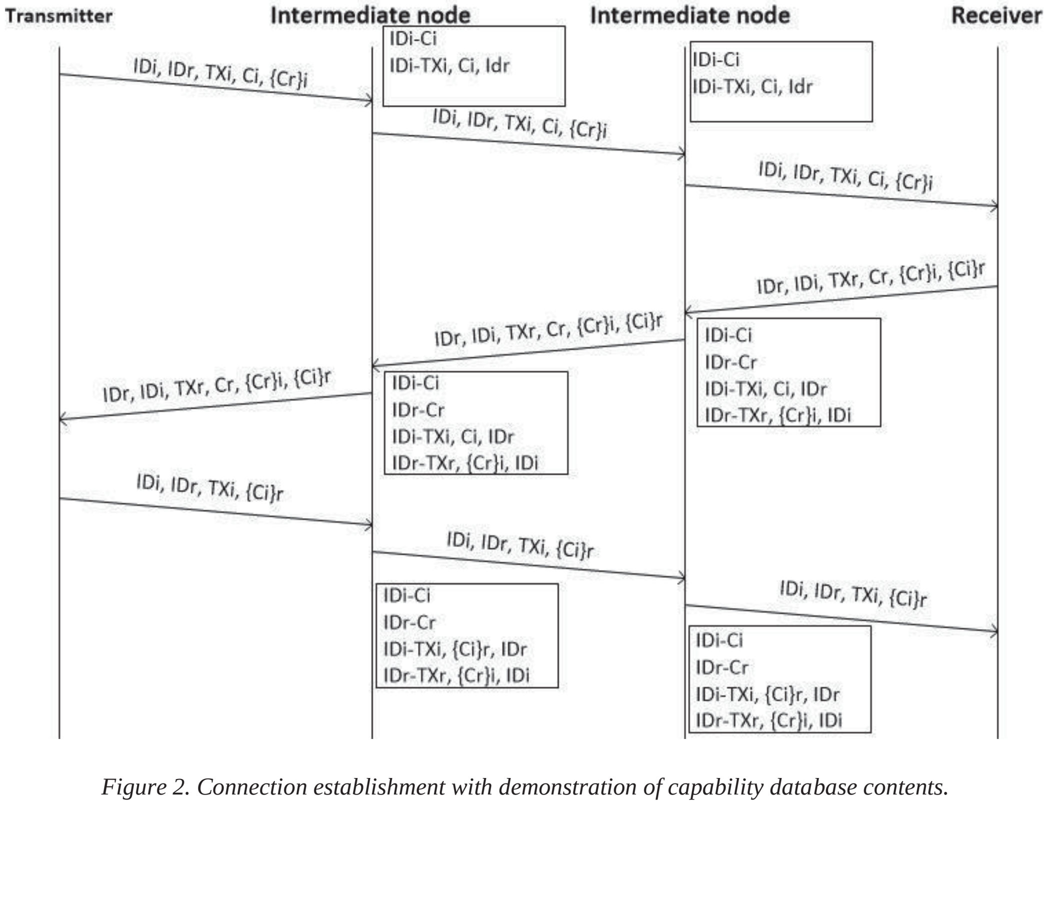


Figure 2. Connection establishment with demonstration of capability database contents.

4 Expected results

Our proposed solution of distributed firewall for MANET should be able to defend against these attacks:

1. Eavesdropping communication between nodes (except routing information) – all communication is encrypted using asymmetric and symmetric cryptography
2. DoS attack of unsecured nodes aimed at the secured nodes – communication from unsecured nodes is blocked by default
3. DoS attack of compromised nodes against secured nodes – if secured node detects capability constraints violation, it instantly drops every communication originating from compromised node
4. Attack of compromised nodes again secured nodes – this attack does not violate any constraints. Its detection is solely based on abusing capability constraints.

Using capabilities in connection establishment and encrypting all communication should increase overall network overhead. Generally the encryption processing will stress devices more and discharge them more quickly, than when not used with encryption.

5 Conclusions

We have presented concept of distributed firewall for general MANET with IPS capabilities. Our solution adds additional security to MANET and protects it against attacks that would otherwise bring the whole network down. We used concept of network capabilities and central authority for distributing firewall across network. We extended existing solution [2] and made several changes to it. We added symmetric cryptography for data encryption and simple IDS for individual nodes which evaluates captured traffic. Added security comes with a price. Using additional information during connection establishment and encrypting all communication proposes increased overhead on the network and additional computing and storage load on the device, leading to faster discharge rate. However, this increased load should not be fatal four battery powered nodes. Our solution does not protect against public key hijacking during initial handshake nor does protect against compromising of CA. Our concept will be tested based on several scenarios. Scenarios will test basic functionality (handshake and data transfer), functionality with moving nodes (route changes) and functionality while being under attack by secured or unsecured nodes. Whole verification will be done and implemented in Omnet++ [4] network simulator.

References

[1] Zhao, H., Bellovin, S.M.: High Performance Firewalls in MANETs. In: *MSN '10 Proceedings of the 2010 Sixth International Conference on Mobile Ad-hoc and Sensor Networks*, (2010), pp. 154-160. 978-0-7695-4315-4.

[2] Alicherry, M., Keromytis, A.D., Stavrou, A.: Evaluating a Collaborative Defense Architecture for MANETs. In: *IMSAA'09 Proceedings of the 3rd IEEE international conference on Internet multimedia services architecture and applications*, NJ, USA : IEEE Press Piscataway, (2009), pp. 229-234. 978-1-4244-4792-3.

[3] Hoebeke, J., Moerman, I., Dhoedt, B., Demeester, P.: An Overview of Mobile Ad Hoc Networks: Applications and Challenges. *Journal of the Communications Network*, vol. 3, (2004), pp. 60-66.

[4] *OMNeT++ Network Simulation Framework.* [Online; accessed January 3, 2013]. Available at: http://www.omnetpp.org/.

Sensor System for Monitoring of Environment Characteristics

Valéria HARVANOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
valeria.harvanova@gmail.com

Abstract. The sensor system for monitoring of environment characteristics is a crucial element of home automation. We have analysed possibilities for monitoring system implementation that would be highly customisable, cost-effective and sufficiently reliable. We have designed home monitoring system that uses ZigBee wireless sensor network with 8-bit microcontrollers and OS Linux based gateway. User defined code can be added and new sensors can be connected to microcontroller's input/output ports. The evaluation of our design is presented together with results of measurements of ZigBee performance in 2.4 GHz wireless spectrum within a building.

1 Introduction

The aim of our work is to design, implement and evaluate a monitoring system of home environment characteristics. Such system can be characterized as follows: it is system collecting environment state information by sensors; it aggregates and stores monitored information; it evaluates collected data and produces feedback if needed; the measured data are accessible anytime via user interface. Furthermore, we identified the following desirable properties of the monitoring system: wireless communication, battery-powered sensors, measured data available anytime via Internet, high availability of system's components (low cost, standards-based), its high customizability by the user.

Our research of commercially available solutions showed that commercial readymade monitoring systems are generally high-priced and not very well customizable. Modular commercial solutions may require paid configuration and installation services, their customization capabilities tend to be limited or they require specific knowledge to implement and customize the system. We also analyzed and compared available wireless low-power protocols. We found the following protocols to be appropriate to be used for environment monitoring: 6LoWPAN, ZigBee IP, ZigBee, ONE-net, DASH7, EnOcean Wireless. Comparison of low-power wireless protocols and short review of studied commercial solutions is available in related diploma thesis.

Based on the requirements specified above and availability of protocol implementations and hardware we decided to design and implement ZigBee monitoring system operating in 2.4 GHz

[*] Master degree study programme in field: Computer Engineering
Supervisor: Assoc. Professor Tibor Krajčovič, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 541–546.

wireless spectrum. For other parts of the system we use Linux personal computer and free online repository services to store and visualize data.

In this paper we provide quick overview of our monitoring system design, present results of our measurements and discuss important aspects of wireless monitoring systems.

2 Design of monitoring system

Monitoring system consists of the ZigBee sensor network, the gateway and the cloud. Sensor network consists of battery powered sensor nodes whose sole job is to sense and upload data, main-powered routers whose job is to relay messages and the gateway that collects data and controls the network. Gateway processes, evaluates and stores data. It uploads summary information to the cloud. User can access data and manage the network via user interface on the gateway. Graphical data representation and statistics are available on the server. The high-level system design is depicted in Figure 1.

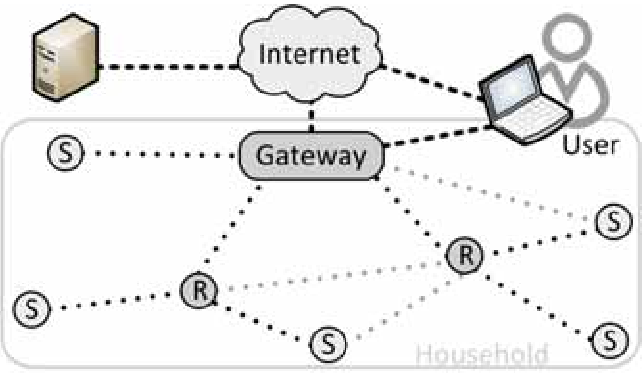


Figure 1. High-level design of monitoring system.

2.1 Hardware design

Hardware design is based on Microchip's PICDEM Z Demonstration Kit [1]. Low-power 8-bit PIC18LF4620 microcontroller with MRF24J30MA 2.4 GHz transceiver is used as the basis of sensor network. It is clocked by 4 MHz crystal and powered via 9 V DC power source. ZigBee coordinator is connected via RS-232 to PC or server running Linux OS. Together they act as the gateway. We decided to use RaspBerry Pi model B with Debian OS to implement gateway's functionality. The PC must be connected to the Internet.

Microcontroller's input/output pins are available for connecting of desired sensors. Their role is determined by the user defined code. Analog input and 1-Wire interface will be used by our demo sensor node. Available power source will be 3.3 V and 5 V.

2.2 Software design

Sensor network is a non-beacon ZigBee Feature Set [2] sensor network with the cluster-tree topology. Microchip's ZigBee-2006 Residential stack protocol implementation in C language is used [3]. Due to requirements of low price and ease of customization we have decided not to implement any public profile specified by the ZigBee Alliance. This would add more complexity to the code and also require more resources. However, this decision means that our ZigBee network will not be interoperable with commercial devices (as far as they are not appropriately reprogrammed). Our proposed protocol uses ZigBee features like Endpoints, Groups and Clusters and adds variable data payload. One payload entry consists of length, type, and data fields. Data can be information with timestamp or command payload. Gateway communicates using commands and sensor nodes communicate using data messages.

Sensor node is usually in low power mode and wakes up only to measure, request and send data. Sampling and sending rates are user defined. Sensor node periodically transmits all previously not uploaded measured data. It stores certain amount of not successfully transmitted historical data to maintain integrity during possible offline mode caused by broken wireless link. Sensor's measuring functionality is performed by user-defined code. Typical sensor communication protocols like SPI or 1-Wire are available. Routers and gateway are always on and they can operate as sensor nodes as well. Routing functionality is performed by ZigBee stack. Coordinator's firmware relays received data messages out via RS-232 interface.

The source code is to be well structured and commented to ensure fast learning curve and ease of use. The sections that have to be modified when a new sensor is being attached should be emphasized. Process of sensor addition will be documented in detail in user's guide.

Linux computer's core functionality is to provide a user interface to data and to the network via web page, receive and store measured data and periodically upload them to the cloud. Computer's tasks will be performed by a set of scripts and commonly used Debian programs. All source code should be portable.

3 Evaluation

This section evaluates important aspects of wireless sensor networks. We also present results of experimental evaluation of IEEE802.15.4 wireless communication.

3.1 Coexistence in the 2.4 GHz band

Both IEEE 802.11 (WiFi) and IEEE 802.15.4 (protocol on which ZigBee is built) devices operate in 2.4 GHz ISM band. Other devices like cordless phones and microwave ovens can also be sources of interference. Given ZigBee's characteristics and its typical low output power the message delivery rate can be considerably reduced if interference occurs. To maximize the reliability of measurement data, care must be taken to ensure coexistence. Primary mitigation technique is channel separation. If spectrum is not fully utilised by IEEE 802.11 traffic, a suitable non-overlapping 2 MHz wide IEEE 802.15.4 channel can usually be chosen from 16 possible channels. Other technique is physical separation. It has been experimentally proven that the distance of the order of meters is generally sufficient to mitigate negative impact of interference [4]. However, strong overlapping IEEE 802.11 network with high duty cycle should be avoided since it can dramatically degrade ZigBee network performance [5].

It is important to note that even in moderately interfering environments ZigBee network can function with suitable performance. The duty cycle of a ZigBee device is usually very low and messages are short (maximum length is 127 B). Therefore relatively few short packets need to be transmitted and using CSMA-CA medium access method the likelihood of an unsuccessful transmission is reduced. Packet can be retransmitted if an acknowledgement is not received. However, if frequent retransmissions are needed, the power consumption will be noticeably increased.

3.2 Range tests

One of the most important steps in implementing a wireless monitoring system is planning out the location of sensor nodes, routers, and gateways. It is important to understand how the environment can influence the wireless signal. Maximum transmission distance depends on power output, receiver's sensitivity and environment characteristics.

Physical effects and phenomena affecting signal propagation are: penetration, reflection, scattering and diffraction on various objects in the signal path. Besides the distance law on the propagation loss in the environment, other factors cause the variation in propagation loss (e.g. the antenna height-gain, depolarization effect, interference with other networks or devices operating on the same frequency or on harmonic frequencies etc.). All these phenomena affecting signal attenuation caused by given environment can be covered with one value that is called the *path-loss exponent* (n). Using this value, we can estimate the remaining signal power at distance d using equation (1) where P_d is the signal power (in dBm) at distance d from the antenna (in meters), P_0 is the signal power (in dBm) at zero distance from the antenna and f is the frequency (in MHz) [6]. For free space n equals 2.

$$P_d = P_0 - 10.n.log_{10}(f) - \ 10.n.log_{10}(d) + 30.n - 32.44 \tag{1}$$

To examine performance and maximum transmission distance of our monitoring system we have performed measurements using Microchip's MRF24J40 2.4 GHz transceiver with PCB antenna. In this article we present results of measurements from two different scenarios. We measured distance, received signal power and number of lost messages. Detailed summary of measured and computed data for scenario A is available in Table 1. Short summary for scenario B is in Table 2.

Measurement in scenario A was performed in an inhabited flat with panel walls and floor, wooden doors and metal door frames. IEEE 802.11n interference was present with signal strength ranging from -59 dBm near coordinator to -70 dBm. Measurement in scenario B was performed in an inhabited completely wooden house with walls of chipboard, glass wool and plasterboard, with wooden floor, doors and door frames. No IEEE 802.11 interference was present.

In both scenarios we measured at different places near a wall. We used channel 26 (2480 MHz) and 0 dBm output power. In scenario A we measured up to 20 acknowledged transmissions at five different positions at each place. Positions 1 to 3 were within 15 cm radius on the floor, positions 4 to 5 were in different height and within 10 cm radius. In scenario B we measured 10 successful transmissions in several different positions at each place. Positions were within 15 cm radius in varied heights. Small measurement error was probably introduced by the fact that the device used to generate messages was not firmly attached to the surface and position of the antenna was being unintentionally changed in the order of millimetres during measurement.

Table 1. Summary of measured and computed data for scenario A.

<table>
<tr><th>d
[m]</th><th>Position
[#]</th><th>Min Pd
[dBm]</th><th>Max Pd
[dBm]</th><th>Average
Pd
[dBm]</th><th>Message
LOSS
[%]</th><th>ACK
LOSS
[%]</th><th>Err.[1]
[%]</th><th>n</th><th>Theoretical
max range
[m]</th></tr>
<tr><td>6,00</td><td>1</td><td>-90</td><td>-84</td><td>-87.3</td><td>55</td><td>0</td><td>55</td><td>4.681</td><td>8.33</td></tr>
<tr><td></td><td>2</td><td>-79</td><td>-78</td><td>-79.5</td><td>35</td><td>0</td><td>35</td><td>3.925</td><td>14.93</td></tr>
<tr><td></td><td>3</td><td>-95</td><td>-94</td><td>-94.4</td><td>25</td><td>40</td><td>95</td><td>5.286</td><td>5.89</td></tr>
<tr><td></td><td>4</td><td>-76</td><td>-68</td><td>-73.1</td><td>35</td><td>0</td><td>35</td><td>3.466</td><td>24.09</td></tr>
<tr><td></td><td>5</td><td>-94</td><td>-78</td><td>-85.2</td><td>25</td><td>5</td><td>35</td><td>4.501</td><td>9.41</td></tr>
<tr><td colspan="10">Condensed information about other measurements:</td></tr>
<tr><td rowspan="2">d
[m]</td><td colspan="2">#1</td><td colspan="2">#2</td><td colspan="2">#3</td><td colspan="2">#4</td><td>#5</td></tr>
<tr><td colspan="2">Pd | n | Err.</td><td colspan="2">Pd | n | Err.</td><td colspan="2">Pd | n | Err.</td><td colspan="2">Pd | n | Err.</td><td>Pd | n | Err.</td></tr>
<tr><td>3,40</td><td colspan="2">-68.4 | 3.89 | 10.0</td><td colspan="2">-70.1 | 4.07 | 0.0</td><td colspan="2">-68,0 | 3.84 | 10.0</td><td colspan="2">-55.5 | 2.49 | 0.0</td><td>-62.0 | 3.19 | 0.0</td></tr>
<tr><td>4,10</td><td colspan="2">-76.6 | 4.38 | 15.8</td><td colspan="2">-78.5 | 4.57 | 5.0</td><td colspan="2">-75,9 | 4.32 | 17.6</td><td colspan="2">-76.6 | 4.38 | 6.7</td><td>-67.5 | 3.48 | 30.0</td></tr>
<tr><td>4,10</td><td colspan="2">-78.3 | 4.55 | 5.0</td><td colspan="2">-90.6 | 5.78 | 30.0</td><td colspan="2">-83.4 | 5.05 | 15.0</td><td colspan="2">-70.0 | 3.73 | 0.0</td><td>-75.5 | 4.28 | 10.0</td></tr>
<tr><td>6,25</td><td colspan="2">-83.8 | 4.31 | 20.0</td><td colspan="2">-77.8 | 3.81 | 0.0</td><td colspan="2">-91.0 | 4.92 | 15.0</td><td colspan="2">-73.0 | 3.41 | 25.0</td><td>-73.6 | 3.45 | 20.0</td></tr>
<tr><td>7,55</td><td colspan="2">-77.2 | 3.51 | 35.0</td><td colspan="2">-76.9 | 3.49 | 30.0</td><td colspan="2">-73.8 | 3.25 | 55.0</td><td colspan="2">-74.8 | 3.33 | 50.0</td><td>-71.1 | 3.04 | 20.0</td></tr>
<tr><td>7,55</td><td colspan="2">-88.9 | 4.44 | 45.0</td><td colspan="2">-85.9 | 4.20 | 50.0</td><td colspan="2">-84.1 | 4.06 | 50.0</td><td colspan="2">-80.0 | 3.74 | 45.0</td><td>-80.5 | 3.78 | 40.0</td></tr>
<tr><td>8,20</td><td colspan="2">-89.2 | 4.34 | 55.0</td><td colspan="2">-93.4 | 4.66 | 70.0</td><td colspan="2">-90.9 | 4.47 | 50.0</td><td colspan="2">-81.9 | 3.78 | 25.0</td><td>-79.2 | 3.57 | 50.0</td></tr>
<tr><td>8,40</td><td colspan="2">-84.8 | 3.97 | 30.0</td><td colspan="2">-91.5 | 4.48 | 90.0</td><td colspan="2">-86.6 | 4.11 | 20.0</td><td colspan="2">-79.9 | 3.60 | 10.0</td><td>-81.3 | 3.70 | 5.6</td></tr>
<tr><td>10,20</td><td colspan="2">-92.0 | 4.25 | 85.7</td><td colspan="2">-88.4 | 3.99 | 30.0</td><td colspan="2">-90.5 | 4.14 | 25.0</td><td colspan="2">-94.9 | 4.45 | 100</td><td>-90.0 | 4.10 | 85.0</td></tr>
</table>

[1] "Err." is the proportional representation of lost messages and lost or incorrectly received acknowledgments.

Table 2. Summary of measured and computed data for scenario B.

d [m]	Pd | n	Pd | n	Pd | n	Pd | n	Pd | n
6,80	-62.3 | 2.43	-65.1 | 2.66	-64.0 | 2.57	-64.2 | 2.59	-57.2 | 2.02
	-60.1 | 2.25	-60.9 | 2.25	-57.6 | 2.05	-	-
12,50	-73.0 | 2.72	-80.8 | 3.24	-78.6 | 3.10	-85.3 | 3.54	-78.4 | 3.08
	-75.8 | 2.91	-71.9 | 2.65	-73.8 | 2.77	-78.6 | 3.10	-
13,00	-81.9 | 3.28	-82.4 | 3.32	-83.5 | 3.39	-89.8 | 3.80	-78.7 | 3.07
	-74.0 | 2.76	-81.0 | 3.22	-70.6 | 2.53	-	-
15,65	-86.1 | 3.38	-86.1 | 3.38	-79.9 | 2.99	-78.8 | 2.91	-85.9 | 3.36
	-84.7 | 3.29	-	-	-	-

Receiver sensitivity of MRF24J40 is -94 dBm. The theoretical maximum ranges using medians of computed path-lost exponents are dMedA=13.2 m and dMedB=46.4 m for scenarios A and B respectively. Maximum range based on the average of lowest path-lost exponents is dAvgMinA=24.5 m using nAvgMinA=3.450 and dAvgMinB=110.0 m using nAvgMinB=2.527 for scenarios A and B respectively. Using the worst measured path-loss exponent with loss rate lower than 50% the values are dMaxA=4.7 m using nMaxA=5.777 and dMaxB=16.8 m using nMaxB=3.803.

Our results demonstrate that correct placement of nodes is very important. The shift of only a few centimetres can significantly improve or impair wireless performance. Path-loss exponent and error rate are highly dependent on actual relative positions of communicating devices. They are influenced by multipath null phenomenon and activity of interferers. Implementation of ZigBee monitoring network in places where interference could be an issue will require denser node placement and use of routers.

Detailed analysis of our measurements is available in related diploma thesis.

3.3 Security

If an attacker accessed data measured and collected by our monitoring system he could be potentially able to use it to his advantage. Therefore the confidentiality of measured data is important. The integrity of messages is important since injected messages would distort gathered information thus possibly aiding an attacker. Device theft is not critical as far as it does not contain any sensitive information.

ZigBee specification defines several modes to protect data communication: message integrity check (MIC) only, encryption only or both. The message frame content generated at the network and higher layers is encrypted using 128-bit AES-based encryption. MIC is generated using enhanced Counter with Cipher Block Chaining Message Authentication Code that uses the same 128-bit shared key as AES. If no shared key has been preconfigured in a joining device it is distributed over an unsecure channel during joining procedure.

The higher security is employed the more complex the computation becomes and more power is consumed. Therefore it is important to choose the least complex security method that is still suitable for the application. Based on the requirements of our system all data communication is encrypted using pre-configured 128-bit network-level key. It is also authenticated using 4 B MIC. The coordinator serves as trust centre operating in residential mode. Since each device contains pre-shared network key, the device theft could be a problem. Connected devices are monitored on the network level. When a device is suddenly lost, gateway notifies the administrator. It could be configured to automatically generate and distribute new network key (using messages secured by the old key). This approach relies on the fact that retrieving the old key from the compromised device is not performed instantly. Disadvantages of this level of security are higher computational complexity and related power consumption and also the necessity to program the network key into each device.

Security of other parts of the system is ensured using standard hardening techniques and secure communication protocols like SSH and HTTPS.

3.4 Battery life

Power consumption is critical especially in battery powered sensor nodes. Actual battery life can be determined if three parameters are known: battery's capacity, average current drained by the node and battery's self-discharge rate [7]. The average current drain can be either estimated by computation or measured. The measurement of our prototype's power consumption is yet to be done. We expect the battery life of a typical sensor node using 9V 800 mAh lithium battery to be in range from 6 to 12 months.

4 Conclusions

We have designed system for monitoring environment characteristics that should be very well customizable by its user. We used ZigBee wireless protocol whose strengths lie in very low power consumption, its popularity among hardware manufacturers, its open status and sufficient level of robustness. We used worldwide available 2.4 GHz ISM spectrum. We performed measurements to empirically evaluate the impact of the environment on wireless signal propagation. We found that a building cannot be characterized by a single path-loss exponent or any narrow range. The propagation characteristics are closely related to relative positions of the nodes. The proper placement of the nodes is thus critical for correct and effective operation of the system. We recommend performing assessment of signal strength and wireless link reliability when connecting a node to sensor network. We stress the importance of selecting a nonoverlapping or not highly used channel for the ZigBee network.

Acknowledgement: This work was partially supported by Grant No. 1/0676/12 of the Slovak VEGA Grant Agency.

References

[1] Microchip Technology Inc.: *PICDEM™ Z ZigBee® and MiWi™ Technology Demonstration Kit (DS51504F)*. (2011).

[2] ZigBee Standards Organization: *ZigBee Specification (Document 053474r17)*. Protocol specification, (2007).

[3] Lattibeaudiere, D. P.: *Microchip ZigBee-2006 Residential Stack Protocol (Application note 1232)*. Microchip Technology Inc., (2008).

[4] Golmie, N.: *Coexistence in Unlicensed Bands - Challenges and Solutions*. [Online; accessed February 20, 2014]. Available at: http://www.ieee802.org/802_tutorials/04-July/802CoexistenceTutorialJuly04a.pdf.

[5] Sikora, A.; Groza, V. F.: Coexistence of IEEE802.15.4 with other Systems in the 2.4 GHz-ISM-Band. In: *Instrumentation and Measurement Technology Conference, 2005, Proceedings of the IEEE*, Ottawa, Canada, (2005), pp. 1786–1791.

[6] Farahani, S.: *ZigBee Wireless Networks and Transceivers*. Newnes, (2008).

[7] Halgamuge, M. N., Zukerman, M. and Ramamohanarao, K., Vu, H. L.: An Estimation of Sensor Energy Consumption. In: *Progress In Electromagnetics Research B. Vol. 12*, (2009), pp. 259–295.

[8] Harvanová, V., Krajčovič, T.: Implementing ZigBee Network in Forest Regions-Considerations, Modeling and Evaluations. In: *Applied Electronics 2011: International Conference on Applied Electronics*, Pilsen, (2011), pp. 149–152.

System on a DaVinci Platform Designed for Visual Check of Circuit Boards

Ondrej KACHMAN*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
ondrej.kachman@gmail.com

Abstract. Embedded systems are widely used for real time tasks, including image processing. For specific tasks, small systems on a chip can be used if they have required computing speed and peripheral devices. The objective of this paper is to present Texas Instruments DaVinci system on a chip and describe how it can be used in area of image processing. This is presented with focus on a specific task - visual check of assembled printed circuit boards. This paper describes libraries and some procedures used to develop image processing applications for DaVinci system running embedded Linux.

1 Introduction

One of the main purposes of computer systems is to help people with their work, to make it easier to complete difficult tasks in a shorter time. Wide tasks can be divided into smaller, more specific tasks. These specific tasks usually do not require as much computational power as bigger tasks, but they also usually have hard deadlines. Embedded systems are systems that are focused to accomplish smaller tasks in the shortest time possible.

This paper focuses on the area of embedded systems, more accurately systems on a chip. These systems on a chip (SoC) have all basic components of a computer located on one single chip and they are able to complete difficult real-time tasks. They are capable of processing audio, radio or video signals in real-time. Processing of this information has low response time. This response time is shorter as faster digital signal processors (DSP) are used for some signal processing algorithms.

One of the greatest SoC manufacturers is a well known company Texas Instruments. This paper describes Texas Instruments DaVinci SoC that combines ARM processor and DSP processor and shows, how it can be used in area of real-time image processing on a specific task. This task is to visually check assembled printed circuit boards for missing components (resistors, capacitors, microcontrollers, etc.) and declare them defective or good.

[*] Master degree study programme in field: Computer Engineering
Supervisor: Boris Dado, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 547–552.

2 DaVinci technology

The DaVinci technology is a SoC video processing platform. The underlying DaVinci hardware has been designed specifically to support video systems, not only serving to reduce board space and component counts, but also to eliminate much of the low-level software development required to integrate a complex system [1]. DaVinci integrates 300 MHz ARM core and 600 MHz DSP core as a part of digital media processors marked as TMS320DM644x. These processors also integrate a video processing subsystem that includes an on-chip image pipeline for camera image capture. Other features relevant to this paper are audio-video interface and USB peripherals.

2.1 DaVinci platform usage examples

Different DaVinci video processors are used for many applications in area of video processing. First introduced into the automotive market more than a decade ago, embedded analytics has become widespread to the point where it is a “must-have” feature on many cars. For example, several vision-based include a lane departure warning system, high-beam assist, traffic sign recognition and forward collision warning system [2].

Security and surveillance systems have also incorporated embedded analytics for quite some time. Besides vision analytics, sound-processing technologies are bringing embedded audio analytics to security applications as well. Alarms can be triggered by sounds of aggression, explosions, sirens, collisions, breaking and other sounds of trouble. DaVinci processors can support many tasks, including face detection, object counting, motion detection etc. [2].

2.2 Application development

DaVinci systems are running embedded Linux operating system on ARM core. Usually, it is MontaVista distribution or Ångström distribution. DSP processor is running its real-time operating system, RTOS. The two processors communicate through the DSP/BIOS Link interprocessor communication software. This low-level communication between the cores enables the developer to work at a high level on the ARM, without having to bother with the DSP or even see into what it is doing [1].

Applications for DaVinci systems are usually written in C++ programming language. Source code is cross-compiled for ARM and DSP processors. Texas Instruments libraries DSPLIB and IMGLIB are written for use with DSP core for fast signal processing. With embedded Linux, multiplatform GTK+ libraries can be used to develop graphical user interface running on SoC.

3 System for visual check of assembled printed circuit boards

As mentioned before, this paper will present DaVinci system on a specific task – visual check of assembled printed circuit boards (PCB). PCBs are assembled automatically and manufactured in hundreds. Before further testing, assembled boards are visually checked and this is a good task for fast video processing embedded system.

First thing that needs to be considered is that this check needs to be fast. Second thing are conditions of image capturing. There can be different light each time, boards may be under different angles, closer to or further from the camera, etc. Captured images need to be matched against one original image of a board with correctly assembled components. For this operation, template matching algorithm is used. Our task can be divided into 4 very basic processes for every image – image capture, image processing, template matching and displaying in the GUI.

3.1 Image capture

Image capture for DaVinci SoC is provided by peripheral camera device connected directly to SoC video processing subsystem. This subsystem is responsible for image capture, scaling and storage into memory. If used camera has too big resolution, it is good to resize captured image directly with video processing subsystem before further processing. This subsystem then saves raw image and pointer to it. This pointer can be accessed by developer to read captured image and process it.

3.2 Image processing

Considering different conditions while capturing images, these images are also different. Circuitc boards can have different positions, they can be rotated and they can have different distance from camera if it was manipulated with between taking pictures. To solve these 3 problems, three operations are executed:

1. Location of a circuit board in the picture
2. Perspective transformation of board area into new picture
3. Scaling

The more of these operations are executed on the DSP processor, the faster will captured image be processed. Two libraries were mentioned before to use with DSP – IMGLIB and DSPLIB. For image processing operations on the ARM core, a well-known library OpenCV is used.

3.2.1 Location of a circuit board in the picture

Operation of locating printed circuit board in the captured image can be accomplished by 4 basic steps:

1. Blur the image
2. Use edge detecting filter
3. Detect contours
4. Find 4 vertices of circuit board area

Blurring the image makes it smoother and edge detecting functions are performed better. After edge picture is created, contour detection functions can be used to create list of two point arrays representing lines in the edge picture. Circuit boards have usually rectangle shape, so to locate it, we are searching for 4 vertices. Searching through contours list, we can find vertices of quadrangle that we expect represents our PCB area in the original picture. For blurring and edge detection, IMGLIB functions *IMG_median_* and *IMG_sobel_* for DSP core can be used, but contour detection is performed using OpenCV library *findContours* function on ARM. With contours found, developer can implement his own way of search for board area vertices.

In the Figure 1, there are three stages of proposed image processing operations. First stage a) displays original image. This is an ideal situation, as there is no background. Stage b) shows image after PCB area location detection. Light grey colour represents all of the detected edges. Darker grey lines connect 4 vertices of printed circuit board area.

3.2.2 Perspective transformation of board into new picture

Detected area becomes the region of interest (ROI). This ROI can be reshaped into new rectangle picture. To do this, perspective transformation algorithm is used. Size of a new picture is given by the two connected sides of ROI. ROI can be transformed using following OpenCV functions:

1. Calculate perspective transformation matrix – *getPerspectiveTransform*
2. Transform ROI into destination image – *warpPerspective*

Perspective transformation 3x3 matrix needs to be calculated first, as it contains translation, rotation, shearing and scaling operations coefficients, that will be used during transformation. Input data of *getPerspectiveTransform* function are 4 points of the original ROI and their positions in the new image. With transformation matrix computed, points of ROI are mapped into the new image using following formula [4]:

$$\begin{bmatrix} t_i x'_x \\ t_i y'_i \\ t_i \end{bmatrix} = map\_matrix \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

where

$$dst(i) = (x'_i, y'_i), src(i) = (x_i, y_i), i = 0,1,2,3 \tag{2}$$

In the Figure 1, stage c), shows the PCB area transformed into the new picture. As original image (a) was taken almost under perpendicular angle and detected area is a trapezoid very close to rectangle shape, no significant deformations can be seen. With rotation of board in the picture and different angles of camera, some deformations after perspective transformation should show.

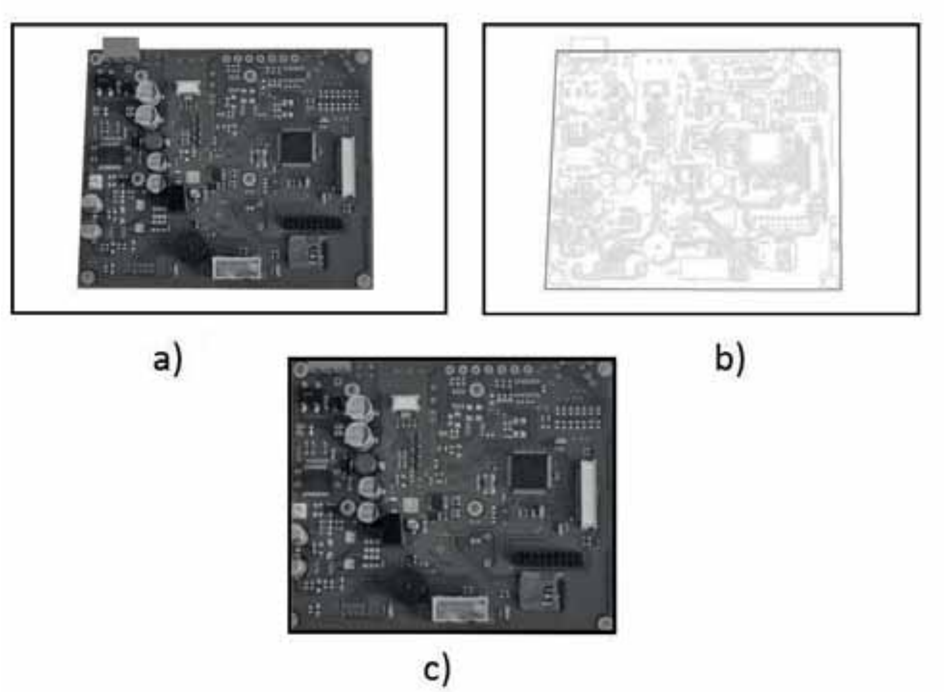


Figure 1. Stages of circuit board image processing.

3.2.3 Scaling

Transformed images may have different sizes after perspective transformation. It depends on size of detected ROI with circuit board. This is solved by simple scaling to the same size for every picture. This operation is very fast using IMGLIB scaling function on a DaVinci DSP core. As this final operation is performed, picture is ready for template matching process.

3.3 Template matching

For template matching, there has to be an example picture of correctly assembled PCB. This picture is captured and processed first. After it is processed with described processes, it can be displayed in the graphical user interface (GUI), where user can mark areas that should be checked in every following picture. After user creates this list of areas, capturing of other images start.

Every image is processed and desired areas are evaluated by OpenCV template matching algorithm.

3.3.1 Creating list of areas to be checked

Adjusted example image can be displayed in graphical user interface (GUI) that runs directly under embedded Linux X window system. DaVinci system includes USB port that enables user to connect mouse controller and interact with GUI. List of areas can be created simply by drawing rectangle areas into the example picture. This list is stored and used later by template matching algorithm. Description of GUI development is described later in this paper.

Feature to choose these areas can save computational time. Circuit boards may have some blank areas that do not need to be checked. This saved time can be used to check the chosen areas and their close surroundings in case some deformations occurred during image processing.

3.3.2 Template matching algorithm

After the example image is processed and the list of areas to check is created, template matching algorithm is applied to every next captured image. This algorithm is performed using OpenCV *matchTemplate* function. Every area in the user created list is copied into new image and serves as a template image. Another image used by matching algorithm is a source image. Source image is a subimage of the captured image cut out in a template area and expanded by a few pixels.

Copying of template and source image consumes some time, but it is faster than running template matching on the whole captured image. For each pair of source and template images, template matching algorithm is executed. The function matches a source image patch against a template image by “sliding” the patch over the source image [3]. This algorithm can use 6 methods to do the matching. The result of a template matching is a matrix of source image size that contains match coefficients. OpenCV function *minMaxLoc* can be used to find the best match value. Threshold value is used to decide, if the checked area will be declared correct or defective.

3.4 Graphical user interface directly under embedded Linux

As mentioned before, embedded Linux can run graphical user interface. Embedded Linux supports X window system – graphical environment. With the use of GTK+ graphical widgets, complex graphical user interfaces can be implemented. DaVinci systems feature VGA video output interface, so X window system can be projected directly to monitor.

For proposed application, easy window capable of displaying two images and some text is enough. GTK+ library provides all the needed widgets and is capable of capturing mouse events needed to implement generation of areas list described in 3.3.1. First image component can display example picture and make user able to draw areas into it, second image can display every other picture after template matching and mark areas as correct or defective using different colours. Also, to inform user about template matching coefficients and additional information, text field widget can be used.

4 Testing

During testing of the described system, image processing and template matching algorithms speed and reliability can be tuned. The speed is influenced by image resolution, amount of detected contours, template matching areas size and count, etc. Using smaller resolutions can speed up all processes but possibly degrade accuracy of template matching results.

Reliability strongly depends on the conditions of image capture. If the conditions of taking a current image are different from the conditions of taking an example image, image processing may fail. Light conditions, camera angle and distance, circuit board rotation and background are the attributes with the greatest impact on proposed algorithm of PCB area location and transformation.

To test speed of proposed image processing (IPR) and template matching (TM) algorithms, 4 static images with sizes of 768 x 512 pixels were used. Circuit board area in every image had different size and template matching was run on different number of checked areas for every picture. Tests were executed on Intel Centrino 2 Ghz Dual Core processor using 2 different operating systems and on DaVinci system. Note, that the DSP/BIOS Link was not used yet, so algorithm was tested only on the ARM core. Because of this, results of embedded system are currently slow, as Table 1 shows.

Table 1. Speed of proposed algorithms measured on different systems.

Operating system	Sizes of the PCBs in the images and number of areas checked							
	609 x 280 4 areas		481 x 399 4 areas		531 x 347 5 areas		730 x 446 7 areas	
	IPR	TM	IPR	TM	IPR	TM	IPR	TM
Linux Mint 32bit (Centrino)	0,06 s	0,02 s	0,06 s	0,02 s	0,07 s	0,02 s	0,06 s	0,02 s
Win 8.1 64bit (Centrino)	0,07 s	0,03 s	0,07 s	0,03 s	0,07 s	0,02 s	0,08 s	0,02 s
Ångström (300 Mhz ARM)	8 s	3,84 s	9,2 s	3,88 s	9,7 s	4,56 s	9,8 s	5,54 s

It depends on a developer, how he will set the resolution of images and threshold values of the template matching. He can even enable user to change these values via GUI. However, image capture conditions are not controllable by system and it may not be able to perform given task correctly if these conditions are too unstable.

5 Conclusion

The DaVinci multimedia processors are widely used in embedded video processing systems. This paper proposed development of an application that uses DaVinci technology to visually check assembled printed circuit boards. It describes a way how images of printed circuit boards can be processed and evaluated using libraries available for the DaVinci SoC and some of their functions. Described solution was implemented and tested. This work can and will be upgraded. As tests have shown, using only ARM core is slow and the real power of DaVinci is hidden in the DSP processor. With a few changes to used algorithms, this system could be modified for similar tasks in other areas besides manufacturing of a printed circuit boards.

Acknowledgement: This work was partially supported by Grant No. 1/0676/12 of the Slovak VEGA Grant Agency.

References

[1] Golston, J., Bhattacharya, R.: *Reaping the Benefits of SoC Processors for Video Applications*. Texas Instruments Incorporated (2007). [Online; accessed February 18, 2014]. Available at: http://www.ti.com/lit/wp/spry096/spry096.pdf

[2] Agarwal, G. et. al.: *“Get Smart” with TI’s embedded analytics technology*. Texas Instruments Incorporated (2012). [Online; accessed February 18, 2014]. Available at: http://www.ti.com/lit/wp/spry201/spry201.pdf

[3] Bradski, G., Kaehler, A.: *Learning OpenCV*. Sebastopol: O’Reilly Media, Inc., 2008. ISBN: 978-0-596-51613-0.

[4] OpenCV dev. team.: *Geometric Image Transformations*. [Online; accessed February 19, 2014]. Available at: http://docs.opencv.org/modules/imgproc/doc/geometric_transformations.html

Application of Software Defined Networking (SDN) in GPRS Network

Peter BALGA, Tibor HIRJAK, Martin KALČOK
Matúš KRIŽAN, Ján SKALNÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
tanker@gmail.com, hirjak.tibor@gmail.com, martin.kalcok@gmail.com,
matus@krizan.se, jan@skalny.sk

Abstract. In this paper, we describe our approach to integrate SDN (Software Defined Networking) into GPRS mobile networks. SDN separates data and control planes that allows simpler, cheaper and vendor independent devices in network (forwarders), managed by a logically centralized controller. While SDN was implemented in mobile networks before, these solutions are focused on 3G and LTE networks due to the fact that implementation of SDN in these networks is relatively easy. We based our solution on *OpenFlow* 1.3 protocol which we extended with *experimenter actions* and custom matches. In the end of paper, we evaluate our modified network architecture and benefits of the use of *OpenFlow* protocol and SDN approach.

1 Introduction

SDN (Software Defined Networking) [5] is an innovative approach to philosophy of networks that stands on idea of separating control and data plane. SDN introduces a node called "controller" which enables centralized control of whole network. In a traditional network, control is distributed in a sense that every device must maintain its own view of the network and then make forwarding decisions based on this view. On the other hand, SDN network consists of forwarders that are connected to the controller and controlled by it. Having every forwarder connected to the controller allows this one to have an objective view of the whole network. Controller sets up every forwarder with appropriate forwarding table, removing responsibility of network nodes to have any kind of information about network topology. With control of network centralized in the controller, there is no need for any control plane in devices, therefore routers are replaced with forwarders. Thanks to controller being a single node, maintaining a network becomes a simpler task. These advantages, combined with fact, that all protocols and interfaces between SDN nodes are open, allow building cheaper, vendor independent networks that are easier to maintain and manage.

[*] Master study programme in field: Computer Engineering
Supervisor: Assoc. Professor Ivan Kotuliak, Martin Nagy, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 553–558.

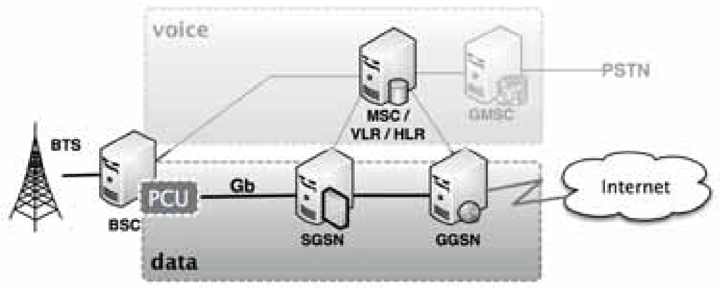


Figure 1. GPRS architecture.

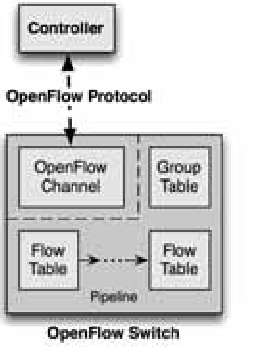


Figure 2. OpenFlow forwarder architecture [6].

In this paper we discuss the application of SDN in a General Packet Radio Service (GPRS) core network. In the sphere of mobile networking, SDN was already implemented into 3G and LTE networks [3], however there were no practical implementations for GPRS, even though it is still widely used. These networks already have separated signaling and user data plane and are more IP oriented so application of SDN on their architecture is much simpler [1]. In the next Section, we briefly describe the architecture of GPRS networks and the OpenFlow protocol. Our approach how to design and implement a GPRS core network based on SDN is proposed in the third Section. Later we describe the implementation process based on our proposal and show the necessary SDN nodes' modifications. To sum up, last Section is dedicated to testing of our prototype and future proposals, since we believe that while somewhat outdated, GPRS maintains its relevance in systems with moderate bandwidth demand, thanks to its lower power consumption and lower costs.

2 Analysis

2.1 GPRS architecture

GPRS has been developed to provide packet services over mobile network. While GPRS shares some of its nodes (Base Transceiver Station (BTS) and Base Station Controller (BSC) nodes for example) with the GSM network, it has also introduced some new nodes, namely Serving GPRS Support Node (SGSN), Gateway GPRS Support Node (GGSN) and Packet Control Unit (PCU). These nodes are responsible for the transport of packets between mobile stations (MS) and other networks. Architecture of typical GPRS network can be seen in Figure 1.

PCU connects to the SGSN via Gb interface which combines both data and signaling protocols [7]. SGSN is the most important node of a GPRS network because it has several important roles: mobility management, routing to correct GGSN, cooperation with databases of users, data encryption and compression and cooperation with other GSM network components [7]. Another important node is GGSN, which serves as a gateway to other external networks and it has these roles: session management, IP address assignment, routing to correct SGSN, etc. GGSN and SGSN are connected via Gn interface [7]. Other GPRS/GSM nodes include Mobile Switching Center (MSC), Home Location Register (HLR), Visitor Location Register (VLR), Equipment Identity Register (EIR), Litigation Gateway (LIG) and their job is support of roles regarding user and device authentication, mobility management, charging, etc.

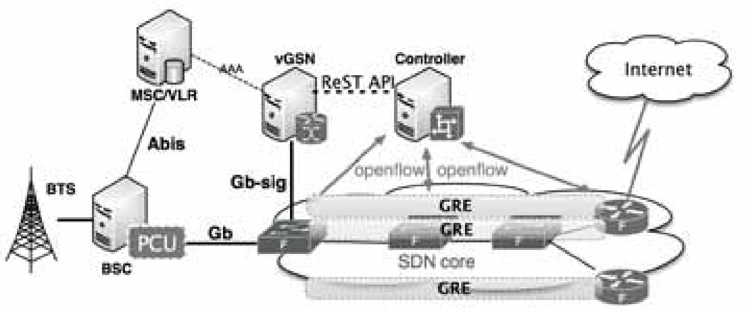


Figure 3. Design of our modified GPRS architecture.

2.2 OpenFlow

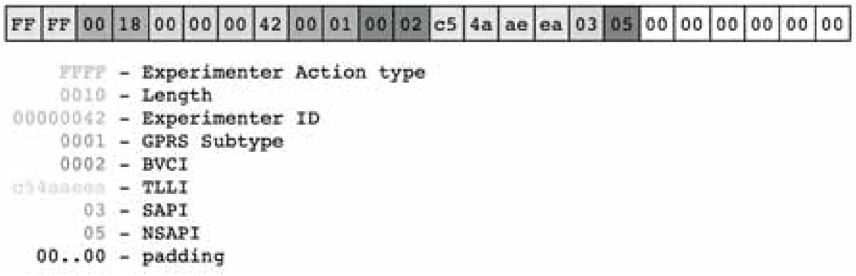
OpenFlow is a southbound protocol in SDN networks developed by the Open Networking Foundation (ONF). It has been proposed as an interface for communication between a network node and its controller. To identify network traffic, OpenFlow uses concept of flows that use static or dynamic traffic matching rules. It allows programming the network "per flow", thus providing granularity of network control and allowing the network to react to changes on application, user or session level. It allows the applications built on top of the network to adjust the network according to their needs.

Every OpenFlow forwarder using protocol version above 1.0 consists of one or more flow tables, a group table used for routing and lookups and a secure channel used for communication with the controller (Figure 2.) [6]. Each flow table contains several flow records consisting of match rules, counters and actions. Matching with the match fields starts at the first flow table and might continue through others, depending on the action taken. Data in packet headers are compared with the matching rules and if a match occurs, forwarder performs actions defined by their respective matching rule. Forwarder can then send the packet to its ports (physical or virtual), change the packet's headers, discard the packet or forward it to the controller.

3 Our Proposal

The ultimate goal of this project is to separate GPRS signaling and user data based on the SDN approach. Figure 3. shows our new GPRS architecture. As depicted in the Figure 3., there are some new nodes previously unseen in a GPRS network (vGSN, Controller) and some GPRS nodes (SGSN, GGSN) are missing, too. The vGSN (virtual GPRS Support Node) node serves as a replacement of both SGSN and GGSN nodes and handles the GPRS signaling. We decided to put the functionality of both SGNS and GGSN into one node which allows us to remove the Gn interface connecting them and allows us not to use the GTP (GPRS Tunneling Protocol) protocol for communication between the SGSN and GGSN. The function of ingress forwarder connected to PCU is to split the user data (LL3, LL5, LL9, LL11) and signaling (LLGMM) in accordance with SAPI (Service Access Point Identifier) address and based on user defined flow tables. The function of vGSN is to maintain GPRS signaling, such as GPRS mobility management (GMM) and session management (SM). SDN controller orchestrates flow control in the network with each Packet Data Protocol (PDP) context activation. Signalization replies from vGSN to PCU, used for NS (Network Service) and LLC (Logival Link Control) management, are being forwarded through the border forwarder (forwarder closest to the PCU), which acts as a switch. We are not using any GPRS Tunneling Protocol (GTP) tunnels in our design, instead we are mapping each active PDP context based on Access Point Name (APN) and Quality of Service (QoS) to pre-defined Generic Router Encapsulation (GRE) tunnels.

By choosing to use OpenFlow, we committed to extend its functionality to support NS, LLC



```
FF FF 00 18 00 00 00 42 00 01 00 02 c5 4a ae ea 03 05 00 00 00 00 00 00

    FFFF - Experimenter Action type
    0010 - Length
00000042 - Experimenter ID
    0001 - GPRS Subtype
    0002 - BVCI
c54aaeea - TLLI
      03 - SAPI
      05 - NSAPI
  00..00 - padding
```

Figure 4. Structure of `PushGPRSNS` *action.*

and SNDCP (Subnetwork Dependent Convergence Protocol) protocols and packets. OpenFlow version 1.2 [4] and higher supports Extension-42, which is meant as a standard for vendor-specific actions, and *experimenter match* rules. We are using both of these functions to implement needed functionality in following way. *Match rules* for identification of different GPRS stack protocol addresses and *experimenter actions* for manipulation with these headers. Communication between vGSN and controller is allowed thanks to custom REST (Representational State Transfer) API, which allows controller to receive data regarding PDP contexts and device mobility.

4 Implementation

4.1 Controller side

As controller, we used *Ryu* [8], which is framework for creating SDN applications. Our work consisted of creating forwarding application and creating *experimenter actions* and custom matches to deal with all kinds of GPRS signaling.

4.1.1 Experimenter action

Ryu already contains support to create *Experimenter Actions*, it is called `OFPActionExperimenter`. Inheriting from this class we were able to create OpenFlow actions in format that is compatible with OpenFlow 1.3.2 specification. We created class `GPRSAction` that inherits from `OFPActionExperimenter` and serialize our GPRS related actions. For actions that do not need additional parameters, generally *POP* actions (actions used to remove GPRS headers from packets originating at MS), it is enough to call `GPRSAction` with appropriate subtype value. Actions like *PUSH* (actions for adding GPRS headers to packet terminating at MS), on the other hand, had to implement its own serialize function responsible for serialization of all the additional parameters into message payload. Structure of `PushGPRSNS` action can be seen in Figure 4.

4.1.2 Custom Matches

To add support for our own match rules, we modified `OpenFlowBasic` class which is responsible for creating standard match rules. According to OpenFlow 1.3.2 specification, every match belongs to match class (`OFPXMC`), for standard matches it is class `0x8000`. We used the fact that our matches have same structure as standard and we just extended controllers ability to create matches with variable `OFPXMC` value which was previously hard coded to `0x8000`. Afterward we extended list of existing matches with our own definitions. For matches related to GPRS we chose class value of `0x7FFF`. Last thing we had to adjust was the order in which match rules are applied. Matches must be applied in order with packet encapsulation (i.e. match for IP address cannot be done before match for MAC address), and so our GPRS related matches had to proceed standard matches.

4.2 Forwarder side

As a forwarder, we decided to use *ofsoftswitch* by *CPqD* [2], which is OpenFlow 1.3 compatible software forwarder. It took significant modification on forwarder side to add support for actions and matches related to GPRS. Our work on forwarder includes adding support for *Experimenter Actions*, creating own *Experimenter Actions* and custom matches to work with GPRS protocols.

4.2.1 Experimenter Actions

Original *ofsoftswitch* already contained few calls related to *Experimenter actions* but these calls were not handled in any way and there were no structures for work with these extensions. Support for *Experimenter actions* was the first thing we implemented on forwarder. For packets that originate at mobile device and go to Internet, we have *POP* actions that strip headers of GPRS-NS, IP and UDP protocols. (`dp_exp_action_pop_gprsns ()`, `dp_exp_action_pop_udp ()`, `dp_exp_action_pop_ip ()`). After stripping away headers, these functions reorder rest of the data in packet to fill hole after missing headers and reduce overall length of packet. Then we have mirror of *POP* actions, *PUSH* actions, these handle packets coming from Internet. They first create space at beginning of the packet and then they add headers for appropriate protocols.

4.2.2 Custom Matches

Using real *Experimenter matches* would require big changes into ways in which *ofsoftswitch* works so we decided to go with defining new match class. This allowed us to use all the existing functions and structures used to work with standard matches. Adding new type of match or match class requires:

1. defining new field and its prerequisites in `oflib/oxm-match.def`,
2. defining new class, type and length of matching field in `oflib/oxm-match.h`,
3. add parsing according to `oflib/oxm-match.h`,
4. add debugging functions into `ofl_structs_oxm_tlv_print` and `ofl_oxm_type_print`.

After adjusting number of defined TLVs - `NUM_OXM_FIELDS` in `oflib/oxm-match.h` file, forwarder was able to correctly process our custom match rules. However to allow *ofsoftswitch* correctly forward frames we had to program support for analysis of GPRS packets into nbee library.

To identify new TLVs we built analyzer of GPRS frames into `nblink_extract_proto_fields` function. In this stage of project we consider every UDP packet with destination port of `23000` to be GPRS frame. Every incoming GPRS frame is scanned for NS type and BVCI address. If the frame is of `NS_UNITDATA` type we analyze PDU type and TLLI address on BSSGP layer. Other TLV fields of BSSGP layer are not considered. If BSSGP layer indicates LLC sublayer, we analyze its content, SAPI address and format of LLC frame (I, S, UI, U). In case the LLC frame is of UI type and has SAPI values of LL3, LL5, LL9 or LL11, we decode also SNDCP sublayer, specifically NSAPI address as well as bits indicating fragmentation of frame.

5 Prototype and Testing

This chapter describes our progress in implementation and its testing. The main goal here was, to verify our implemented extensions of OpenFlow 1.3 protocol [6]. The forwarder has to separate GPRS signaling from user data on the forwarder and send them to proper interfaces. The test setup is in the picture below (Figure 5). During the testing we first connected the forwarder with the controller, where we created a static PDP context. After the connection, the controller created basic

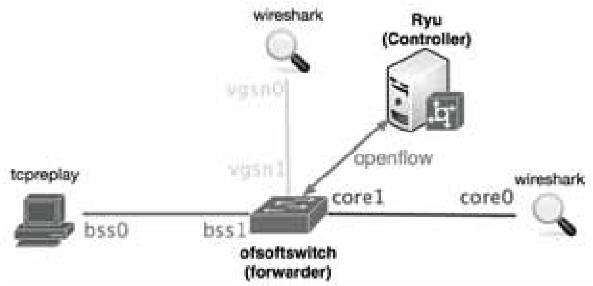


Figure 5. Testing of our prototype.

match rules on the forwarder. The controller then created match rules for all active PDP contexts, which were responsible for removing GPRS headers and sending user data through the core interface. To verify proper functioning, we connected packet analyzers both to core and vGSN interface. Using tcpreplay we injected previously captured data to the bss interface. Packet analyzer on the vGSN interface captured signaling messages but no user data. Packet analyzer on the core interface captured only user data. By proving this, we fulfill our goal and show that it is possible to build a GPRS network with separated data and control planes according to the philosophy of SDN.

6 Conclusion

In this paper, we show how a GPRS architecture might look like, if it is designed by applying the principles of SDN. We propose the design of new architecture, modifications to the *ofsoftswitch13* forwarder, *Ryu* controller and benefit from using *experimenter actions* together with our custom matches. To manage the network nodes, we use *Ryu* controller allowing us to do the management work from one place. Our experiment shows that it is possible to implement SDN ideas into GPRS network, what is proven by our working prototype. In the future, we plan to implement a working GPRS network with two-way communication according to the principles of SDN instead of the one-way communication simulation we have now.

Acknowledgement: This work was partially supported by Slovak National research grant 1/0676/12 and by Tatra banka foundation under the contract No. 2012et011.

References

[1] Network Functions Virtualization – Introductory White Paper, 2012.

[2] CPqD: OpenFlow 1.3 Software Switch. `https://github.com/CPqD/ofsoftswitch13`.

[3] Kempf, J., Johansson, B., Pettersson, S., Luning, H., Nilsson, T.: Moving the mobile Evolved Packet Core to the cloud. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, 2012, pp. 784–791.

[4] Open networking foundation: OpenFlow Switch Specification 1.2, 2011.

[5] Open networking foundation: Software-Defined Networking: “The New Norm for Networks.”, 2012.

[6] Open networking foundation: OpenFlow Switch Specification 1.3.2, 2013.

[7] Seurre, E., Savelli, P., Pietri, P.: *GPRS for Mobile Internet*. Artech House mobile communications series. Artech House, 2003.

[8] Telegraph, N., Corporation, T.: Ryu SDN Framework. `http://osrg.github.io/ryu/`.

Built-in Self-repair for a Processor Multiplier

Andrej KINCEL*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
andrej.kincel@gmail.com

Abstract. The paper deals with an implementation of a built-in self-repair architecture into LEON3 processor environment. New reconfigurable logic block architecture (self-repair wrapper) of the LEON3 multiplier is designed providing self-repair ability; moreover reliability parameters (R and the mean time to failure) of the multiplier are improved as well. The test algorithm necessarily required in the self-repair procedure is also proposed. Estimations of the area overhead required for BISR, multiplier reliability parameters and fault coverage of the test algorithm are presented at the end of the paper.

1 Introduction

Device reliability and the mean lifetime are important issues in semiconductor industry. Built-in self-repair (BISR) is a possible solution for both requirements. A logic core can acquire BISR capabilities if transformed into several reconfigurable logic blocks (RLBs). RLB ensures the repair function of the core or the assigned part of the core. The part consists of several identical function blocks (FBs) for which an additional backup block (BB) with the same function is attached. The basic reconfigurable architecture RLB^{3+1} uses three FB and one BB. Such as RLB has four logic states; the first one leaves BB unused, while each of the other states replaces one faulty FB with the BB [3].

The paper is organized as follows. Section 2 describes the basic configuration of the LEON3 multiplier. Section 3 presents the new multiplier design upon a set of requirements. Last Section 4 summarizes the results and concludes the paper.

2 LEON3 hardware multiplier

Targeting the reliability of a processor and getting possibility to obtain a description at the suitable hierarchy level (logic gates) one specific hardware multiplier structure is examined (based on source code) for BISR implementation utilizing the RLB architecture, i.e. segmentation into several RLBs.

16x16-bit multiplier considered in the paper is producing 64-bit product with latency of 4 clocks. The multiplier is implemented in LEON3 processor, but can be used also in various RISC processors (e.g. MINIMIPS). This multiplier core is comprised of three blocks (Figure 1):

[*] Master study programme in field: Computer Engineering
Supervisor: Dr. Marcel Balaz, Institute of Informatics, Slovak Academy of Sciences

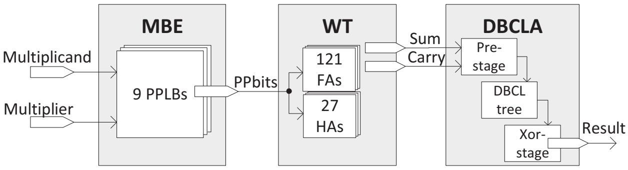


Figure 1. LEON3 multiplier block diagram.

Modified booth encoder (MBE) is used for the multiplier recoding and formation of the reduced number of partial products. The regular structure of MBE results from the partial products creation, since each partial product is formed by one partial product logic block (PPLB). As the multiplication with help of MBE requires $\lceil (M+1)/2 \rceil$ partial products (where M is the count of the multiplier bits), 16x16-bit multiplication needs 9 partial products, therefore 9 PPLBs should be utilized.

Wallace tree (WT) is used for the summation of the partial products that results in two vectors. This adder's tree consists of $\lceil log_{3/2}(M/2) \rceil$ levels of full adders (FAs) and half adders (HAs) interconnected in the tree structure to reduce M inputs down to two carry-save redundant form outputs.

Distributed binary carry-lookahead adder (DBCLA) adds the two vectors to obtain the final multiplication product. DBCLA is composed of three blocks: Prestage, DBCLtree and Xorstage. These blocks are highly data dependent inside, thus any further analysis of their inner structure is not necessary for this work.

3 Design of the multiplier with a self-repair wrapper

The new designed multiplier includes a self-repair wrapper while the functionality of the multiplier is preserved. A redesign of LEON3 multiplier to a multiplier with self-repair capabilities by applying the RLB method has necessarily to satisfy the defined requirements [1].

3.1 Area requirements

The self-repair characteristic of the RLB architecture is built upon hardware redundancy. The required added area impacts (lowers) the overall reliability of the reconfigurable multiplier, because every additional transistor or interconnection creates a new possible defect spot followed by a failure. Therefore, the area overhead must be minimized as much as possible. Four main sources of area overhead were identified:

BISR controller proposed and implemented with maximum effort to minimize its area [1] is simply reused in the multiplier design, so any other area reduction is not possible.

RLB switches - a path selector (demultiplexing) is necessary to choose the path from a primary input to FB or BB. On the output side, another path selector (multiplexing) is deciding, whether the outputs from FB or BB will be propagated to a primary output. Four transistors are needed to create such path selector (multiplexing or demultiplexing) for one bit switching. The size of the path selector cannot be reduced any more but it is still possible to minimize the number of used path selectors. Since this number depends on the quantity of the input/output bits of a block, the blocks have to be chosen wisely in terms of inputs and outputs which must be switched.

RLB architecture - area overhead introduced by each BB depends on the number of FBs, which can be replaced by this BB. In other words, implementing RLB^{3+1} architecture means that one BB is added to three FBs, therefore BB will cause overhead equal to $1/3$ of an original system area. Other case is RLB^{1+1} architecture, where the BB introduces overhead equal to the original system area. The RLB^{3+1} architecture brings less overhead so it should be preferred. However, also the fact should be considered, that if strong data dependency exists inside the system smaller switching circuitry (fewer path selectors) is required in RLB_{1+1}. Breaking N-bit wide data path requires $3N$ switches if creating RLB^{3+1}, while RLB^{1+1} needs only N switches. Then it is more advantageous to use RLB^{1+1} which is a form of duplication, where only the primary inputs/outputs of the system are switched [1].

Test unit - as the target multiplier is the part of LEON3 processor, the processor's resources (computing power, registers, memory, etc.) are available to be used. Therefore, software-based self-test (SBST) [4] should be implemented to avoid additional hardware overhead.

3.2 Functional requirements

The functional requirements of the redesign are very straightforward and simple, because the most functionality is covered by the BISR controller. They can be listed as follows:

The correct multiplier function after the redesign to multiplier with the RLB structure must be preserved and verified.

Input/output switches belonging to one RLB (either RLB^{3+1} or RLB^{1+1}) should be controllable by one control signal generated by the BISR controller.

The switching circuitry has to be able to isolate any faulty FB and connect respective BB instead.

3.3 Test requirements

The multiplier should be tested as a black box and the SBST test algorithm is executed several times until the BISR controller discovers the correct configuration of the multiplier segmented into RLBs, which has passed the test. Hence the requirements for the test algorithm can be stated:

Because the test is executed more than once, it influences the reconfiguration time. Therefore, the test should consist of the limited number of instruction cycles, to keep the execution time at a reasonable level.

The fault coverage of the test should be sufficient for on-line testing purposes, i.e. the case of 100% covered faults is an ideal but not a must for such test. A reasonable trade-off between the fault coverage and the execution time should be find out.

When execution of the test is required, the BISR controller rises the value of signal *test_enable* to the high (active) level, and then is expected a *go/nogo* result concurrently with high value of the *result_valid* signal from the test unit. The implemented test algorithm has to exactly fit this BISR controller interface [1].

Optionally, the number of test vectors should be the same for different multiplier sizes; e.g. for either 32x32-bit multiplier or 16x16-bit multiplier.

3.4 Design of the multiplier with a self-repair wrapper

The multiplier BISR architecture shown in Figure 2 consists of properly interconnected blocks:

LEON3 instruction unit executes a test algorithm. The trigger for the algorithm start is an interrupt request (IRQ) generated by fault detection/localization procedure signal *test_enable*. The test result is either *go* or *nogo* signal concurrently with active value of *result_valid* signal.

Fault detection/localization procedure (FDLP) is the main component of the BISR controller. With this block, the multiplier gains self-repair capabilities because in the case of the *nogo* test result it searches for the fault-free working configuration of the multiplier with the RLB structure [1].

One **state blocking circuitry (B)** is required for each RLB, thus 7 Bs should be implemented in the presented case. The purpose of B circuitry is to preserve the fault-free state in RLB after recovery from a faulty configuration. This demand is set by active *save* signal from FDLP, while the need of RLB configuration change is signalized with appropriate bit of vector signal v [1].

Multiplier with the RLB structure contains the original multiplier of LEON3, which is divided into 7 RLBs. The redundant BBs added into each RLB enable the multiplier to recover from up to 7 single faults (in condition that each fault will occur in different RLB).

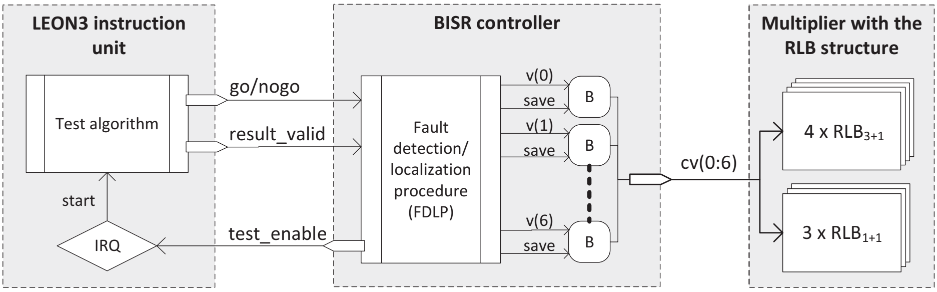


Figure 2. Block diagram of the multiplier with a self-repair wrapper.

3.5 Multiplier with the RLB structure

According to the above mentioned requirements, the multiplier was partitioned into 4 RLB^{3+1} and 3 RLB^{1+1}. The proposed partitioning is shown in Figure 3. The white blocks inside each RLB represent FBs, while the black blocks represent BBs. SWS stands for the switching circuitry built from the path selectors as analysed before. The block **state controller (SC)** is connected to individual RLB architectures. SC holds the state which determines the RLB configuration, which are changed in cyclic order (state 0, 1, 2, 3, 0...). The change of state is triggered by the active value of one bit signal *cv*. The SC is inseparable part of the RLB architecture and for every RLB one SC is required [1].

The **MBE** for 16-bit multiplier consists of 9 PPLBs and this regularity can be advantageously used by creating three RLB$_{3+1}$ architectures, each comprising three PPLBs plus one redundant backup PPLB.

The **WT** used in 16-bit multiplier is comprised of 121 FAs and 27 HAs interconnected in the tree structure. The WT can be divided into three FB consisting of the same number of FAs and HAs but all input bits have to be switched. The number of FAs is not divisible with three, what will cause that the first FB will contain 41 FAs and the other two FBs only 40 FAs. This will break the rule of the equivalent FB structure, hence one redundant FA is added to the second and the third FB.

The **DBCLA** consists of three smaller blocks (in terms of area) but these blocks have a large amount of input or output bits. Unfortunately, deeper analysis of these blocks showed up that they are even more complex inside (in terms of data paths bit width). Trying to build RLB^{3+1} architecture will lead to an enormous increase of area overhead because of the required switching circuitry. The circumstances for using RLB^{1+1} architecture stated in the area requirements are met, therefore the most reasonable way is to duplicate each of the three blocks and switch only their outside input/output ports.

3.6 Test algorithm

The test algorithm published in [2] is realized in form of hardware built-in self-test (BIST). In the presented design the algorithm was implemented as SBST with no need of additional area overhead.

The algorithm is intended to test a multiplier by applying test patterns to its input operands (multiplicand and multiplier) and then observing the results on the multiplier output. The repetitive test patterns are used, where **number** k is termed **the repetition length** of the bit pattern. Bits with a distance of k positions always receive the same binary value. For example, when targeting 16-bit multiplier and using $k = 4$, both operands receive the 16 different values, thus producing $16 \times 16 = 256$ test patterns together [2]:

$$\begin{aligned}&\mathtt{0000\ 0000\ 0000\ 0000}\quad, \mathtt{0001\ 0001\ 0001\ 0001}, \ldots\\ &\mathtt{1110\ 1110\ 1110\ 1110}\quad, \mathtt{1111\ 1111\ 1111\ 1111}.\end{aligned}$$

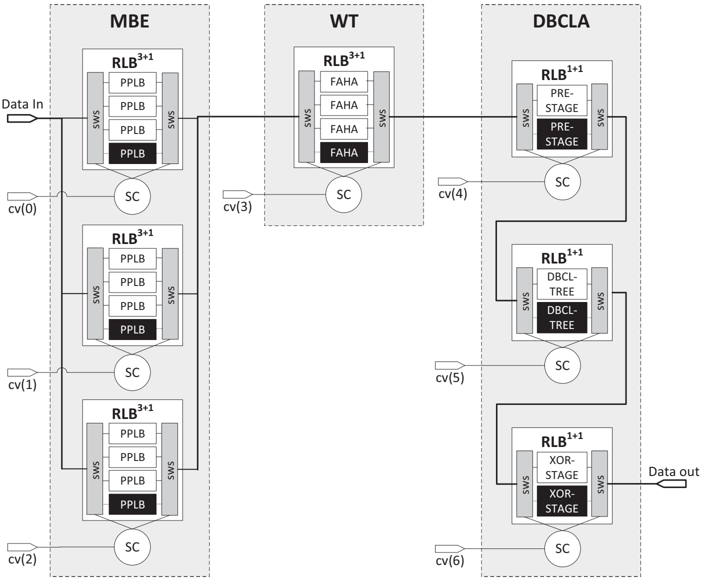


Figure 3. Block diagram of the LEON3 multiplier segmentation into 7 RLBs.

The number of patterns which have to be produced is equal to 2^k for each of two multiplier operands, regardless of the size of the operands. Therefore, the number of test patterns is independent on multiplier bit-size. Moreover, in the same time unequal number k can be used for multiplicand and multiplier (e.g. 3 and 5).

In terms of SBST implementation of this algorithm, the test patterns are pre-stored in the RAM, which is available in LEON3 environment. While using the same k for both operands, the test vectors for multiplier and multiplicand are the same, thus only the test vectors for one operand have to be pre-stored. For example, choosing $k = 4$ and producing patterns for 16-bit (2B) operands, only $2 \times 2^4 = 32$B of memory should be allocated and filled. The test patterns are then loaded independently for both multiplication operands by use of two different memory address pointers. After loading, patterns are applied to multiplication operands, using multiplication instructions UMUL (unsigned) and SMUL (signed) from the SPARC instruction set.

The correct test results can be also pre-stored in the RAM, but this could obviously consume a considerable amount of memory. For example, using $k = 4$ and expecting 64-bit (8B) multiplication result, $8 \times 2^4 \times 2^4 = 2048$B of memory should be preloaded. The comparison of a multiplication result with the expected one, is done after every multiplication and the correctness of the multiplication results evaluated.

4 Results and conclusion

As the estimation in Table 1 shows, the area overhead is kept on the reasonable level of 88.97%, especially in comparison with other hardware redundancy architectures (e.g. TMR). The column *FB transistors* corresponds to the original multiplier architecture without any added circuitry, while the columns *BB, SWS and CTRL* represent the area added for self-repair.

The test algorithm was implemented with different numbers k, with multiplier $k = 3$ and multiplicand $k = 5$. The proposed test fault coverage of 98.27% faults was proven by fault simulation targeting single stuck-at faults. Also the test speed was evaluated and is expressed as 13 instruction cycles needed to apply one pattern, thus one test run takes $13 \times 256 = 3328$ instruction cycles. As the fault detection provided by FDLP needs, on average, 5.57 test runs for BISR built

Table 1. The area estimation of multiplier architecture with a self-repair wrapper.

Part	FB tr.	BB tr.	SWS tr.	CTRL tr.	Overhead %
MBE	9684	3228	768	594	47.40%
WT	5178	1726	2836	198	91.93%
DBCLA	6128	6128	1056	570	126.53%
FDLP				1570	
Total	**20990**	**11082**	**4660**	**2932**	**88.97%**

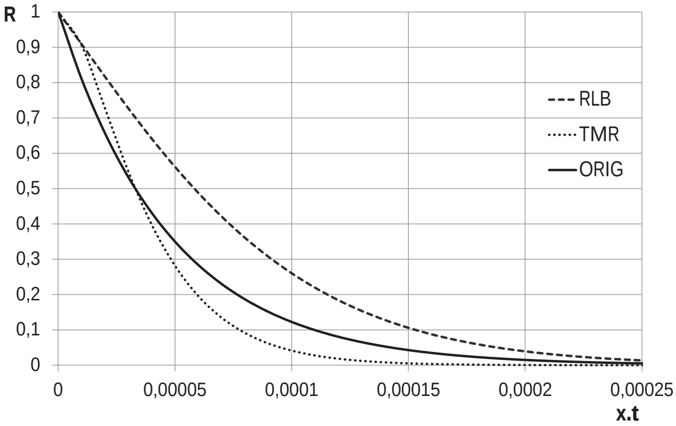


Figure 4. R(t) = probability of still working multiplier at time t.

of 7 RLBs [1], the average fault detection time stabilizes at 18537 instruction cycles. The test algorithm was implemented only for 16x16-bit multiplier targeted in the paper but the test vectors count independence of an operand (multiplicand/multiplier) size could be used in future, for a bigger multiplier testing.

According to the well-known reliability model [5], the reliability (R) of the new proposed multiplier architecture was estimated. In Figure 4 the R functions of three multiplier designs are compared: with (RLB) and without (ORIG) self-repair wrapper, and using TMR method. It should be noticed that the reliability of RLB takes into account all necessary added area, while TMR reliability is calculated without majority voter overhead.

Acknowledgement: This work was supported by Slovak national project VEGA 2/0034/12.

References

[1] Balaz, M., Kristofik, S., Fischerova, M.: Generic Built-in Self-Repair Architectures for SoC Logic Cores. In: *Design and Diag. of Electronic Circuits Sys. (DDECS)*, 2014.

[2] Gizopoulos, D., Paschalis, A., Zorian, Y.: An effective built-in self-test scheme for parallel multipliers. *Computers, IEEE Transactions on*, 1999, vol. 48, no. 9, pp. 936–950.

[3] Koal, T., Scheit, D., Vierhaus, H.T.: A concept for logic self repair. In: *12th Euromicro Conf. Digital System Design: Architectures, Methods and Tools, DSD 2009*, 2009, pp. 621–624.

[4] Psarakis, M., Gizopoulos, D., Sanchez, E., Reorda, M.: Microprocessor Software-Based Self-Testing. *Design Test of Computers, IEEE*, 2010, vol. 27, no. 3, pp. 4–19.

[5] Shooman, M.L.: *Reliability of Computer Systems and Networks: Fault Tolerance,Analysis,and Design*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

Therapeutic System for Children with Movement Disorders

Kamil BURDA, Rudolf GREŽO, Marek HASIN, Lukáš KOHÚTKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
imagine-cup-2013-pss@googlegroups.com

Abstract. In this paper we propose a system that improves the physical therapy of children with movement disorders such as cerebral palsy or apraxia. The system encourages children to exercise by playing computer games using Kinect and Leap Motion sensors during therapy sessions. By playing together cooperatively, children can find new friends and improve their social skills. The system also allows physiotherapists to track children's progress and adjust the game difficulty to adapt to their individual needs. This paper is focused on the features to be used by the children and the physiotherapists.

1 Introduction

The life of patients with limited motor abilities, caused by movement disorders such as cerebral palsy or apraxia, is in many ways different from healthy people. These patients perform everyday activities with great difficulties and are often dependent on the help from other people.

Patients with cerebral palsy (CP) are unable to properly coordinate their movements or, in more severe cases, move one or more body parts at all [1]. CP is caused by improperly developed brain before, during or after birth, which may also affect their intelligence. Patients with apraxia have difficulties performing learned, skilled movements [2], such as catching or throwing a ball. Being able to move all of their body parts [3], these patients have greater potential to improve their motor skills than patients with CP. Children with apraxia are often rejected by their schoolmates [4].

In traditional physical therapy sessions conducted by physiotherapists, patients, especially children, are often not motivated enough to perform exercises willingly. Computer games controlled by motion sensors (such as Kinect) prove to be a feasible option for increasing the therapy efficiency [5, 6, 7, 8]. Engaging multiple patients with similar movement disorders in therapy sessions allows them to socially interact with each other, improving their social skills [9]. Patients also benefit more if the therapy starts at an early age[1] [10].

[*] Master degree study programme in field: Computer Engineering
Supervisor: Martin Nagy, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

[1] http://www.medicalnewstoday.com/articles/151951.php

Before the school age, the main social contact of children is focused on their parents. At the moment the child begins to go to school, it begins to feel the need to socially contact with its peers. Social isolation and rejection in the early childhood may negatively impact their social integration and establishing relationships later in their lives.

The proposed system aims to combine the concepts of therapy gamification and socialization of patients by introducing a set of serious games controlled by motion sensors, where patients cooperate with each other while playing. Having fewer opportunities to socially interact with their peers and find new friends, the proposed system improves the social aspect of their lives in addition to making the physical therapy a means of entertainment. The proposed system is to be used primarily by children with CP or apraxia in preschool and school age (from 5,5 years onward) and by physiotherapists.

This paper is structured as follows. Section Related work discusses existing therapeutic systems with similar purposes. Section Architecture introduces the architecture of the proposed system and its components. Section Game describes the first game implemented in the proposed system. Section Web platform describes the web platform to be used by both the therapists and the children.

2 Related work

Computer games controlled by motion sensors such as Kinect, Wii Remote or EyeToy are often too difficult for patients to win or achieve sufficient score, discouraging them from playing the game and exercising [5]. In the recent years, therapeutic or virtual rehabilitation systems using motion sensors have been developed. These systems implement games which can be configured to adapt to patients' varying disabilities. We have examined Jintronix Rehabilitation System [11], SeeMe Rehabilitation[2] and a serious game called Voracy Fish[3].

In general, these systems use the Kinect sensor and consist of two modules - a set of serious games for patients and an application for therapists. The games can track patients' movements and provide feedback to the patients. Using the application, therapists can configure the game difficulty and track patients' progress by reviewing data collected in the games.

Jintronix Rehabilitation System allows patients to perform exercises at home, assigned individually by therapists via the application. Voracy Fish allows multiple patients to play against each other. Given the mild violence (the player character, a fish, devouring various objects or smaller fish), we consider this game inappropriate for young children.

The systems lack the possibility of cooperative playing, which, in our proposed system, can improve the children's social skills to better integrate into the society.

3 Architecture

This section describes the architecture of the proposed system. The system implements serious games controlled by two motion sensors, Kinect for Windows and Leap Motion, both of which are suitable for improving different sets of motor skills. The system supports multiple players, fostering the cooperation of patients. The system also allows therapists to control the game difficulty and track patients' progress. Based on these requirements, we designed the architecture of the system shown in the Figure 1.

[2] http://www.virtual-reality-rehabilitation.com/products/seeme/what-is-seeme

[3] http://www.voracyfish.fr/?lang=en

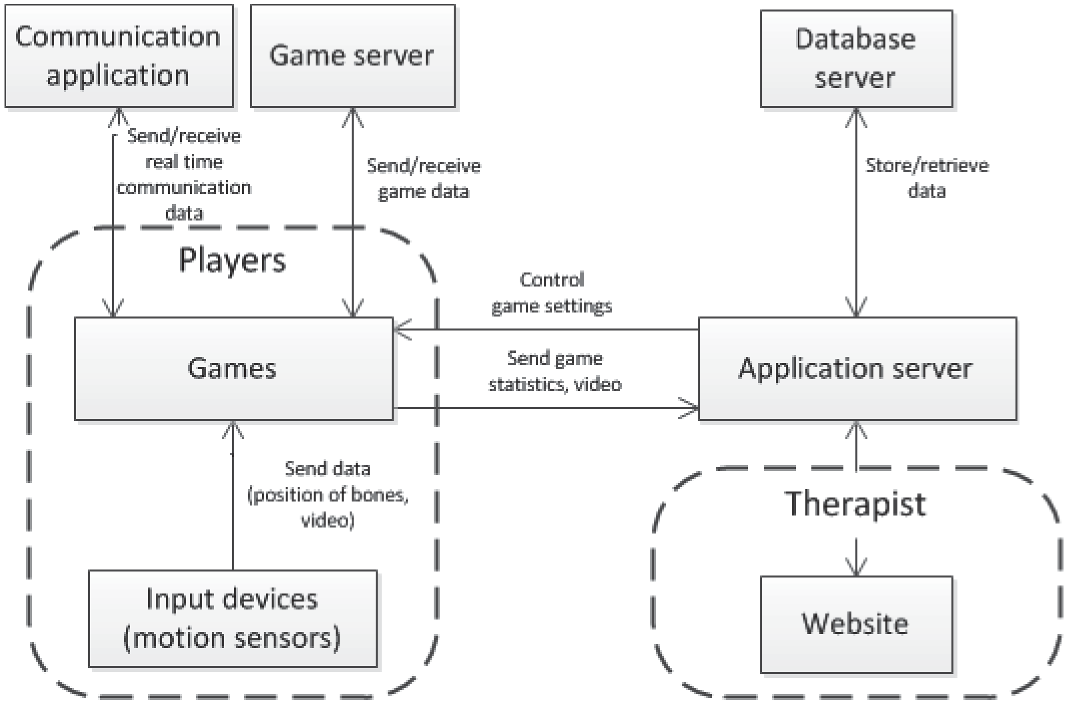

Communication application
Game server
Database server
Send/receive real time communication data
Send/receive game data
Store/retrieve data
Players
Control game settings
Games
Send game statistics, video
Application server
Send data (position of bones, video)
Therapist
Input devices (motion sensors)
Website

Figure 1. Architecture of the proposed system.

Kinect and Leap Motion sensors are commercially available, supported on Windows (which is the target platform) and allow easy development of applications via their software development kits. We determined that Kinect is suitable to improve patients' gross motor skills (such as throwing or catching a ball, running or jumping), which involve moving larger body parts and faster, less precise movements [12]. Leap Motion tracks the movement of hands, individual fingers, hand closing or rotation, suitable to improve patients' fine motor skills (such as grabbing objects, cutting with scissors, drawing), involving slower and more precise movements [13].

The serious games are developed using the Unity[4] game engine, allowing easy development of 3D game environment, levels, game mechanisms, quality graphics and integration of both motion sensors. The games can be played in three modes: single-player, local multiplayer and online multiplayer. In the single-player mode, one player uses one of the sensors and the rest of the game is controlled by the game automatically. Local multiplayer mode allows two players with both motion sensors to play together on one computer cooperatively.

Games in the online multiplayer mode are played over the Internet, allowing patients to play the games from home. The game server allows the players to log in and choose teammates and transfers real-time game data (such as the current position of the player character) among all players. The communication application allows players to communicate using instant messaging (IM), voice or pictographs. The database server is used to store data for each patient such as account information or game statistics (played games, achieved score, etc.).

Using the website, therapists can adjust the game difficulty to adapt to the motor abilities of individual patients. The application server collects data sent by the games while patients are playing them. Therapists can then display and review the collected data, allowing them to track the progress of patients in terms of their physical condition.

[4] http://unity3d.com/

4 Game

The first game currently in development, called My Fly (shown in the Figure 2), is described in this section. The player controls an aircraft flying among rock mountains and valleys and shoots stars that increase the player's score. The goal is to collect as many stars as possible in the shortest time before crossing the finish line.

The player using Leap Motion controls the position and rotation of the aircraft by moving one of his or her hands above the sensor. The player using Kinect controls the aiming and shooting from the aircraft by moving his or her left hand (by default). The aircraft moves automatically forwards to allow patients to concentrate on controlling the crosshair position and the aircraft position and rotation.

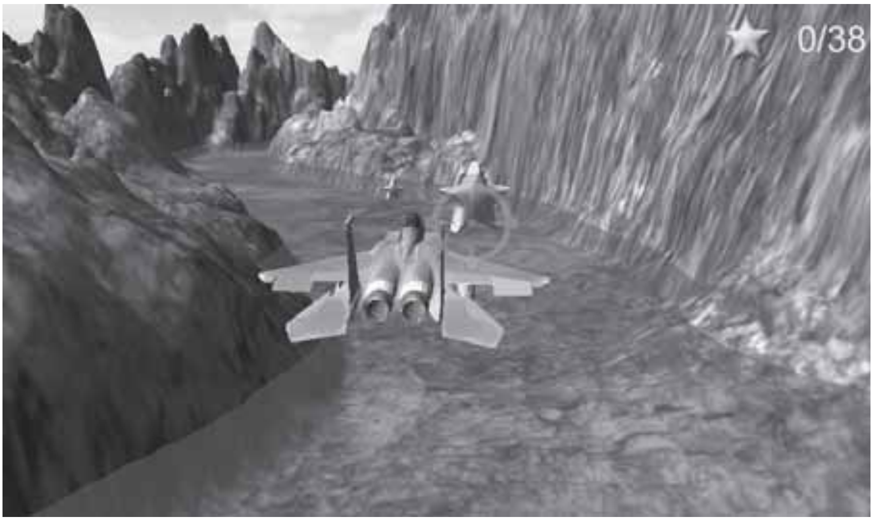


Figure 2. Screenshot from the game My Fly in its current state.

We consulted the game concept with therapists at the Research Institute of Child Psychology and Pathopsychology, Children's Center (Výskumný ústav detskej psychológie a patopsychológie, Detské centrum; hereinafter "VÚDPaP") in Bratislava. According to the consultation, patients often have wrong estimation of their movements - e.g. if they are instructed to move one of their hands from point A to point B, their hands are positioned outside the proximity of the point B. To make their movements more precise, patients perform movements that train their spatial orientation.

Because flying in the game allows moving in all possible directions in space, it challenges the patients in terms of their spatial orientation - in order to achieve sufficient score, they need to properly control the aircraft to avoid terrain collisions and to let the other patient aim at the stars with the crosshair to shoot them. Moving the aircraft and the crosshair with the motion sensors provide immediate visual feedback to the patients, allowing them to better estimate their movements and train their motor skills.

Important configuration parameters in this game include flight speed, duration of aiming at star before shooting, sensitivity of moving the aircraft and the crosshair and selecting the body part to control the crosshair. Configurability is important to prevent patients from failing too often, while retaining an appropriate difficulty, and to adapt to patients' disabilities.

5 Web platform

The web platform allows therapists to track the patients' progress in terms of their motor abilities and adjust the difficulty of games. The overall success of patients in games corresponds to the correctness of movements, which is determined by score gained from each level. The score is computed from individual statistics collected from games - e.g. number of collected stars or time to finish a level in case of My Fly.

To quickly review the patients' progress, the therapists can view score in the few last game sessions in the form of a graph. For detailed review, therapists can review individual game statistics or records of the patients playing the game, displaying the game and the visualization of the motion sensor movements next to each other.

The web platform also serves as a social network for children. The form of communication is configurable, so that children with limited motor, writing or reading skills are be able to communicate differently - verbally (using the Skype technology) or via pictographs. Pictographic communication is suitable for speech-impaired children. The children will be able to choose from a set of pictures, representing specific objects or words, and create sentences from them. These sentences are translated for other children into a different form (text, voice or pictures) according to their personal settings.

Third-party developers can join this platform and add their games to the list of serious games. At least one motion sensor must be used and the game needs to use our application programming interface in order to send the data from the game to the application server.

6 Evaluation

The proposed system will be regularly tested in the VÚDPaP institution to gather feedback from both the children and the therapists and to determine priorities during the system development. For each game, the following will be tested: ease of game control, additional important configuration settings we have not considered and the overall satisfaction of the children with the game. The improvements of patients' motor skills will be evaluated by the therapists for a longer period in multiple therapy sessions. In terms of the web platform, the therapists will help us determine what statistics are useful to collect from the games and how to adjust the weight of each statistic that determine the patients' score to better assess the success of patients in games.

7 Conclusion and future work

In this paper, we described the proposed therapeutic system, whose integral parts are a set of cooperative serious games controlled by Kinect and Leap Motion sensors and a web platform. Our goal is to make the physical therapy for children with movement disorders more entertaining, resulting in better performance while exercising and improved social skills of the children. The rules of the games are designed to be simple so that children can easily understand the gameplay and the goal of the games.

The system offers a web platform that serves as a social network for the children, reports results of the therapy sessions to the physiotherapists and can be extended by games from third-party developers. Game statistics are sent from the games to the web platform in order to show the therapists the results of the therapy. Online communication tools like IM are primarily to be used by children without mental disorders. On the other hand, the pictographic communication can be used by children with these disorders that impair their speech. The web platform is considered our future work, as it is currently in the conceptual stage.

Our future work also includes improving the existing game, My Fly, by introducing new levels and new mechanics such as health points, negative objects (decreasing the player's health and score or giving a penalty). These additions give the patients more challenge to further improve their motor skills and to keep them entertained.

Another game currently considered is about a crane driver controlling a crane using Kinect and another player carrying objects by grabbing them using Leap Motion. Another game is a football game, where the goalkeeper's hands are controlled by Kinect to catch the ball. Other players using Leap Motion control the ball with two adjacent fingers. These games will allow exercising different types of movements.

Acknowledgements: This project was partially supported by Tatra banka foundation under the contract No. 2013et029. We also thank Lenka Lesayová and Jana Bieliková for guidance regarding the psychological aspect of the game and the staff of the Research Institute of Child Psychology and Pathopsychology, Children's Center (Výskumný ústav detskej psychológie a patopsychológie, Detské centrum) for guidance regarding the physical aspects of the therapy.

References

[1] Rosenbaum, P. et al.: A report: the definition and classification of cerebral palsy April 2006. Developmental medicine and child neurology. Supplement, (2007), vol. 109, pp. 8-14.

[2] Goldman R. G., Grossman, M.: Update on Apraxia. Current neurology and neuroscience reports, (2008), vol. 8, no. 6, pp. 490-496.

[3] Gibbs, J. et al.: Dyspraxia or developmental coordination disorder? Unravelling the enigma. Archives of Disease in Childhood, (2007), vol. 92, no. 6, p-. 534-539. DOI 10.1136/adc.2005.088054.

[4] Boon, M.: Helping Children with Dyspraxia. Jessica Kingsley Publishers, (2001). ISBN 9781853028816.

[5] Geurts, L. et al.: Digital Games for Physical Therapy: Fulfilling the Need for Calibration and Adaptation. In: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction, (2011), pp. 117-124. ISBN 978-1-4503-0478-8.

[6] Chang, Y.-J. et al.: A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in Developmental Disabilities, (2011), vol. 32, no. 6, pp-. 2566-2570.

[7] Maclean, N. et al.: The concept of patient motivation: a qualitative analysis of stroke professionals' attitudes. Stroke; a journal of cerebral circulation, (2002), vol. 33, no. 2, pp. 444-448.

[8] Burke, J.W. et al.: Optimising engagement for stroke rehabilitation using serious games. The Visual Computer, (2009), Vol. 25, no. 12, pp. 1085-1099.

[9] Levitt, S.: Treatment of Cerebral Palsy and Motor Delay. John Wiley & Sons, (2013). ISBN 9781118699799.

[10] Scrutton, D.: Management of the Motor Disorders of Children with Cerebral Palsy, Cambridge University Press, (1984). ISBN 9780521412100.

[11] Norouzi-Gheidari, N. et al.: Interactive virtual reality game-based rehabilitation for stroke patients. In: 2013 International Conference on Virtual Rehabilitation (ICVR), (2013), pp. 220-221.

[12] Smith, J.L.: Activities for Gross Motor Skills Development. Teacher Created Resources, (2003). ISBN 978-0-7439-3690-3.

[13] Smith, J.L.: Activities for Fine Motor Skills Development. Teacher Created Resources, (2003). ISBN 978-0-7439-3689-7.

Extended abstracts

Beyond Adaptive Web Design

Ján ANTALA[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
hello@janantala.com

Extended Abstract

The Web is continually evolving and we need to evolve with it. It used to be easier to manage browsers when there were just a few of them on the desktop. Today we not only have to deal with a wide range of desktop browsers but mobile devices, tablets, televisions, weareable devices and more. Even for the average web site things have changed a lot over four years: browser share, operating systems, screen resolutions, and more [3]. The basic approach how to provide an optimal viewing experience is to use "Responsive web design". While creating flexible layouts is important, there is a lot more that requires remarkable adaptive web experiences.

Adaptive web design is fundamentally "Progressive enhancement", but it is being applied to a much larger and more diverse landscape. We now have Web-enabled smartphones, tablets, e-readers, netbooks, watches, TVs, phablets, notebooks, game consoles, cars and more. We also have many types of internet networks with different speed, latency and quality. However there are many more factors we need to think about. It is also important to consider as well ergonomics, input methods, internet connections and other features that can be detected. We consider "adaptive web design" as an equal with creating a single Web experience. We can adjust it based on the capabilities of the device and browser. Website can access sensors in devices and use them to enhance user experience.

We have detected several web components [1] and input methods and tried to verify which of them provide better experience for the website users and are also useful for the web developers. There have been made series of experiments on both adaptive input methods and adaptive web components. To prove the usefulness of the proposed concept we have designed and implemented several reusable modules and components and we have used them in propotype web projects. We have tracked amount of saved web traffic and web requests, webpage rendering time and interest in alternative input methods. We have also received a lot of feedback from the community which helped us to verify the proposed concept. The experiments have been attended by thousands of visitors.

Adaptive input methods provide alternative way of web application control and extend current approach. We have been experimenting to control web applications using voice commands, motion detected by a gyroscope and a video camera.

[*] Master study programme in field: Information Systems
Supervisor: Assoc. Professor Michal Čerňanský, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 573–574.

We have created an experiment based on our module which provides voice control over to-do list. There were 7 voice commands using exact expressions and 9 using regular expressions. This has been the most popular input method and the experiment has been attended by thousands of users. One of the biggest problems of the voice commands is however incorrect speech recognition. To solve this issue we can conditionalize regular expressions or use utterance error correction.

The another experiment based on gyroscope rotation which allows to scroll in the application has been attended by thousands of users. However some problems have been detected. The biggest one is that browser vendors do not use specification correctly and use own orientation ranges and directions. This causes some scroll issues.

Almost every modern device contains a video camera. This is a great opportunity to control web sites using motion gestures. This experiment has been attended by hundreds of users. Using video motion detection we can enhance experience in websites and games. We can also detect device motion and orientation changes so it can be used as a supplementary input method to the accelerometer and gyroscope rotation. However we cannot take a full control over the web app and need a proper ambient lighting. There can be detection issues in too dark and too bright scenes.

There are many commonly used web services that produce many requests and unnecessary traffic even when the website user does not want to use them. There is lot of studies that highlight the importance of speed. More than half of people with a bad loading experience on mobile, will not come back [2]. 73% of mobile internet users say they have encountered Web pages that are too slow [4]. There are also native mobile application for that services with better perfomance which provide full user experience. This is a great opportunity to utilize conditional loading to serve the best experience for the right context. We have used the Mobile First principle to develop Google Maps and Youtube videos web components. Using adaptive web components we reduce initial traffic by 400 kB and a page load times by 350 ms for a youtube video in the average. The amount of saved traffic and load times are various for the map elemets and depend on element size. This approach works quite well for simple use cases. We can show multiple map types, markers or videos. However, more interactive elements require additional consideration. Even then use of an embedded elements directly still might not make sense because of unnecessary traffic. We can still use adaptive techniques to reduce it and replace basic static image with richer elements.

We have received a lot feedback and have usage and popularity results of adaptive input methods. As we have expected, the most popular input method is speech input since we can control an entire web application using only voice commands. Gyroscope and video motion detection are also useful input methods, but there are limited possibilities where to use them and can be used as supplementary input method only. All of them however provide a great opportunity to enhance user experience. Adaptive web components are a great to save an unwanted web traffic. They also increase user experience because the web page produces less requests and loads faster. However the limitations also exist because many websites need custom elements and design. As a result the adaptive web components produce less interest than adaptive input methods.

More changes are coming. So we have to prepare the Web to this evolution and provide the best user experience, design for many inputs and save the inessential web traffic.

References

[1] Bidelman, E.: Web Components: A Tectonic Shift for Web Development. In: *Google IO*, San Francisco, CA, USA, 2013.

[2] Grigorik, I.: Optimizing the Critical Rendering Path. In: *Breaking Development*, San Diego, CA, USA, July 22-24, 2013.

[3] Gustafson, A.: Building Adaptive Designs Now. In: *User Interface 17 Conference*, Boston, MA, USA, November 5-7, 2012.

[4] McLachlan, P.: Page Speed is Only the Beginning. In: *Breaking Development*, San Diego, CA, USA, July 22-24, 2013.

An Approach to Capturing Intention in Source Code

Michal BYSTRICKY*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`m@mby.sk`

Extended Abstract

In 1972, there was a concept of Dynabook—a personal computer for children of all ages [4]. Kay introduced a personal computer with a software environment where the users, primarily children, would be able to edit their own files and programs in order to extend their mental model. For example, one of the first programmes a user would write was a filter to eliminate advertising. In this concept, everybody could do programming and customize their own environment. The result of Kay's research was Smalltalk—a reflective programming language. A programmer could extend a Smalltalk instance at run time and create a living system.

We can see a huge difference, a gap between the Dynabook concept and nowadays programming. Programmers have no chance to develop software in such an easy way. There is a huge number of things they have to handle in order to create a program.

Data, Context and Interaction paradigm (DCI) is about representing users' mental model to source code [2]. Users are engaging with software at deep level by creating stories and use cases for programmers [3]. Then, people can comprehend the code easier, there is less rework and system is consistent. If the program does not represent the way how users think, they will not be able to use it. On the other side, if the program represent programmer's thinking, users have to understand complicated aspects of programmer's mind [1].

Also AOP can improve modularization by allowing the separation of crosscutting concerns. Basically AOP extends OOP about some useful extensions, which can be used to improve intention in source code. Either way, the AOP is considered asymmetric, so aspects are externally accessing to source code without leaving trace in source code. A programmer cannot really see which method is he doing while looking at the method. However we could make source code more lightweight, e.g. to extract specific low-level source code or extract design patterns to make algorithm more visible. Therefore AOP can be used to make source code lightweight and it can improve intention in source code.

DCI and aspect oriented programming (AOP) have in many ways the same goal - to improve intention in source code. For example, they are both able to divide source code to use cases.

* Master study programme in field: Software Engineering
Supervisor: Assoc. Professor Valentino Vranić, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

However, most innovations that AOP provides are closely related to the programmer's perspective, but DCI is focusing more on the end user's mental model. AOP can be viewed as the opposite to the DCI, because in specific cases the source code in AOP can be confusing and thus the programmer is losing intention in source code.

In this work we analyze approaches to software development like DCI and AOP that can be used to improve intention in source code. By comparing these approaches we were able to identify techniques, which help to capture intention of programmers while programming. One of the results is that by pulling out source code to use cases, the programmer can see all related source codes in one place and thus he does not have to go anywhere else to see related source code. Searching for such related source code can cause the loss of focus.

Our approach modularizes source code to use cases, because use case stories capture user's intent. Then use cases extend object model about new functionality by decorator or annotation located in an object model. This way we can create mappings in object model to use case classes. Therefore a programmer can see algorithm in use case classes, object model is lightweight and moreover he can see a big picture (connections, mappings to use cases, behaviour) of an application by looking at the object model.

As we mentioned, the extending of basic model is done by decorator. The decorator used in our approach is similar to control flow pointcut type in the AOP, because it operates in a flow. The difference is that the decorator is not asymmetric as control flow pointcut, thus injection of source code can be seen in the basic model.

The result is use case classes that contain specific behaviour (methods) for specific use case and the basic model, which maps methods (e.g. empty methods) to use case classes. A programmer by looking at the class of the basic model can see a big picture instead of specific implementation details. He can focus more on his intent, because related source code is in use case and the base model is organized and easy to navigate.

Therefore we achieved to make the object model more lightweight. This object model extends and maps its behaviour to use case classes. The algorithm is visible in use case and a programmer can easily comprehend big picture of an software by looking at the object model.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Coplien, J.: James O. Coplien, Gertrud&Cope, is being interview by Kresten Krab Thorup from Trifork. GOTO International Software Development Conference, 2008.

[2] Coplien, J.: The DCI Architecture: Supporting the Agile Agenda. Oredev Developer Conference, 2009.

[3] Coplien, J.: The DCI Architecture: Lean and Agile at the Code Level. QCon International Software Development Conference, 2010.

[4] Kay, A.C.: A Personal Computer for Children of All Ages, 1972.

Group Recommendation of Multimedia Content

Eduard FRITSCHER*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
Eduard.fritscher@gmail.com

Extended abstract

Nowadays we are literally overwhelmed with amount of information available. As the world changes, the access to the Internet also changed. People collaborate more often with each other the recommendation techniques have to adapt to these new trends mainly collaboration between users. Typically the solution to this problem is the use of collaborative recommendation methods, but often the target of the recommendation is not a single user but a group of users. This is the point when group recommendation comes in handy. But in a scenario when we are recommending to a group of users we have to consider the satisfaction of each user and the overall satisfaction of the group. Each user in the group could have different taste and preferences in the domain in which we are trying to recommend content. Different personalities could also influence the group.

The aim of our proposed method is to enhance group recommendation with personality awareness and the power of graph traversal algorithms. In our method we will build the personality models by using data extracted from social networks.

Recommender systems are nowadays a popular research domain. Two basic approaches are used in order to generate recommendations: collaborative filtering and content-based [2]. Both approaches could be improved by using graph representations. When we talk about group recommender systems three main approaches are generally used in order to solve conflict preferences - preference aggregation, recommendations aggregation or group model construction [3]. We have no information about a group recommender system that uses graph algorithms for recommendation and includes personality awareness, but there are a few approaches that include both. In the work of Juan A. [4] they have proposed and implemented a group recommender system that uses the Thomas-Kilmann Conflict Mode Instrument to include personality awareness.

The main contribution of our method is to enhance group recommendation with personality awareness. To accomplish this enhancement we decided to add the Big Five personality model because of its wide usage. We can split our recommendation method into four main steps data extraction, personality model generation, aggregation strategy and graph recommendation (Figure 1). The system Televido provides us with the user preferences. The data that we need to generate the personality models for each user will come from the integrated social networks like Facebook with the use of its graph API. After the necessary information about the users from the group are available, the personality model generation begins. The process used for generating the personality models is described in the [1] where the authors identified correlation between data stored on Facebook and the Big Five personality model. The next step after the personality models are

[*] Master degree study programme in field: Software Engineering
Supervisor: Michal Kompan, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

generated, is to combine the preferences and personality of each user to single group model. This group model represents the preferences of the whole group towards to the starting nodes of the graph recommendation algorithm. For the group model generation we use our aggregation strategy which combines the personality models with the user preferences. The preferences used for each user are selected using TOP-N selection.

The aggregation strategy increases or decreases the weight of preference provided for the user depending on the difference between the positive and negative characteristics. After the aggregation strategy changed the weights of the starting nodes the graph recommendation will be applied which we describe in next section.

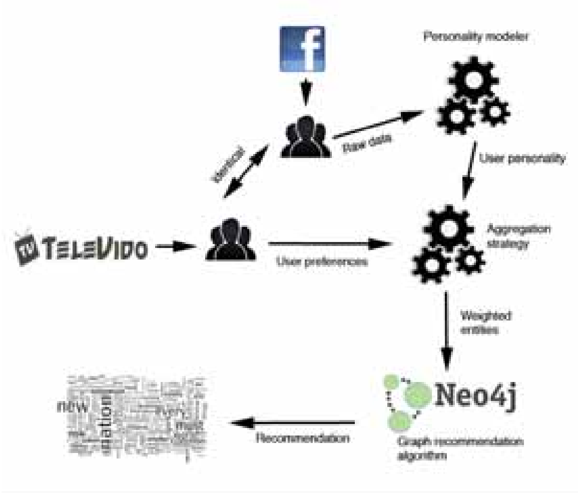


Figure 1. Personality aware group recommendation method.

Our recommendation method is based on a graph algorithm that traverses threw initial starting nodes. The initial nodes are weighted and represent the preferences of the group. The algorithm we use, is a modified version of the activation spreading algorithm.

Acknowledgement: This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

References

[1] Yoram Bachrach, Michal Kosinski, Thore Graepel, Pushmeet Kohli, and David Stillwell. 2012. Personality and patterns of Facebook usage. In *Proceedings of the 3rd Annual ACM Web Science Conference (WebSci '12)*. ACM, New York, NY, USA, 24-32.

[2] De Pessemier, T., Dooms, S., & Martens, L. (2012). Design and evaluation of a group recommender system. *Proceedings of the sixth ACM conference on Recommender systems - RecSys*'12, 225.

[3] Masthoff, J., & Gatt, A. (2006). In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, 16(3-4), pp. 281-319.

[4] Juan A. Recio-Garcia, Guillermo Jimenez-Diaz, Antonio A. Sanchez-Ruiz, and Belen Diaz-Agudo. 2009. Personality aware recommendations to groups. In *Proceedings of the third ACM conference on Recommender systems (RecSys '09)*. ACM, New York, NY, USA, 325-328.

Monitoring of Process Resources Usage

Pavol FÜLÖP[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`pavolfulop@gmail.com`

Extended Abstract

Memory is one of the main resources for a process. It is up to programmers how they use it, and if their use is efficient enough. For purposes of evaluating memory usage efficiency, a mechanism for tracking memory allocations is needed. Since not all memory requirements for program runtime are known during the process of programming, dynamic memory allocation often takes place. However, tracking these allocations is not an easy task.

For purposes of dynamic memory allocation in C, functions `malloc()` and `free()` are used. When the `malloc()` function is called inside a program, and the heap is not large enough, system call `brk` is executed. If `brk` were called every time `malloc()` is used, it could result in slow performance of the process. This is mainly due to the fact that `brk` is a system function and also because `brk` needs to check if the process is allowed to do so or if the enlarged heap does not overlap some other memory region [1]. With this fact in mind, there is no direct option how the kernel could know how much memory a program is actively using, because heap management is left to the process itself.

To fully monitor memory usage of memory, we need to watch every memory allocation and deallocation in program. This information can then be preserved trough time to give back information suitable for programmers. In regard to the fact that usage of memory is left to the process, our solution is included directly in code.

Our solution is basically a library: it consists of functions for measuring time (three options are available), options for outputting results (stdout, stderr, file), options for storing made allocations, time delay between providing results, and in case of using the library with thread support, if the methods should be synchronized or not. Thanks to configurations, the programmer is able to see in real time what is happening in the heap. The basic information provided are time of the most recent memory change and total amount of memory used by a process represented by just one scalar value M, i.e. the average value per a unit of time. It can be calculated as an integral over the time interval (given by Equation 1).

$$M = \int_{t_1}^{t_2} m(t)dt \tag{1}$$

[*] Bachelor study programme in field: Information Systems
Supervisor: Dušan Bernát, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

Where m(t) is the amount of memory occupied by the process as a function of time. Some basic properties of this function are determined by the virtual memory system. We suppose, that allocation (or releasing) of memory is always done in a single time instant, so m(t) changes its value only at discrete points t0, t1... tn. Between the two points of change, its value is constant.

To get the amount of memory occupied by the process, we need to access sizes of allocations and deallocations at the time of allocation or free. When it comes to `malloc()`, the information is required by the function itself, but in case of freeing the block, there is no information carried along the process.

Of course, all data about allocation and freeing memory are pretty much useless without the corresponding time. There are multiple ways implemented in the operating system to measure time.

Three ways of measuring time are available in standard GNU C libraries: RDTSC (times stamp counter), CLOCK_GETTIME, TIMES. RDTSC is very accurate because of cycle measurement, but because of that, specific conditions need to be met. One of those conditions is the presence of TSC register. TSC is a 64-bit register that counts number of cycles since reset. These cycles can then be converted to time, when number of cycles per second is used. Multiple CPU cores can provide wrong results in case the TSC registers for each core are not synchronized. Even hibernation or power saving capabilities of the CPU can invalidate the results. CLOCK_GETTIME is another option with has multiple clocks to choose from. However, the variety of clocks depends on the operating system. And the last one is `times()`. This option is most universal among the others.

One of the aspects that has to be considered is multithreading. Programs that are using advantages of multithreading must take concurrency into account. If multiple allocations occur at the same time, the stored time and size of variable data will be accessed at the same time from multiple threads. For this purpose there is an option to choose if programmer made the code synchronous enough and there will be no same access to `malloc()` or `free()`, or there is an option to let the library handle locking.

In our solution, to ensure two threads are not in same critical section, we use Mutual Exclusion. In case of accessing `malloc()` or `free()` methods in time when log or thread initialization is made, mutex is locked before doing any of the operations. If the mutex has been locked before, operation will wait until other threads leave the critical section. To better track memory usage of individual threads, every thread has an index assigned in the initialization of the thread. This index ensures individual treatment of each thread.

Application memory management is one of the main goals that should be encouraged in programmers. Some programming languages and developer kits provide enough tools for making such management easy. This tool will hopefully provide information needed to find any memory leaks and increase proper memory usage.

Acknowledgement: We are grateful to Slovak Science Agency VEGA which supported this work by the project No. 1/0722/12.

References

[1] Bovet, D.P., Cesati, M.: *Understanding the linux kernel.* O'Reilly Media, Inc., O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2005.

[2] McCallum, J.C.: Memory Prices (1957-2013), 2013.

[3] Valgrind: Valgrind. `http://valgrind.org`, [Online; accessed January 23th, 2014].

Object Recognition Based on a Description of the Superpixels Neighborhood

Martin GEIER*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
geier.xmartin@gmail.com

Extended abstract

Nowadays there are no limitations in audiovisual content creation. There are many devices around us, which can create such content like mobile phones, cameras etc.

This content is mostly described only by wordy description entered by its creator. This process leads to wide gap between creation of audiovisual content and it's accessing for other people. To overcome this gap it is necessary to create and combine appropriate methods of the automatic image recognition.

One of the most used approaches of the image recognition is image description by descriptors. There descriptors are classified later. In this way there were created some common used algorithms, for example algorithm for face or people recognition [2]. We use this concept in this algorithm too.

The base steps of the algorithm are:

- object segmentation, not a key part of this work, but can be done in the future,
- superpixel segmentation,
- overlaying element through the superpixels.

Input of the algorithm is already segmented image into foreground and background. The foreground represents the analyzed object and the background is an image fill. In segmented image there is necessary to cluster into superpixels which will outline the image [1]. Created superpixels are clearly divided into superpixels belonging to recognized object and superpixels belonging to background, see Figure 1. The recognition algorithm uses a structured element. The element is set over the center of superpixels belonging to recognized object. The beams of the overlayed element clearly identify the neighbors we are testing, figure 1. The algorithm decides whether the target superpixel belongs or does not belong to the recognized object. These logical values create a binary vector. By overlaying the element through all superpixels of recognized object the algorithm creates a set of vectors. The composition of the same vectors positions creates a histogram of superpixels occurrence frequencies belonging to recognized object.

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Vanda Benešová, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

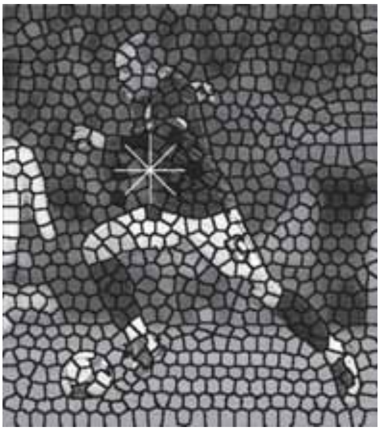


Figure 1. Element overlayed through a superpixel.

By overlaying the element through all superpixels of recognized object the algorithm creates a set of vectors. The composition of the same vectors positions creates a histogram of superpixels occurrence frequencies belonging to recognized object. Adding more beams to the element is a simple way to change the length of descriptor and increase the ability of image recognition. Thus the element is only a composition of tested superpixels, one beam may include various test points and may test multiple superpixels in one direction. An important factor in recognizing images is scalability invariance. Therefore it is necessary to change the structured element to always show the various dimensions of the image as the same superpixels. Element is not defined as an absolute distance from the center of the superpixel to the adjacent superpixel, but as a decimal number indicating the ratio of the length and width of the image.

Descriptor accuracy was evaluated in combination with SVM classifier. SLIC algorithm was set to make 400 superpixels. Structured element has eight beams and element shape was asymmetrical. Input for evaluation was binary pictures, where pixel values determine foreground and background. As positive trained data were used set of forty animated football players and as negative trained data were used furty random pictures. Test data contained segmented pictures of football players from video record and horse pictures from different views. Global accuracy of this approach is 87.5%. We plan to evaluate our method on larger dataset.

Table 1. Method accuracy

Image type	Number of images	Correctly recognized	Accuracy
Football player	71	61	86%
No football player	71	63	89%

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Achanta R., Shaji A., Smith K., Lucchi A., Fua, P., Süsstrunk, S.: SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, pp. 2274-2282.

[2] Jiang M. R., Sadka A. H., Zhou H.: Automatic Human Face Detection for Content-based Image Annotation. In: *Centre for Sensor Web Technol*, pp. 449-454.

Towards Feature-oriented Software Development with Design Features

Michal GRANEC*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xgranec@fiit.stuba.sk

Extended Abstract

The software product line approach is one of the most successful approaches in the area of software reusability. It is based on a fact that products that share domain are similar. This approach was designed for software development of a family of similar products.

Feature modelling as part of a domain analysis has proved to sufficiently capture the varied and common parts of software product line. It also provides the possibility of configuration of software product line. Depending on selection of features in the feature model configuration the features of certain software product are defined.

The most used and highly rewarding strategy in the process of creating a new software product line is to derive it from existing products. There are several challenges connected with this strategy. One of the challenges lies in the identification of features in existing products which in most cases do not have feature model. The second challenge is proper connection of software artefacts with features. This connection will allow effective generation of products from a software product line based on features that are selected in the feature model configuration.

Features in FOSD (feature oriented software development) are considered to be the first class objects that are used for building software products [3]. The main process of FOSD consists of modularization software into feature modules. The software product is then built by these feature modules depending on selection of corresponding features in feature model configuration.

However feature can be seen as very abstract and not concrete enough. It can be really difficult to connect the software artefacts with adequate features. Furthermore code scattering and its entanglement doesn't help this problem at all. The need for an additional layer between features and software artefacts led H. Lee and K. Ch. Kang to propose a new approach called design features [1]. The design features represents new concept of features that are designed to better connect source code with feature model. The problem with the approach of design features is that it came with the idea of separated design feature model. Combination of feature model and the design feature model can lead to automatic generation of products from software product line and thanks to the use of design features the feature modules will be small enough and not too complicated for implementation.

For purposes of creating a combined feature model it is necessary to extend basic Czarnecki-Eisenecker notation by additional markup. The new additions are abstract and concrete features.

[*] Master degree study programme in field: Software Engineering
Supervisor: Assoc. Professor Valentino Vranić., Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

They are suitable candidate for integration of these two models. Abstract features are features that are not mapped to any feature module and therefore have no impact on the implementation level. The other features are called non-abstract or concrete features and they are mapped to at least one feature module [4].

To address the problem of feature identification I have suggest a reversed process of creating design features together with partial feature model to avoid the need of creating full feature model [6]. This way the technique is used to model only optional abstract features and leave out the mandatory features which represent the essential core of software product line. In several steps we assign to every block of code that is different in each product a design feature and get a set of potential design features in the end. In the next steps we identify the parent features of these design features and also relation between these discovered design features. This process can repeat several times until there are no other potential parent features or relation to discover.

To be able to generate products from software product line it is necessary to create the common partial feature model of software product line. The model can be created by merging feature models of each product into one common. Merging can be done under certain conditions and it is based on representation of feature model in formal notation [5]. After the feature models are expressed in formal notation then the merging is performed as a sequence of set operations. To complete the process of creating a software product line the feature modules need to be implemented. This can be done in various ways for example we can use aspect-oriented programming and in this case the aspect-oriented refactorization is required in the terms of extracting features from existing objects into aspects [2].

This method is also tested on the creation of software product line in order to evaluate its efficiency. The test scenario models a situation in which a software company develops a complex product for first customer and then it is simplified for second customer. Some of the first product functionality is removed and some is added. Additional evaluation is needed in future work in the form of software metrics or other ways of measuring the efficiency of the proposed method. It can be also interesting to add another sample product to test set and this product could be at the intersection of two domains.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Lee H. and Kang K.C.: A design feature-based approach to deriving program code from features: a step towards feature-oriented software development. In Proc. of the 7th Int. Workshop on Variability Modeling of Software-intensive Systems, VaMoS '13, New York, NY, USA, (2013), ACM, pp. 51-56.

[2] Monteiro M.P. and Fernandes J.M.: Object-to-aspect refactorings for feature extraction. In Proc. of 3rd Int. Conf. on Aspect-Oriented Software Development, ACM Press, (2004).

[3] Thüm T., Kästner C., Benduhn F., Meinicke J., Saake G. and Leich T.: Featureide: An extensible framework for feature-oriented software development. In Sci. Comput. Program., (2014), pp. 79-85.

[4] Thüm T., Kästner C., Erdweg S. and Siegmund N.: Abstract features in feature modeling. In Software Product Line Conference (SPLC), 2011 15th International, (2011), pp. 191-200.

[5] van den Broek P.: Intersection of feature models. In Proc. of the 16th Int. Software Product Line Conference - Volume 2, SPLC'12, New York, NY, USA, (2012), ACM, pp. 61–65.

[6] Vranić V., Menkyna R., Bebjak M. and Dolog. P.: Aspect-oriented change realizations and their interaction. In *e-Informatica Software Engineering Journal*, Vol. 3, (2009), pp. 43-58.

Retaining Use Cases at Source Code Level

Ján Greppel[*]

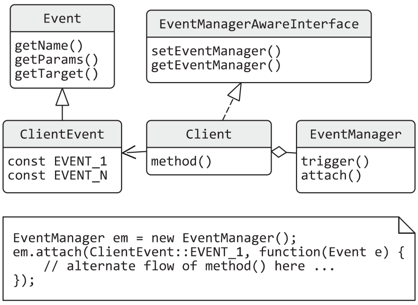
Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`xgreppel@stuba.sk`

Extended Abstract

Over the years, use-case driven development has yielded positive results and it has earned the reputation of accepted technique for modeling of system behavior. However, as use cases have to be transformed into a source code, crosscutting concerns emerge and use cases are no longer primary focus. As a result, verifying the use cases become more difficult [3], request management gets more complicated and readability of the source code decreases.

To address this issues, two new proposals are presented. One of them is called The Event Pattern, which retain so-called extension use cases. The second proposal focuses on organization of whole use cases in the system with Front Controller Pattern to foster cleaner code and better retention of use cases at source code level.

Every extension use case has extension points that are defined in other use cases [2]. Once these points are defined in the code as one-liner, extension use case can be attached to them. This can be emulated by OOP design pattern with one object holding mappings of defined points and attached behavior. Class diagram of this pattern is shown in Figure 1.



```
EventManager em = new EventManager();
em.attach(ClientEvent::EVENT_1, function(Event e) {
    // alternate flow of method() here ...
});
```

Figure 1. Class diagram of The Event Pattern.

[*] Master study programme in field: Software Engineering
Supervisor: Assoc. Professor Valentino Vranić, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 585–586.

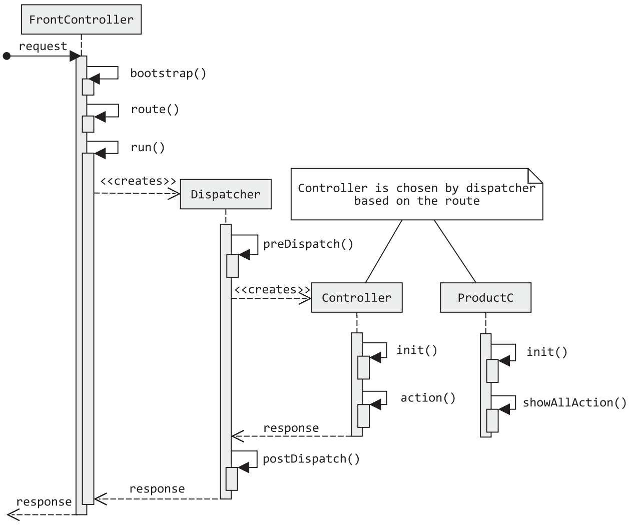


Figure 2. The Front Controller pattern.

In addition to retaining of extension use cases, Front Controller Pattern [1] can be used to organize whole use cases (or at least larger blocks of use cases) into respective controllers. This pattern, as is shown in Figure 2, provides centralized way of handling requests. Every request made by the user will be caught by Front Controller and (based on the actual request) dispatched to given controller, that will handle the request. Therefore, use cases as the controllers will be first application's classes that will be executed upon user interaction.

The benefit of this approach gives the programmer better overview of application's behavior and offers the way to organize new use cases into the system without loosing sight of all use cases in the system. Moreover, since specifications will be in the form of use cases, implemented use cases are probably the first software artifacts that will programmer work with when changing the functionality or creating new ones. That's why keeping use cases together in one place can enhance understandability and degree of the clean code.

Retaining use cases in the source code is not trivial problem. The Event Pattern offers alternative solution for retention of extension use cases, thus providing the choice to use most appropriate solution for meeting the needs of software engineers. Also overall organization of the use cases can be enhanced with the Front Controller patterns. In the future, these two approaches should be unified and given as a whole. In addition, more solid example of the case study should be demonstrated.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Wiley, 1996.

[2] Jacobson, I., Ng, P.W.: *Aspect-Oriented Software Development with Use Cases*. Addison-Wesley, 2005.

[3] Kannenberg, A., Saiedian, H.: Why Software Requirements Traceability Remains a Challenge, 2009.

Hand Gesture-based Language and its Application in a Game

Marian KURUC[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`mariankuruc1@gmail.com`

Extended abstract

Human-computer interaction (HCI) provides multiple ways to communicate with computers. One of the continuously evolving fields is the one that deals with alternative methods of input. We focus on the use of 3D hand-gestures to control applications, which are received from a gesture recognition device. Applications that use various types of gestures often utilize their own set of gestures, which makes it hard for people to remember. Gestures are most of the time designed poorly, without taking into account the two main principles. One of them tells us that gestures need to be designed intuitively for people to be easily remembered. Each gesture also needs to be significantly different to the point that a computer can uniquely identify it. As of today, there is no language of hand gestures that is being used application-wide. The lack of this makes it a problem.

However, there already are a few languages of gestures. Touch gestures performed on mobile phones, which are used mostly across all applications are an example of a small language. But this language is limited by a very low amount of gestures. Another example is a sign language that uses gestures detected by a Kinect device. This language unfortunately assumes you already know the sign language and is only usable by a small circle of people. We have created our own intuitive hand gesture-based language that counters these problems, and later on we would like to extend it with different types of gestures. We also want to release this language for public use and will try to make it usable in games as a standard. Our language has an advantage among others, it learns and constantly evolves based on the gestures people use the most and the ones that are the most intuitive.

We have analysed existing recognition devices and decided to use the Leap Motion controller device for gesture recognition, as it is currently one of the most accurate devices. A gesture in our language is a set of movements one can do with fingers on both hands. Even a slightest movement of rotating a single finger is detectable and may be usable as a gesture in the language. Obviously, the longer the set of gestures is, the harder is for user to follow the pattern and successfully perform a given gesture. System predicts what gesture a user wants to use and has a defined margin for error which helps to detect a gesture. Leap motion surprisingly can detect a gesture even at a quite high speed, but the detection effectiveness decreases the higher the speed of the movement is. Recognition of gestures is initialized by placing hands above the Leap Motion

[*] Master degree study programme in field: Information Systems
Supervisor: Dr. Juraj Štefanovič, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

controller and finalized by withdrawing them, or after a certain time passes. A gesture is limited by the duration of two seconds and each sentence can contain three of them, but this is still a subject of testing and may change in the future. Gestures are inserted into sentence after detection and evaluated. They provide us with an output in the form of an action.

To be able to add a new gesture to the language and later recognize it, we have designed an initial solution to determine position and rotation of every single part of each hand, divided by bones. Next, thanks to the limited movement one can do with each part of the hand, we have a limited set of motions, for which we assign a given number or letter. Recorded gesture contains 12 characters (10 for each finger and 2 for each hand), which are stored in a hash table with the characters serving as hash, thanks to which we can increase the speed of the search. This way the language constantly evolves and as of now it contains around 20 different gestures.

Designed test evaluates every new gesture and determines its similarity to the gestures that already exist in the language and might decline them on the spot. Without a direct human contact we are unable to test its intuitiveness. That is why we have designed a game, which collects data from players while it is being played. Example of the game controlled by the Leap Motion device can be seen in the Figure 1.

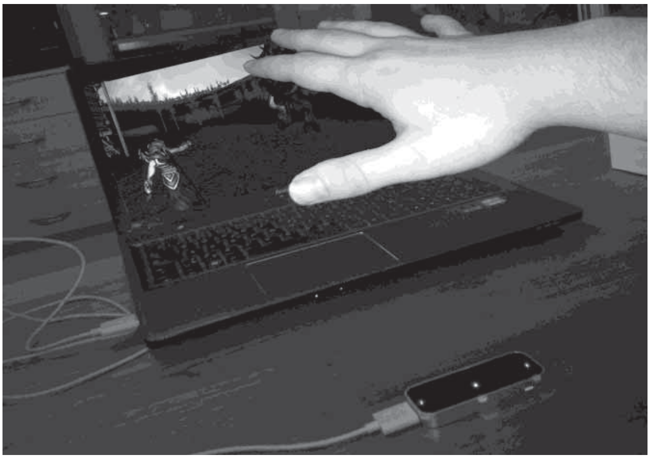


Figure 1. Sample of the leap motion controller in action (designed game).

The game is implemented using the Unity3D engine. The point of the game is to defend a village that is being attacked by monsters. Player is controlling a chosen character, which interacts with the events in the game and fights monsters. Player controls the game with hand gestures, and plays with either the default set of gestures, or he can define his own and assign them to actions in the game. Default gestures change over time as the language evolves and are set to the ones players use the most.

References

[1] Unity3D: Create the games you love with Unity. [Online; accessed February 18, 2014]. Available at: http://unity3d.com/unity

[2] Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the Accuracy and Robustness of the Leap Motion Controller. Sensors (Basel). 2013 May; 13(5), pp. 6380-6393.

[3] Rautaray, S., Agrawal, A.: Vision based hand gesture recognition for human interaction. Springer Netherlands. 2013. pp. 0269-2821.

A New Innovation in the e-Learning Systems: Knowledge Testing with Graphical Input

Lukáš LENČÉŠ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
luckhass@gmail.com

Extended abstract

A complex e-Learning application consists of several components and modules, such as Learning Content Design System (LCDS), Learning Content Management System (LCMS) and Learning Support System (LSS). A parts of LCMS are *Knowledge Capture Tools*, or sometimes called *Assessment Tools* [1].

These knowledge testing tools are often based only on text input, selection from the offered options or selecting options TRUE/FALSE. More sophisticated tools support method Drag and Drop. Answers from these tools can be easily processed by evaluation system because they don't need to be transformed other (logical) form for evaluation.

There is no tool for knowledge testing with drawing answers with automatic evaluation in the LMS and LCMS. Some companies have developed their own proprietary systems for learning and testing in specific area (e. g., Cisco has *Cisco Packet Tracer* to learn about computer networks).[1]

Another tools are *Hades*[2] (Hamburg design system) and *CircuitLab*[3] which allow rendering a logic circuit. However these programs allow only to simulate your drawn logical circuit and do not support automatic answer evaluation.

A major disadvantage of nowadays assessments tools in LMSs and LCMSs is the absence of ability to draw the answer. This limits the efficient use of e-Learning systems, especially in technical fields. For example, in computer science we can draw logic circuit or computer network. Automatic evaluation of a response that can be drawn is more complicated.

In this topic we introduce one solution to the problem of evaluating drawn answers. When we want to evaluate logical circuits, we need to transform them to some data structures. Next a computer can compare these structures and evaluate them. The algorithm, which we present here, is using graph data structure. This algorithm, named SiaL, is a solution to problem of comparing graphs and was presented in [2].

Figure 1 presents a (simplified) design of tool for knowledge testing based on drawn answers. There are three modules in this knowledge testing tool.

[*] Master degree study programme in field: Computer Engineering
Supervisor: Boris Dado, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

[1] https://www.netacad.com/sk/web/about-us/cisco-packet-tracer

[2] http://tams-www.informatik.uni-hamburg.de/applets/hades/

[3] https://www.circuitlab.com/

The *Drawing Module* provides the possibility of drawing the students' answers using the appropriate graphics libraries. For example, students can here draw logical circuits and this module saves these drawings as a graph data structure.

Creating XML file module transforms the graph data structure to a XML format.

The *SiaL* module implements the SiaL algorithm, which we are describing.

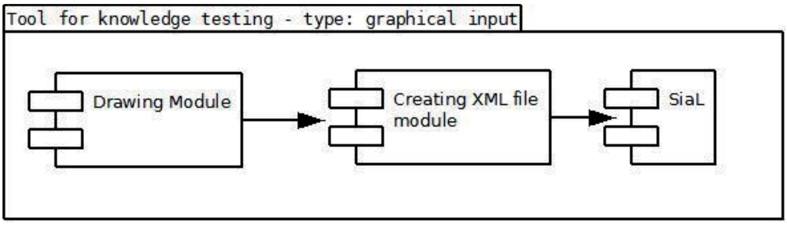


Figure 1. Tool for knowledge testing with graphical input.

SiaL is an algorithm which fully automatically compares two graphs given in XML format. For example, below displayed two simple logical circuits (saved in computer as graphs) were drawn (one representing correct answer and another one representing student's answer). The function of SiaL algorithm is to find differences between them. Output of this example will be: in second graph is missing one vertex and one vertex is with bad type.

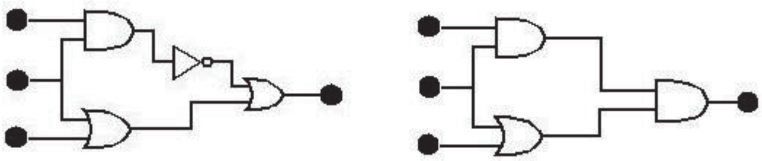


Figure 2. Two graphs as a simple input for SiaL.

Algorithm SiaL takes into account more details in graphs, such as types and weight of vertexes / edges, orientation of edges and finds the most similar sub-graph.

The principle of SiaL is based on mathematical theory of finding isomorphism between two graphs. Our solution is based on graph represented by an adjacency matrix. The isomorphism can be found by permutations over rows and columns. Algorithm SiaL modifies adjacency matrix and inserts all attributes from XML file of appropriate graph into it. This modification allows to find all differences between compared graphs.

Another function of SiaL is to find the best matching sub-graph. This function is necessary because computer does not see compared graphs as a picture. In mathematics we can say: yes, these graphs are isomorphic. But in knowledge testing we need to say how many differences are there; in which vertexes/edges; is the weight of difference between types of vertexes / edge more than weight of missing vertexes/edges? All these and more questions, as well as answers to these questions can be found in [2].

References

[1] Ismail, J.: The design of an e-learning system, Beyond the hype. In: *The Internet an Higher Education, (2001),* vol. 4, pp. 329-336.

[2] Lenčéš L.: *Comparing Graphs.* Bachelor's thesis, (2013) (in Slovak).

Image Processing on System-on-Chip Platform

Ján MAZAG*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xmazag@fiit.stuba.sk

Extended abstract

The limits on image processing power and performance consumption were lowered by never-stopping increase of computing power on standard PCs, which allowed developing powerful algorithms processing image and video. PCs offer relatively satisfying amount of memory and processing power with which we can freely implement neuron networks, self-organizing maps or any memory and processing-demanding algorithm. Devices usually perform satisfactorily.

The aim of this work is to implement an application on a system-on-chip device that will not be affected by physical conditions and show that System-on-Chip devices can be a relevant alternative to desktop computers providing satisfactory conditions for image processing algorithms. This can be accomplished by using robust yet compact device that contains sufficient architectural and hardware features to run demanding algorithms. There are multiple vendors implementing useful technologies such as DSP to meet this goal.

Practical image and video processing algorithms consist of specific data intensive high-bandwidth, low and intermediate-level operations such as feature extraction or filtering. They also include irregular high-level low-bandwidth operations such as classification. The development and research was aimed to eliminate bottlenecks in order to speed up the process. Low-bandwidth processes can be improved by hardware modifications to give more time to perform high-level operations [1].

Digital Signal Processor (DSP) is a microprocessor with high performance, low power consumption and small size. That allows it to process intensive tasks and be used in embedded devices. In the early era of DSP they were not suitable for image or video processing. Their bandwidth requirements were not high enough at that time and did not meet video and image requirements. Newer high-performance DSPs are architecturally designed to address the throughput barrier. DSPs are very efficient in executing critical loops containing very little branching and control operations [2].

The purpose of proposed System-on-Chip application is to perform any type of image or video processing algorithm and be able to map one of several output signals to it. This means that user is able to create a new algorithm, add it to the “pool” of runnable algorithms, modify the configuration of the algorithm in its runtime and map output signals to it.

For the execution of processing we are choosing DSP-based System-on-Chip device. It offers high computation performance level and excellent performance on image filtering, video surveillance and object recognition. In addition, its power consumption is low which increases its

* Master degree study programme in field: Computer Engineering
Supervisor: Boris Dado, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 591–592.

usability. Depending on the complexity of the algorithm, a DSP can be adequate for housing a complete system. The demandingness of the algorithm can lower the FPS rate of processing. We defined a minimum rate of 3 processed FPS for system to be able to achieve acceptable positive detection rate of monitored objects.

Outputs are triggered, when the algorithm detects a defined action. We are providing several examples of algorithms and describing situations in which an event is triggered:

- Face detection algorithm. Triggered when at least one face is present in the input video for specified length of time.
- Movement detection. Triggered when defined amount of grouped pixels change its attributes.
- Landscape analysis. Triggered when a new object appears in the landscape and stays there for several delayed frames. New objects will become a part of the landscape and are reported only once.

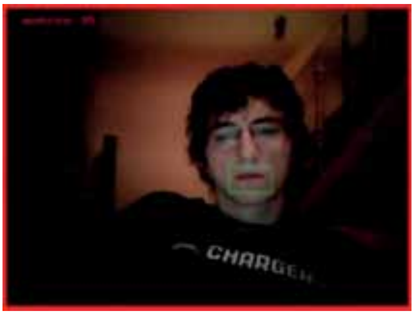


Figure 1. Detection of a mouth.

Events are triggering an output set by the user. Events are also displayed in the interface in a form of graph, which allows users to watch their occurrence. Events are grouped by a day and the figure shows occurrence of events in one month. Events are stored with video or image data and users are able to access them also individually.

We have briefly analysed features allowing high-performance image and video processing on System-on-Chip devices. We have proposed universal system capable of running several image processing algorithms and described hardware and architecture demands of such a system. The system contains an interface which allows users to modify the configuration of algorithms in its runtime, set outputs and view triggered events.

Next possible steps could be implementing more algorithms to process image and video multimedia and create more ways of signalling their results.

References

[1] Kehtarnavaz, N., Gamadia, M. N.: *Real-time Image and Video Processing: From Research to Reality*. Morgan & Claypool, (2006).

[2] Vetterli, M., Kovačevic, J, Goyal, V.: *Foundations of Signal Processing*. Cambridge University Press, (2013).

Component for Editing Text with Graphical Enhancements

Šimon MIKUDA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
simon.mikuda@gmail.com

Extended abstract

This kind of editor can be called Syntax-Directed editor for which Khwaja and Urban defined some features that they should consider in their design [3]: external representation, internal structure type, level of abstractness, target documents, types of errors, error handling, incremental system, execution level, environmental support.

External presentation

There are many programming languages but every is some kind of structural text with basic constructs. Every construct has its own properties and sometimes textual representation doesn't correspond to way it behaves. For instance in *if-else* condition only one part of code is executed, so we can display these parts of code next to each other. Also we can display comments or even syntax errors as floating units.

We have been inspired by Nassi-Schneider diagrams [5] and created possible visual representation of these language constructs. All The usability and performance of display will be tested more when the prototype will be done by writing plug-ins and usability testing. We can create almost every possible layout with these graphical elements: text element, image element, anchor layout, horizontal and vertical layout.

Internal Structure Type

There are lot of possible data structures that can represent text: string, tokens, abstract syntax trees, hyper-graph etc. Every structure can be generated by some grammar by some parser [1] for example tokens can be created by regular grammar, string by finite choice. Since program code is structured text we need grammar that is good for nesting and that is context-free grammar which is generating abstract syntax trees (AST). As parser we have chosen PEG parser and its implementation in LPeg.

[*] Master study programme in field: Information Systems
Supervisor: Dr. Peter Drahoš, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 593–594.

This has main advantage that is quite simple to add new grammars for more languages because Lua is script language that is easy to extend [2]. Last but not least is that we do not need tokenizer [4].

Types of errors and error handling

There are errors that are identified by compiler in compile-time or in run-time by application. For us are more important errors that are identified at compile-time. Specifically state when text doesn't match our provided grammar. There are more approaches how to handle this: (1) disable text modification that are invalid according to grammar, (2) continue editing in presence of errors, (3) automatically fix encountered errors. When we want to continue to edit code when grammar doesn't match we need to keep track of changed graphical elements because they are not homogeneous like characters. We can change text in more graphical elements to state that text doesn't satisfy defined grammar and then we will fix syntax errors. Application must update all changed graphical elements to default text in returned abstract syntax tree (for example when we change number "6523" to two numbers "65 23", we create one element with "23" and original change to "65").

Incremental system

When we are editing text we need to update graphical elements that are visible on computer screen. LPeg parser can only parse everything or nothing it can not utilize results (AST) from old parsing. That's why we created algorithm for comparing old AST with new one. This will give us create, update and delete operations. For this task we created greedy algorithm with O (n) time complexity crawls trees to depth and it is comparing type and text of elements in each node. Since we edit text mostly in one place or block, algorithm compares each nodes from front and when it doesn't match it goes from back.

Conclusions

This paper identifies problems and gives some possible solutions that we have used. Standard process for enriching text with graphical components based on its grammar is: loading suitable grammar for text, parsing text with grammar while constructing AST, create create, update, delete operations utilizing old parsing results, imbue AST nodes with graphical information, create graphical elements and their layout on screen. Applications can display graphical elements and update them from AST. We expect further development of this component and everything is properly documented and revisioned on GitHub.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Grune, D.: *Parsing Techniques: A Practical Guide*. 2nd edn. Springer Publishing Company, Incorporated, 2010.

[2] Ierusalimschy, R.: *Programming in Lua, Second Edition*. Lua.Org, 2006.

[3] Khwaja, A.A., Urban, J.E.: Syntax-directed Editing Environments: Issues and Features. In: *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing: States of the Art and Practice*. SAC '93, New York, NY, USA, ACM, 1993, pp. 230–237.

[4] Medeiros, S., Ierusalimschy, R.: A parsing machine for PEGs. In: *Proceedings of the 2008 symposium on Dynamic languages*. DLS '08, New York, NY, USA, ACM, 2008, pp. 2:1–2:12.

[5] Nassi, I., Shneiderman, B.: Flowchart Techniques for Structured Programming. *SIGPLAN Not.*, 1973, vol. 8, no. 8, pp. 12–26.

Evaluation of Source Code Quality

Jana PODLUCKÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`j.podlucka@gmail.com`

Extended abstract

Since the creation of programming languages, many techniques have been formed to detect bugs in the code automatically. Most of them rely on formal methods and a sophisticated analysis of the code, although these techniques do not necessarily discover the bugs every time. Those are especially the logical errors that can be discovered only during the testing process. In the end, the sooner we are able to discover the bug, the easier it is to repair.

There are many metrics used for code measurement, however in certain cases, their value is a subject to discussion. Among the simplest ones could be the LOC metric which is very easily measured due to being based on counting code lines. McCabe metric is, on the other hand, specialized on code bugs. It calculates the cyclomatic complexity which measures the logical strength of software code in a quantitative way. Shortly said - the higher the complexity of code is, the higher the probability of bug occurrence becomes.

Our effort is focused on the programmers' context in search for bugs. It means that we have to analyze conditions of the code creation, as we expect the important influence of the surrounding environment to the programmers work. For example, just a simple phone call can interrupt concentration and the task context significantly.

Code quality is not only influenced by programmers' knowledge of the programming language but also by the actions that he faces during his work. Mark et al. [2] discuss the influence of outer actions on the programmers‘ work. They have defined the value of diversion as the time needed by a person to return to the context of the work interrupted. With these diversions, several other aspects can be connected, such as stress, for example. The value of interruption is not only based on the type of diversion, but it can be raised or lowered by the personal factors of programmer himself.

Interruptions are also subject of the work of Mark et al. [3] and, according to their researches, the more time person spends on one activity, the higher the probability of being interrupted and also that this diversion is going to be significantly longer. The context of interruption decides whether it is beneficial or not. If it is necessary to switch working spheres, it could have negative influence over the work, while the diversion connected to actual working sphere of somebody else could be considered positive.

Khan et al. [1] studies the mood and its impact on programmers‘ debugging performance. An experiment was created, based on watching several mood-inducing movie clips by programmers

[*] Bachelor degree study programme in field: Informatics
Supervisor: Dušan Zeleník, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

that were tasked to make a debugging test afterwards. In the end, the productivity of programmers was compared.

Context of the source code creation is connected to the programmer and to the conditions of the code creation. As a context of the programmer we could consider following factors.

- Outer environment - represents the environment programmer works in and all the influences he faces there. It can be, for example, the office where he is disturbed by his colleagues. Researches show that a person working in the office is facing various diversions in average of four times per hour.
- Inner environment of the programmer – this means his experiences in the field and his actual mood and state as well. Experience of the programmer can be described as the number of years he spent by programming. However, when programmer works on certain problematic area, his understanding of that field and his ability to get and process new information is important as well.
- Time – explains, when the programmer is creating the code. It can be a part of the day, a day in the week, holidays or various unexpected situations – and we could assume that a code created in these situations would be more probable to contain errors and failures.

Our method to determine the bug probability in source code is based on evaluation of the programmers’ activities in the work. Various combinations of these activities could lead to a rise or a drop in the probability rate. In this part, we will introduce the method to get evaluation of these activities.

At our disposal, there is a data set with the logs of several programmers, working on various software projects. On the basis of these data, we are able to analyze the activity of programmer during the software development. We will select some projects from these and there, we will observe and analyze the bug reports and we will connect them to the programmers’ context during the creation of particular part of code. Based on the influence of various conditions on the final code, we will evaluate these conditions and, in the same time, we will examine various combinations of these conditions and their impact on the code. Final result of our work will be the evaluation in numerical value that will determine the probability of bug appearance.

Our project is focused on determination of quality of a code based on the context of programmer. We examine various factors that influence programmer during his work and their impact on the source code. However there is still possibility to improve this method in the future - we would prefer to focus on examination of programmers' skills and experiences in the process of bug detection. This could be a way to achieve more precise results in practical use.

Acknowledgement: This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] Khan, I. A., Brinkman, W. P., & Hierons, R. M. (2011). Do moods affect programmers’ debug performance?. *Cognition, Technology & Work*, *13*(4), 245-258.

[2] Mark, G., Gudith, D., & Klocke, U. (2008, April). The cost of interrupted work: more speed and stress. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems.* ACM, pp. 107-110.

[3] Mark, G., Gonzalez, V. M., & Harris, J. (2005, April). No task left behind?: examining the nature of fragmented work. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, pp. 321-330.

Three-dimensional Visualization of UML Diagrams

Matej ŠKODA[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`skoda.matej@gmail.com`

Extended abstract

Graphics and graphical systems are used for a long time to express information in more comprehensible way than written text [3]. Due to known fact, that many today's software systems are becoming more and more complex, it is important to use techniques, which are able to represent those complex systems in comprehensible way.

Paul McIntosh studied benefits of the 3D solution compared to traditional approaches in UML diagrams visualization. Main focus of his study was state machine diagrams. X3D-UML displays state diagrams in movable hierarchical layers (see Figure 1 – X3D-UML), in which also filtering can be applied.

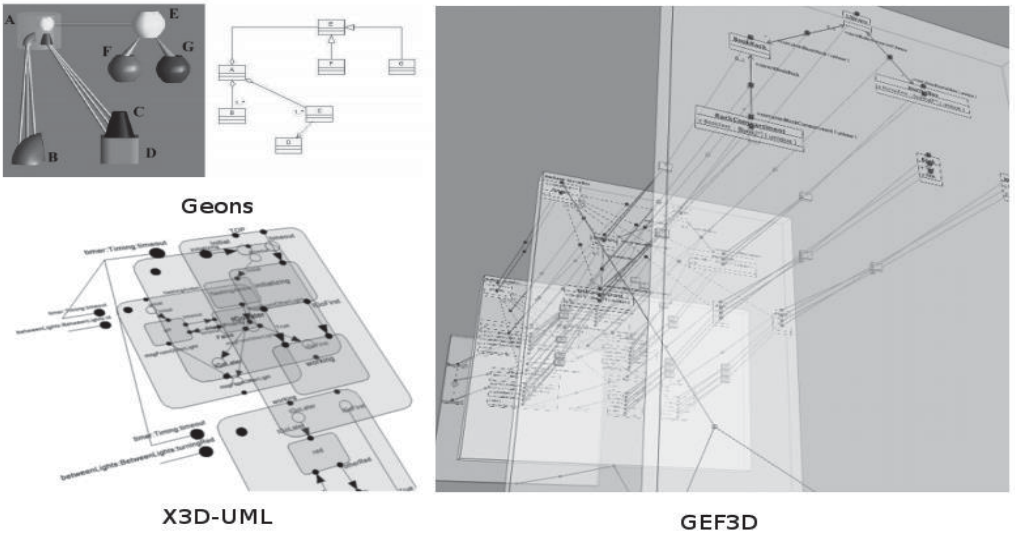
One of main problems will be user-friendly interface for navigation in more complex diagrams in three-dimensional space [3]. This means that possibilities for user interaction and their ability to effectively and naturally control modeling tool will be crucial.

Different concept for displaying UML diagram is usage of geons. Geon diagrams uses various assistance tricks for understanding and memorizing those diagrams. Main objective of this approach is to visualize diagrams by drawing different geometric primitives and shapes. Compared to classical UML diagrams, diagrams of those structures are closer to human perceptions (see Figure 1 – Geons) [5]. By using these shapes, reader is able to create mental map in his mind [1].

GEF3D is a 3D framework based on Eclipse GEF (Graphical editing framework) developed as Eclipse plugin. GEF3D uses so called “multi-editor”. This is result of what they realized – they distinguished between inter-diagram and inter-model view [6].

Main approach of GEF3D is to use third dimension for visualization connections between more 2D diagrams (see Figure 1 – GEF3D). For this purposes are used planes in three-dimensional space, onto which are projected two-dimensional diagrams. This way of visualization can be very useful for model driven development, where the series of models are chained [6]. Another concept that they introduced in field of UML visualization in 3D space is visualization on virtual box [2].

[*] Master degree study programme in field: Software Engineering
Supervisor: Dr. Ivan Polášek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava



Geons

X3D-UML

GEF3D

Figure 1. Various visualizations of UML diagrams in 3D [1, 3, 6].

Our research is oriented to discover new possibilities of using layers to display UML diagrams. Layers contain interconnected classes, objects or activities in activity, sequence and class diagrams [4]. Our scope is to exploit layers for particular components, time and author versions, particular types (GUI, Business services, DB services), patterns and anti-patterns, aspects, etc. Layers allow visualizing alternative or parallel scenarios, or the best, worst and daily use scenarios.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Casey, K., Exton, Ch.: *A Java 3D implementation of a geon based visualisation tool for UML*. Proceedings of the 2nd international conference on Principles and practice of programming in Java, June 16-18, (2003), Kilkenny City, Ireland.

[2] Duske, K.: A *Graphical Editor for the GMF Mapping Model*. (2010). http://gef3d.blogspot.sk/2010/01/graphical-editor-for-gmf-mapping-model.html

[3] McIntosh, P.: *X3D-UML: User-Centred Design. Implementation and Evaluation of 3D UML Using X3D*. PhD. Thesis, RMIT University, (2009).

[4] Polášek, I.: *3D Model for Object Structure Design* (In Slovak). In: Systémová integrace. - ISSN 1210-9479. Vol. 11, No. 2 (2004), pp. 82-89

[5] Ullman, S.: *Aligning pictorial descriptions: An approach to object recognition*, Cognition, Vol. 32 (1989), pp. 193-254.

[6] Von Pilgrim, J., Duske, K.: *Gef3D: a framework for two-, two-and-a-half- and three-dimensional graphical editors*. Proceedings of the 4th ACM symposium on Software visualization, September 16-17, (2008), Ammersee, Germany.

Securing the Last Mile of DNS with DNSSEC

Adam ŠTEVKO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
`adam.stevko@gmail.com`

Extended abstract

The last mile refers to a network between the client and recursive name server. The querying client has no easy way of proving the validity of the received response. This is slowly changing due to higher deployment rate of DNSSEC. The recursive name server will set AD bit [1] to notify client if the answer was validated. The problem here is that the communication between the client and recursive name server is unauthenticated.

Proposal. The following steps are required in securing the last mile of DNS:

- Key distribution and identity verification of the recursive name server the client will be using for domain name resolution. This is the most crucial part.
- Key exchange for shared secret material between the client and the recursive name server.
- Use of shared secret to authenticate every DNS transaction.

Identity authentication. The most difficult part of securing the last mile is to verify the identity of the recursive name server. If the client wants to establish trust with the recursive name server, it needs to have a secret known only to itself and the name server. This secret is obtained via key exchange mechanism. However, the client needs to know the correct parameters of the key exchange. DNSSEC will be used for identity authentication. DNSSEC provides end-to-end data integrity thanks to digital signatures which can be trusted once the signature is validated. By using this property of DNSSEC, we can use DNS as a secure key distribution channel. However, this adds a requirement on DNSSEC being deployed in the recursive name server's infrastructure.

Key exchange. Key exchange is performed by using Diffie-Hellman mode of TKEY resource record. This Internet standard describes methods for secret key establishment in DNS. Once the client starts the key exchange with the name server, shared secret will be established on both sides. The newly established shared secret will be used for TSIG.

[*] Master study programme in field: Computer Engineering
Supervisor: Assoc. Professor Ladislav Hudec, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2013, Bratislava, April 23, 2013, pp. 599–600.

Transaction authentication. TSIG key is known to both client and the recursive name server and every DNS transaction between them is signed by this key. The client will be using the TSIG key until it expires and will be notified by the name server and repeat the whole bootstrapping process.

Drawbacks. This solution adds some latency to the name resolution process and hurts it's performance. This is caused by the recursive name server's lookup of the correct TSIG key in its key store. As there can be thousands of TSIG keys stores in server's key store, the lookup will take some time. However, this can be mitigated by using effective algorithms for storing keys.

Evaluation. The prototype consists of two parts: client resolver and a DNS server. The evaluation phase covers measuring latency of all relevant actions and comparing numbers with traditional name resolution. Those actions are:

- Validation phase.
- Shared secret establishment.
- TSIG transactions and key lookup.

If the latency is high, we will need to modify the prototype to do a better caching and further optimize key lookup time.

Conclusion. We have presented the concept of solving the last mile problem of DNS with DNSSEC. The idea was to use DNS as a distribution channel for keying material and DNSSEC for validating purposes. Key exchange and DNS transaction signatures are not new, so the final solution is much easier to adopt and implement.

References

[1] Wellington, B., Gudmundsson, O.: Redefinition of DNS Authenticated Data (AD) bit. RFC 3655 (Proposed Standard), 2003.

Textual Representation of Data Models in Identity Management

Tomáš ZBOJA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xzboja@fiit.stuba.sk

Extended abstract

An identity management system manages identities in a concrete installation of this system. Because various organizations install such system, the respective software product provides only core entities and attributes for the system, which are common for all of the installations. Along that, each installation requires its own entities or extending attributes. For this purpose there is a schema that describes the entities provided by the core system as well as those in customers' extensions. From this schema, an executable code is generated later. Another very important factor is that data schemas in identity management are very extensive, so it is very requisite to have light simple schema language.

Only one paper related to this paper's topic was found [1]. It, however, does not solve a problem of better format for schema representation.

In presented work various languages were analyzed, on how well they fulfill set requirements. The analyzed languages are: ASN.1 [2], JSON Schema[1], OWL, UML – in three different representations[2] [3]; XML Schema and YANG [3]. Among the main requirements are: ease of work, efficiency, textual representation, extensibility and processing information. Every language has more or less serious flaws in fulfilling the requirements. Some languages have serious problems because even though they are schema definition languages, they were created for very concrete field, which, however, is not identity management. Therefore they have features, which are useful for their field, and do not fulfill requirements set for the identity management field.

Because as one of the languages that best fit the criteria is UML, this language was chosen as the main inspiration for the proposed language. The proposed language could be described as some attempt of textual UML, combined with some elements of XML Schema, and added features to fit certain criteria of identity management. XML is from the analyzed languages the second most important inspiration, especially with its namespace representation. Syntax for textual UML was inspired in language USE, because it provides minimalistic and easy-to-read syntax.

In this paragraph the most important requirements are listed, and how the proposed language satisfies them. Namespaces are a must in supporting extensibility. They are supported via dot rep-

* Master degree study programme in field: Software Engineering
Supervisor: Dr. Pavol Mederly, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] http://www.json-schema.org/latest/json-schema-core.html

[2] http://www.db.informatik.uni-bremen.de/projects/use/use-documentation.pdf

[3] http://tools.ietf.org/html/rfc6020

resentation in class' name, similarly as in XML Schema. In future is planned an extension without the need to inherit class' attributes, but directly "inject" new attributes into an existing class. Information for processing, which are information on how to process concrete elements or data, but at the same time are not data about model itself, are supported via at-notation (`@info`). Efficiency, which is also very important and related to readability and maintainability is at good level. In the first measurements, which are however not perfectly representative, but give a rough idea, a drop in number of lines was about 40%, and in number of characters was nearly 50%, in comparison with the original XML Schema file.

Even though the proposed language's first intention is not to provide complete textual representation for UML, later the idea of textual UML seems to be a right step. Therefore this language has created also artifacts which are not applicable or usual for schema representation, but could possibly one day help it to start its textual-UML purpose. At this time the only supported diagram is the class diagram. However, this has to be noted in the source code, as possible future use might allow also other types of diagrams of UML language. Some yet supported features for use as UML are the following:

- methods of classes,
- interfaces,
- various relations between classes,
- visibilities.

In the Figure 1 is provided a simple example of UML model and corresponding code.

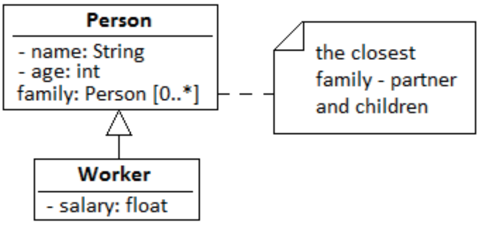


Figure 1. Simple UML example.

```
class Person {
  attributes {
    - name : String;
    - age: int;
    family: Person [0..*] {
      @{the closes family -
       partner and children}@
    };
  }
}
class Worker < Person {
  attributes {
    - salary: float;
  }
}
```

The proposed language is going to be tested in real identity management project. If it will gain success, it can become not only an academic language, but also a practical language for real use. In addition to this, the language can serve purposes for UML textual representation.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/1221/12.

References

[1] Cao, Y., Yang, L.: A Generalized Identity Specification Language Based on XML Schema, *DIM '11 Proceedings of the 7th ACM workshop on Digital Identity management*, ACM, ISBN 978-1-4503-1006-2, New York, (2011), pp. 3-12.

[2] Cassel, L. N., Augusting, R. H., Richard H.: Computer Networks and Open Systems: An Application Development Perspective, *Jones & Bartlett Learning*, ISBN 978-0763711221, (2000).

[3] Fowler, M.: UML Distilled, *Third Edition, A brief guide to the standard object modeling language*, Addison-Welsey, (2004), ISBN 0-321-19368-7.

Accompanying Events

TP Cup – The Best Student Team Competition Showcase at IIT.SRC 2014

Mária BIELIKOVÁ*

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
maria.bielikova@stuba.sk

Abstract. Best student team competition TP-CUP is organized sixth time this year. The competition is aimed at excellence in development information technologies solution within two semester long team project module in master degree programmes. This year 13 student teams presented in IIT.SRC 2014 showcase their projects. Key concepts of their projects are included in the following sections of the proceedings.

1 Background of the Competition

Team projects play an important role in the education of engineers. Team projects have a long tradition in informatics and information technologies study programmes at our university. Module firstly named *Team project* was introduced in the academic year 1997/1998 in software engineering and in subsequent years it was adopted as compulsory module for all master degree students. Its intake is each year 25-30 teams of 5-7 students in all study programmes. The main objective is to give students a hands-on experience with different aspects of working in team on a relative large task.

In designing a team project as a part of a curriculum, we considered several aspects or different alternatives to particular issues such as team formation, team communication methods, team assessment, problem assignment, development process and team supervision. Our experience with such projects is that a satisfiable solution (in terms of the team project objectives, i.e. experience with different aspects of working in team on a large problem) requires time longer than one term, so we designed our team project as two semester module. Supervisors who are available (either academic staff or an industry partner) determine problems being solved. Teams consist of 5-7 students. They are created under our active control. Our criteria aim at balancing differing specific knowledge of team members and different experience in various team roles. We also respect the students' preferences to some extent (a student can specify one student to become a member of the same team).

We let teams bid for problems proposed by supervisors. A competition between teams is established and students have opportunity to exercise writing and presenting the bid. The students bid with their knowledge, skills and achievements related to the selected problem, and with

[*] Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

a preliminary sketch of solution based on the open question-answer session with a customer (mostly a supervisor).

Although the quality of the final result is an important measure of a success of a team, we markedly concentrate on the process applied. Through the years of providing Team project module we adopted the development process with at least two iterations. Six years ago we have started with agile developments methods. First four years several teams employed agile development methods each year. From academic year 2012/2013 all teams follow agile development according selected utilized agile methodology. This year all teams work according the Scrum methodology.

The amount of freedom and supervision should be balanced in order to create a true learning experience for students. To simulate the reality, students should have a considerable amount of freedom. On the other hand, since students usually have no or just little project experience, some amount of supervision, monitoring and guidance is needed to ensure sufficient progress and a successful result. In order to reach balance between freedom of students and supervision we specify in advance certain requirements on the content of documentation to be produced. Students have to prepare and follow a detailed project plan. We prescribe certain parts of the project plan, such as list of activities, milestones, dependencies, and responsibilities according to established team process. Students are free to define the activities that are necessary to successful accomplishing of the project. We accompany the Team project by lectures on project management, teamwork, and quality assurance.

2 Stages of the Best Student Team Competition

In order to emphasize excellence of the students' teams we established the Best Team Competition called TP Cup in academic year 2008/2009. The competition is aimed at excellence in development information technologies solution within our two semester long team project module in master degree programmes.

The competition has several stages. It starts with an application in the middle of the first semester.

- First stage finishes by the end of first semester when the teams submit interim report. We filter out teams which do not fulfil basic criteria on quality of work performed.
- Second stage culminates in the middle of second semester when students submit key concepts in form of two page report into IIT.SRC proceedings and present their projects in the TP Cup showcase organized as a part of the IIT.SRC conference. This year 13 students' teams presented in form of showcase their projects at the IIT.SRC 2014.
- Third stage presents finalizing the projects. It ends by our grand finals where board of judges consisting experts from industry selects the winner team which lands the challenge cup – "*Best FIIT Student Team of the Year*".

More information about TP Cup can be found on the Web:
`http://www.fiit.stuba.sk/tp-cup/`

Three-dimensional UML

Gabriela BRNDIAROVÁ[*], Ivan MARTOŠ[†], Andrej ŠTAJER*, Matej ŠTETIAR*,
Erik ŠUTA†, Andrej VALKO*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
gamatepi@gmail.com

CASE tools are ordinarily used for modelling in two-dimensional space. This concept constrains designers and software architects to look on software on limited level of abstractions. The third dimension can create an illusion of space and create more attractive environment. CASE tools should support work with model including create, read, edit and delete principles [1], [2].

UML sequence diagram is used for modelling behaviour of software system. Sequence diagrams are composed of lifelines, messages and fragments. Every element used in this type of UML diagram has its typical graphic representation. In this paper, we focus on three-dimensional view of sequential diagram modelling. Our approach is based on multiple layers located on scene in different depths.

Project is implemented using Ogre3D framework in programming language C++. Designed tool provides functionality for drawing three-dimensional scene with layers containing sequence diagram elements. These elements can be added or removed from scene. It is also possible to save and load the scene. Tool architecture is visible in Figure 1. Each module provides complex functionality. *Application management* module is used for user interaction with software, *Core* module is used to provide functionality for application management, *Graphics* module is used to interaction with Ogre3D framework, *Data structure* module is used to store UML properties and is based on UML meta-model and *Serialization* module is used for saving and loading of the model.

To provide better control of the three-dimensional space on regular devices we provide set of controls. Using these controls user defines layer which he wants to work with. This approach is called two and half-dimensional space, too [3]. Each layer can contain lifelines, messages and fragments. Messages can be also placed between lifelines located on different layers. Lifelines are automatically aligned to the left and messages to the top of the layer. It is possible to insert message/lifeline between two existing messages/lifelines. Then, the whole scene is reorganized to maintain integrity of the model. Alignment of the elements is modified after act of deleting, too.

The main purpose of the tool is creation of multiple layer diagrams. We believe that this approach will provide better usage of the diagrams. In these diagrams, each layer can represent:

- use case scenario,
- version of scenario,
- version of the author in real-time collaboration,

[*] Master degree study programme in fields: Information Systems
[†] Master degree study programme in fields: Software Engineering
Supervisor: Dr. Ivan Polášek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

- object types (e.g., GUI, logic, database),
- aspects and objects,
- optimistic, pessimistic and daily use scenarios,
- alternative and parallel scenarios,
- patterns and anti-patterns,
- specification of the abstract parts of scenario.

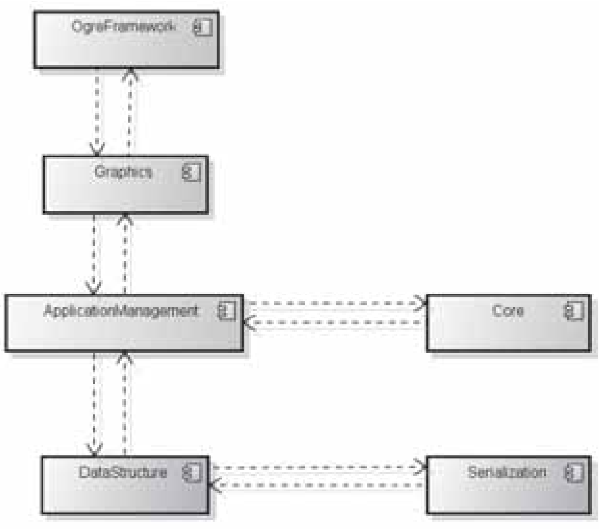


Figure 1. Architecture of 3D-UML tool.

This project is proof of concept of three-dimensional view of UML modelling. Project idea will be tested by creating a simple prototype. Prototype can be used for creating 3D UML diagrams. If we can prove, that three-dimensional CASE tool can be fully used for software development and modelling, there is a great perspective for improving development and design process by providing new diagram dimension – depth. As we mentioned, in section **Error! Reference source not found.**, we believe that this approach can extend basic UML. This extension should bring better and faster understanding of diagrams by their division to use cases or functionality, multiple levels of abstraction, collaboration in model creation, etc.

References

[1] Paul McIntosh, Margaret Hamilton, Ron van Schyndel: X3D-UML: enabling advanced UML visualisation through X3D. Proceeding, Web3D '05 Proceedings of the tenth international conference on 3D Web technology, 2005.

[2] Polášek, I. 3D Model for Object Structure Design. In: Systémová integrace. – ISSN 1210-9479. Vol. 11, No. 2 (2004), pp. 82-89

[3] Von Pilgrim, J., Duske, K.: Gef3D: a framework for two-, two and a half- and three-dimensional graphical editors. Proceedings of the 4th ACM symposium on Software visualization, September 16-17, (2008), Ammersee, Germany.

EIVA: Efficient Interactive Video Annotation

Jaroslav BUCKO*, Matej ČÁRSKY*, Peter JURKOVIČ†, Ján KEBÍSEK*, Marián KURUC†,
Viktor MARUNA†, Máté VANGEL†

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
timovy-projekt-14-2013-2014@googlegroups.com

Nowadays, many of the problems, otherwise unresolvable because of the need of human interaction, can be solved by crowdsourcing. One of the most common problems that require human interaction is the missing solution for efficient search in multimedia. It is nearly impossible for computer to correctly identify the content of a picture, video or sound. Therefore, the human interaction is required to determine the correct description of the data provided. Unfortunately people do not want to spend their free time on a task from which they gain no benefit. The difficult part is to design a way for people to do it willingly and in the best case, in large numbers.

In this project, we would like to resolve the problem of intelligent tagging of a video, with the help of a mobile game and gamification. We provide users with a way to entertain themselves and while doing so, the game gathers much needed information. One of the most difficult issues is to design a game that is really catchy, otherwise we will end up with a low user base and the speed and amount of the gained data will be much lower [1, 2]. The principle of gaining data faster by crowdsourcing might not apply in this case.

The designed game is based on a popular board game Dixit. Main goal of our game is to guess a card, submitted by a storyteller with the help of a hint, which he provides. Hint can differ among several types, whether it is a text, emotion or a meme picture. Game contains a lot of unique cards, which are animated, funny and beautifully designed. The dataset is gathered from entertainment websites, like 9GAG, that already have the data sorted, by popularity. This ensures that the provided cards are not boring and we do not need to sort them ourselves. We will gain two types of collected data: emotions gathered from meme pictures and plain text metadata gathered from clues in the game. All the clues from various languages will be translated to English and the quality of these metadata will be ensured by different preprocessing techniques and approaches of term weighting.

We are also trying to publish the game on as many platforms as possible, ranging from mobile platforms, such as iOS, Android and Windows Phone, to desktop platforms - Windows, Mac and Linux. The more platforms the game supports, the larger amount of players can be obtained - we want the user base to be as large as possible, so we can gather a huge amount of the data [3].

[*] Master degree study programme in field: Software Engineering

[†] Master degree study programme in field: Information Systems
Supervisor: Dr. Jakub Šimko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Additionally, players gain reward as they progress in the game, which they can later use in acquiring additional cards. They can also acquaint their friends about their success in the game, by sharing their score on social networks - Facebook, Twitter and others.

There is an interest in using the data we have collected in psychology, more precisely in research of human emotions and what images might invoke such emotions. There are many possible uses of the data we have collected, but this is only one of the possible ways.

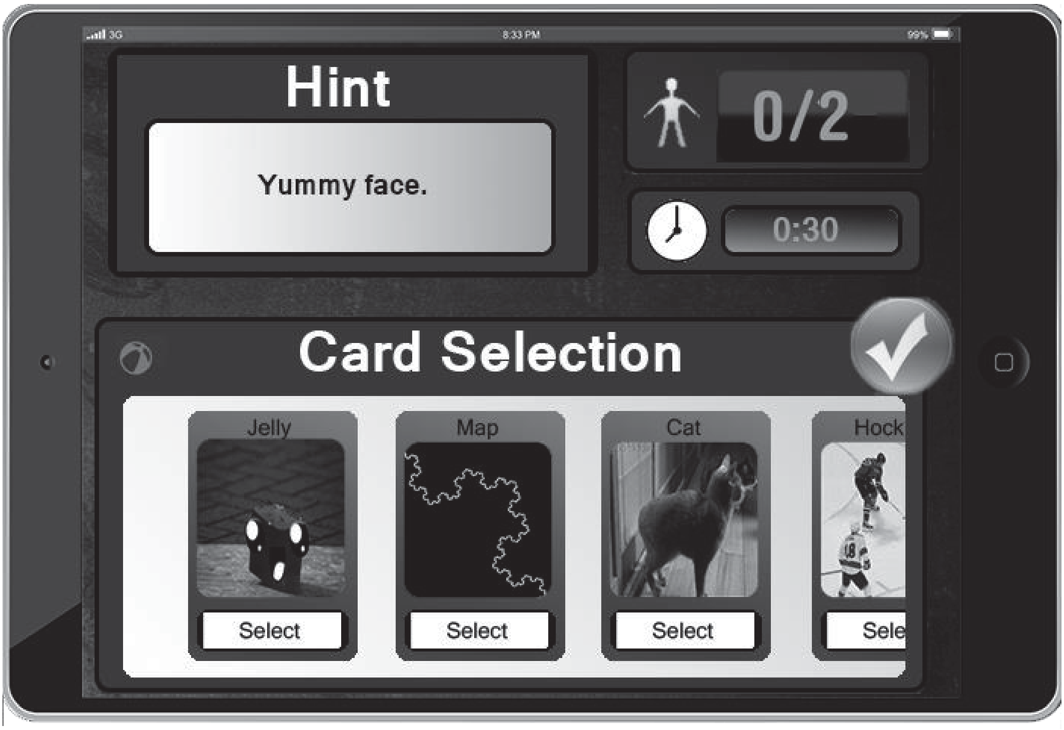


Figure 1. A sample from the video game EIVA.

References

[1] Hunicke, R., LeBlanc, M., Zubek, R.: MDA: A formal approach to game design and game research. In: *Proceedings of the AAAI Workshop on Challenges in Game AI*, 2004, pp. 1-5.

[2] Schell, J.: *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[3] Crawford, C.: *The Art of Computer Game Design*. Osborne/McGraw-Hill, Berkeley, CA, USA, 1984.

A Mobile Application for Quick Information Retrieval Associated with a Building

Lukáš CÁDER[*], Martin DUŠEK[*], Jaroslav DZURILLA[*], Roland GÁŠPÁR[*], Martin LONDÁK[*], Michal ŠEVČÍK[*], Matej TOMA[†]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova2, 842 16 Bratislava, Slovakia
timovy-projekt-6-1314@googlegroups.com

When a man comes to a new building, he feels disoriented and do not know anything about area. The main idea of our project is to ease these unpleasant effects of being in unfamiliar indoor space or its close surrounding. To do so, we offer all important information about critical building and area from different information sources in a form of mobile application. Its key property is to offer the needed information as easy and as quickly as possible using innovative searching methods. Such solution is portable to different types of buildings from different domains e.g. schools, business centers, shopping centers, hospitals and so on.

There are several applications which deal with part of this problem – with indoor navigation but there are not as many complex solutions when it comes to combining this problem with the problem of getting important information really fast. The one system that is very close in ideas to ours is the project called “School of the Future” [1]. Goal of this project is to help students navigate inside campus and school buildings and to get important information with the help of augmented reality in their smartphones. According to the authors of the paper, this system is also suitable for other buildings from different domains just like our application.

To test these ideas, our predecessors decided to develop an application for our faculty – FIIT STU as web page and OS Android version. They named it Virtual FIIT. In faculty, the majority of users are students. According to their needs, the application’s main function is to offer a possibility to look up their lessons (time schedule), teachers (contact, office location and schedule) and interactive maps of all floors in the building. Application also consists of a barcode scanner for QR codes, which are located on every room of our faculty. QR codes contain information about the person within the room. The application also contains information about objects in nearby area such as actual food menus of different canteens and bus departures from the nearest bus stops.

We have taken over this ongoing project. It is programmed in JavaScript, HTML5 and CSS3, what makes it also a webpage. When deployed by PhoneGap technology, it is a native application for Android OS, but it can be deployed for other operation systems, too. Application architecture design minimizes time of source code changes in case of reimplementation for other building from other domain. To enrich this multiplatform application we have corrected all of the interactive maps, added the map of the nearby area with bus stops and canteens, remade the whole design of

[*] Master degree study programme in field: Information Systems
[†] Master degree study programme in field: Software Engineering
Supervisor: Dr. Alena Kovárová, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

the application (see Figure 1), added new feature in a form of RSS stream, implemented other various features and fixed majority of bugs[‡].

The most important features from user's point of view which we had implemented with the highest priority are: periodical storage of data, so the users can access any information without internet access, and complex search function. These two features are of the greatest importance to our users and give our application unique value. Since the quick information searching is leading property of our application, our plan is to empower it even more so it will be able to process requests in natural language by detecting key words and pairing them with answers from our database. This pairing will be calculated from graph with weighted vertices and edges, which represent the weights of keywords and weights of relations between them. This is the main unique feature which will differentiate our application from any other.

The future work in which we see potential is use of Bluetooth chips and Bluetooth Low Energy technology as a tool for indoor navigation. This technology allows smartphones to determine their orientation and distance from the Bluetooth chip transmitter [2]. Another idea for future work is to modify interface of our application to support Google Glass platform.

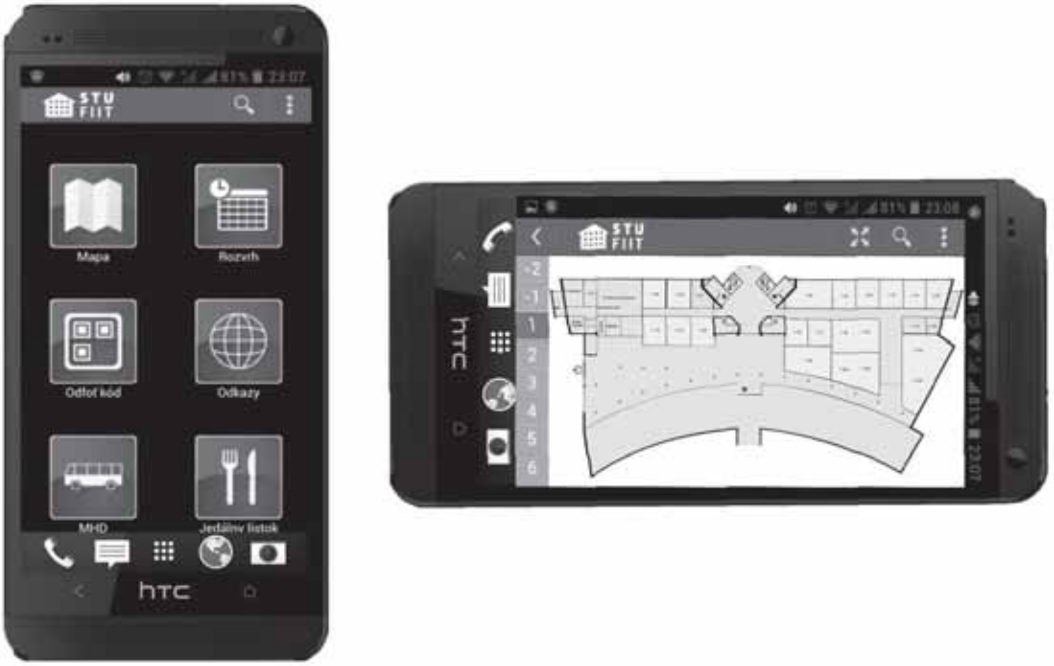


Figure 1.Virtual FIIT mobile application, main screen (left) and map of the first floor (right).

References

[1] Vert, S., Vasiu, R.: School of the Future: Using Augmented Reality for Contextual Information and Navigation in Academic Buildings. In: *Proceedings of the 2012 IEEE 12th International Conference on Advanced Learning Technologies (ICALT '12)*, IEEE Computer Society, (2012), pp. 728-729.

[2] Want, R., Schilit, B., Laskowski, D.: Bluetooth LE Finds Its Niche. *IEEE Pervasive Computing 12*, (2013), vol.12, no. 4, pp. 12-16.

[‡] Virtual FIIT, https://play.google.com/store/apps/details?id=sk.stuba.fiit.virtfiit&hl=sk

Gaze Tracking for Usability Testing of Dynamic Web Applications

Dominika Červeňová, Jakub Daráž, Lukáš Gregorovič
Martin Janík, Róbert Kocian, Michal Mészáros
Kristína Mišíková*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
tp1314team1@gmail.com

Usability is one of the most important features of every application. Application developers need to provide a graphic user interface that is good, intuitive and easy to understand, so users can find and use all functions and features effectively.

There are many metrics to measure usability. In usability experience testing combination of them is usually used. We can gain information by watching users with the help of a common hardware, e.g. track user mouse clicks, scan mimics with a webcam, etc. Although these types of gained data are very useful, they still do not provide enough information. For instance from mouse clicks alone, we are not able to differentiate whether users do not interact with a website element because they do not find it relevant or because it is not visible enough. There is also a possibility of an interaction with users during testing to gain more complete information. However, this may cause negative influence on testing results, because our interaction and interruptions may influence users' behavior.

We can address these problems by tracking users' eyes [1]. Gaze tracking has become very popular among researchers recently and there are many surveys, focusing on this topic, such as work of Wass et.al. [2]. In this document, we present a platform that enables a researcher to conduct usability experiments, cooperating with eye tracking devices. The hardware, we are working with, is provided by a company called Tobii Technology[1], which is the world leader in eye tracking and gaze interaction. Their devices produce useful data to a researcher for UX testing. While participants look at the screen, they gather gaze data including validity code, timestamp, eye position, relative eye position, 3D gaze point, 2D gaze point and a pupil diameter.

Tobii Technology company also provides software that cooperates with eye tracking devices Tobii Studio, but this application has a few disadvantages. For example Tobii Studio does not provide a simultaneous remote data collection.

Our project's main focus is on providing infrastructure for dynamic web applications testing with many participants at once, that would require no interference with the tested applications. Our infrastructure displayed in Figure 1, consists of three main parts.

[*] Master study programme in field: Software Engineering / Information Systems
Supervisor: Róbert Móro, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] www.tobii.com

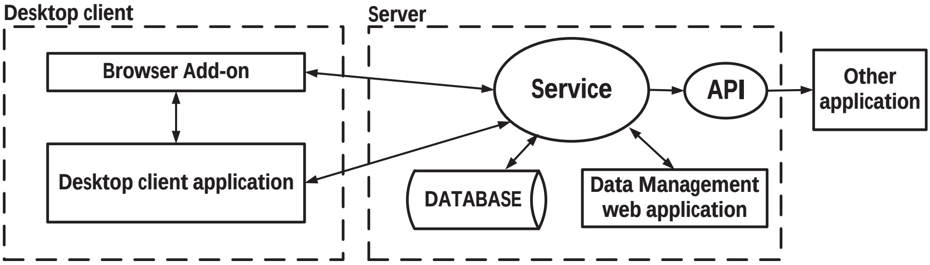


Figure 1. A model of infrastructure for testing of dynamic web applications.

The first one - desktop client provides communication with eye tracking device, preprocesses the gained data and sends them for further processing. As the only part, that experiment participants have to interact with, except for the tested application, it enables to sign in and calibrate eye tracking hardware. The second part is a browser add-on, which for now supports just Mozilla Firefox browser. This add-on enables a researcher - our key user to select his “areas of interest” - specific elements of his website that he wishes to track users’ eyes on. Through add-on we gain information about what element is user currently looking at, hovering mouse above or clicking on. Having these information we are able to create for instance automatic tags, depending on users’ interactions, that are useful for the researcher later for data analysis. Data collection and evaluation is provided by the third, server layer, database, service and data management web application. Through the web application researchers can set up and configure an experiment. They can create a project and sessions within it. Each session represents one experiment, where researcher adds participants - users of the tested application. The web application also provides statistics, data visualization and automated real-time data annotation.

Our client application enables to add various types of devices. For example eye-tracking device produced by other company then Tobii or a simulator if there is no eye-tracker available. The behavior of simulator is similar to the eye tracking hardware. It was primarily made to help us develop and test our projects’ core items, therefore we focused on the 2D gaze point values, so we could simulate participants’ gaze. The simulation is based on using normalized mouse coordinates instead of 2D gaze point values, so the mouse moves represent moves of participants’ eyes.

Moreover, we provide an API, that enables other researchers, e.g. at our faculty, to use our infrastructure to conduct usability experiments. They can use gained data and our simulator not only for developing new applications related to usability experience testing. With our API it will be also possible for instance to develop adaptive applications using users’ gaze. Our application is able to track many users at once, the only restriction is the number of eye tracking hardware. We plan to deploy our solution to the UX Class at our faculty, where it will be used by 20 devices at the same time.

References

[1] Poole, A., Ball, L.J.: *Eye tracking in HCI and usability research.* Encyclopedia of Human-Computer Interaction, 2006.

[2] Wass, S.V., Smith, T.J., Johnson, M.H.: *Parsing eye-tracking data of variable quality to provide accurate fixation duration estimates in infants and adults.* Behavior Research Methods, 2013.

Data Visualisation in Augmented Reality

Duško DOGANDŽIĆ*, Dávid DURČÁK*, Ján HANDZUŠ*, Patrik HLAVÁČ*,
Marek JAKAB*, Matej MARCOŇÁK*, Daniel SOÓS*, Martina TRÉGEROVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
teamtp05@gmail.com

The key to success is to be able to transform information into knowledge. These days, we create a big amount of data in different areas. Simple viewing of these data in the form of database may not be only a hard task, but time consuming, and, in most cases, it will not lead to any results. Therefore, other form of viewing these data must be chosen in order to get useful information, which can lead to possible knowledge.

Possibly the best approach to start with is the usage of data visualisation, where seeing data in the form of graph will help us to understand these data much more easier. Also, this technique creates a chance of getting a new perspective in terms of data relation. Visibility of some unknown characteristics through the graph can be obtainable. Data visualisation gives us a basic view of analyzed data and, also, helps us in finding usable information, or it may give us better orientation in selected data. The more easier and userfriendly are these data provided to the user in the form of a graph, the more knowledge can be obtained in significantly less time. Apart from the functionality, on the other hand, the design of the graph must be considered to provide the expected results as well.

The main focus of our research is to provide better control of data shown as a graph set in real environment. Eventually, these changes will substract unnecessary attention from data manipulation and improve information extraction, as we can fully focus our attention on the specific problem. Consequently, our goal is to free the user from using standard input and output devices such as keyboard, mouse or monitor and offer them the possibility of controlling the graphical form of data through body movements and gestures, respectively.

Given the fact that people are more used to the real environment rather than virtual reality, it persuades us to utilize this in further development. To amplify the usability effect, we intend to keep the data drawn in real environment and create an augmented reality application, whilst taking into consideration the simple physical laws and constraints. Such an example of this is to make sure that drawn data will not interfere with real objects, which will be found around the rendered graph, so the user gains the feeling that the visualised object really exists.

In the recent past, several approaches were made to complement or even replace standard devices, though none of them solved this functionality for a similar application that is proposed in our paper. For example, in [3], infrared camera images were used in order to detect the user's hand and fingertips to create an augmented desk interface. Hand detection through image segmentation performed on a Kinect device is proposed in [2] and [1] provides a hand tracking algorithm optimized for the same device, too.

[*] Master study programme in field: Software Engineering / Information Systems
Supervisor: Dr. Peter Kapec, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 615–616.

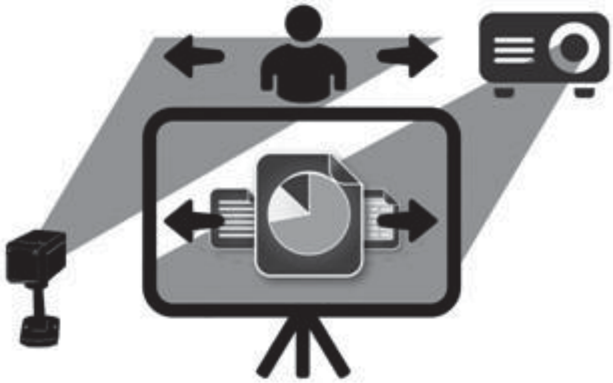


Figure 1. Scene set to make our application run.

To create such a project, there is a need of using a specific hardware. To be able to set the graph in real environment, we need to apply a projector through a special glass, where the user has the possibility to see the desired graph, together with the real objects behind it. To attain information about objects and recreate the scene behind the glass, we use a Kinect sensor, which provides us the depth information. Furthermore, this information can be used in cooperation with the objects in the scene, so the user is able to move the graph, and, for instance, place it on the table.

Afterwards, the same Kinect sensor or another one can be used for user input, too. In this case, the sensor captures the movement of the user. Implicitly, the aim of the capturing is to manipulate with the graph thanks to body movement, similarly to [1]. We provide graph control based on head movement with normal camera as well, if there is no Kinect sensor available at the very moment.

A possible usage scenario is to determine constraints such as the physical wall behind the special glass, when the user also has a choice of setting up the background to the graph. Another case is a hand motion based mouse controller that supports the projection. In addition, a feature that offers speech recognition is available too. Speech commands help the user to make even more straightforward actions towards the graph (e.g. select a subgraph of it).

Figure 1 shows the scene set we use, with glass in front of the user, the Kinect sensor facing towards the user and the projector behind the user projecting the 3D graph.

Acknowledgement: This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VEGA 1/0625/14.

References

[1] Frati, V., Prattichizzo, D.: Using Kinect for hand tracking and rendering in wearable haptics. In: *World Haptics Conference*, 2011 IEEE, (2011), pp. 317–321.

[2] Raheja, J., Chaudhary, A., Singal, K.: Tracking of fingertips and centers of palm using Kinect. In: *Computational Intelligence, Modelling and Simulation*, 2011 Third International Conference, (2011), pp. 248-252.

[3] Sato, Y., Kobayashi, Y., Koike, H.: Fast tracking of hands and fingertips in infrared images for augmented desk interface. In: *Automatic Face and Gesture Recognition*, Fourth IEEE International Conference, (2000), pp. 462-467.

Automated Acquisition and Standardization of Citations

Michael GLOGER[†], Tomáš JÁNOŠÍK[†], Daniel KĹČ[†], Šimon KOMPAS*,
Rastislav KOSTRAB*, Stanislav KUBICA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
tp-1314-10@googlegroups.com

Research institutes are mainly financed through grants. To gain funds for their research, it is important to support the process of research and development. Evidence of the research process is then made according to the number of citations to the publications published under corresponding institutes.

According to the law, for every institute in Slovak Republic it is mandatory to register their publications in libraries, which then use international standards for processing and recording them. Unfortunately there is no control over the standards compliance, as well as there is no National Authority Database. This leads to several ambiguities in data evidence.

Also some errors may be made during the process of registering the publications. For example, one of those errors may be incorrect character entry when there is no corresponding character in the national character set. Researchers must take into account these errors when they are trying to find citations to their publications in external sources. This leads to challenging and difficult process of manual citation search.

Because of the problems described above, we have decided to build an information system that could relief the researchers from this difficult work. The result is an information system with high level of automation of the process of harvesting, evidence gathering and processing citations of the publications. This may form infrastructure for the future extension of the system, which could bring us various perspectives on research rank.

Our system is based on previous work of the students under our supervisor. We have used their knowledge and experience to further improve the developed system. With respect to the standards and maintainability goal, we have designed a new modular architecture. It is designed to make the addition of new components or modification of the existing components easier, because particular components can change frequently. It reflects the structure and principles of the architectural style blackboard. Particular components communicate through the data stored in the database. We are hence allowed to achieve a high level of maintainability, although it places requirement on the common data structure.

We have developed various forms of data acquisition. Besides the bibliographical formats such as MARC21 or UNIMARC, the input may be formatted in XML, for example the exports

[†] Master degree study programme in field: Software Engineering, *Information Systems
Supervisor: Dr. Nadežda Andrejčíková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] CREPČ - Centrálny register publikačnej činnosti a Centrálny register umeleckej činnosti (http://cms.crepc.sk/).

from CREPČ[1] (central register for publication activity). Our system can also communicate with libraries directly through the standards OAI PMH or Z39.50. In the case of direct communication through the above mentioned standards, there are several formats we can process (MARC XML, ISO 2709 or line MARC). We have designed the system to accept any of these formats. It is possible to extend the system to add new formats in the future.

Various formats have data organised in different forms and contain different data fields. Our data model has been carefully designed to represent any data field found in the used formats. We have also modelled the relations between the entities, which are required to be very generic. Because the data model can change often, we have employed one common Object relational mapping. For example data model can change when support for new data format is implemented. Thanks to this property we have achieved the requirement of system flexibility.

Our goal was also the improvement of the identification process and comparison algorithms. We have discussed several special cases, which are tough to identify automatically. Such cases include different authors with exactly same name or cases, in which misspelling the name of particular author exactly matches the name of another author.

Several methods and techniques have been studied [1, 2] to decide which one would be the most effective in this field of study. We have employed Jaro-Winkler algorithm to be able to compare the names recorded with errors. This algorithm returns the result with high fineness. In entity matching, we take into account several attributes of the entity and combine them into final result with the cosine similarity method. The entity matching is made up of these algorithms and specially designed rules, which were employed thanks to experiences and prior knowledge of this field of study.

In order to make the identification process efficient, parallelization of entity matching is planned in the future. We are planning to develop web service, through which the entity comparison would be provided for other users. Also several different methods of comparison would be available.

In our system three user roles have been discovered. At first there is the role of the administrator of the system, who can authenticate the authors, and has the right to change the data contained in our system. Secondly there is the author, who can look up citations to his publications and who can help us with manual entity matching. Lastly, there is the general user, who can use information we offer to find interesting information about the authors, publications and relations between them.

We have designed and implemented the information system, which can offer valuable information to its users. It is build thanks to knowledge of the fields of bibliography, effective algorithms and several other standards. To provide information with even better quality, we are improving our system by testing the implemented methods and through new features, which can be added in the future.

Our system can serve as an infrastructure base for the research rank in the future. Normalized data model has been tested and prepared for the ranking methods development. We have implemented an analytical tool for examining some basic relations between entities, although several other relations are still to be examined.

References

[1] Carvalho, A. P. DE et al.: Incremental Unsupervised Name Disambiguation in Cleaned Digital Libraries. In: *Journal of Information and Data Management,* ISSN 2178-7107, (2011), vol. 2, no. 3, p. 289-304.

[2] Hernandez, M. A.: The Merge/Purge Problem for Large Databases. In: *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*. ACM, New York, (1995), p. 127-138.

CodeReview: Organizing and Reviewing Software Projects

Tomáš KEPIČ*, Patrik ORISKÓ*, Michael SCHOLTZ*, Július SKRISA*, Patrik SAMUHEL*, Matej CHLEBANA*, Zuzana GREŠLÍKOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`tp.1314.07@gmail,com`

1 Introduction

Software projects are made by teams of developers and communication between developers is important for successfulness of projects, e.g. developers in one team have often different knowledge and skills. As a result a developer would not see a problem in his source code, but another teammate could see it immediately just because he had the same problem in the past. Therefore if developers are able to review their codes effectively and immediately after source code modifications, the quality of projects will increase and developers can learn new knowledge and skills from more skilled developers. Our system CodeReview[1] helps developers, especially students, to communicate directly through their source code and to find and predict problems as soon as possible.

The uniqueness of our project is based on adding information tags. We are cooperating with other project that is creating environment for usage of these information tags in code. User can easily add information tag to code through plug-in in VS or Eclipse or through our web portal CodeReview. This tag has information about the mistake, bad habit or bug in the code. Author of code that was marked will be notified by our web portal to fix or check this issue/information.

The main goal of the project is to simplify and accelerate the development of software projects and enhance project quality by direct feedback and good communication between software developers. System should be used like a tool where team members communicate and control source code quality.

2 Architecture

The frontend application is written as a web application using the powerful .NET environment. This application is primarily based around displaying source code, data and metrics to our end users. It uses separate database system in order to connect users to their projects.

The backend application is powered by AST-RCS (Abstract Syntax Tree Revision Control System). Each repository is initially added into AST-RCS. After this initial setup, this repository is

[*] Master degree study programme in field: Software Engineering/Information Systems
Supervisor: Karol Rástočný, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] http://labss2.fiit.stuba.sk/TeamProject/2013/team07is-si/

regularly monitored and every change is being recorded inside AST-RCS system. Frontend layer then uses these changes and displays them upon users request (see Figure 1).

Users have to initially register into our system. After they successfully log in, they are allowed to create "Projects". Each project can include multiple repositories and multiple users. These users have access to view and interact with source code files. They can also compare different versions of source code and they can also view code review comments (in form of information tags) from their fellow colleagues.

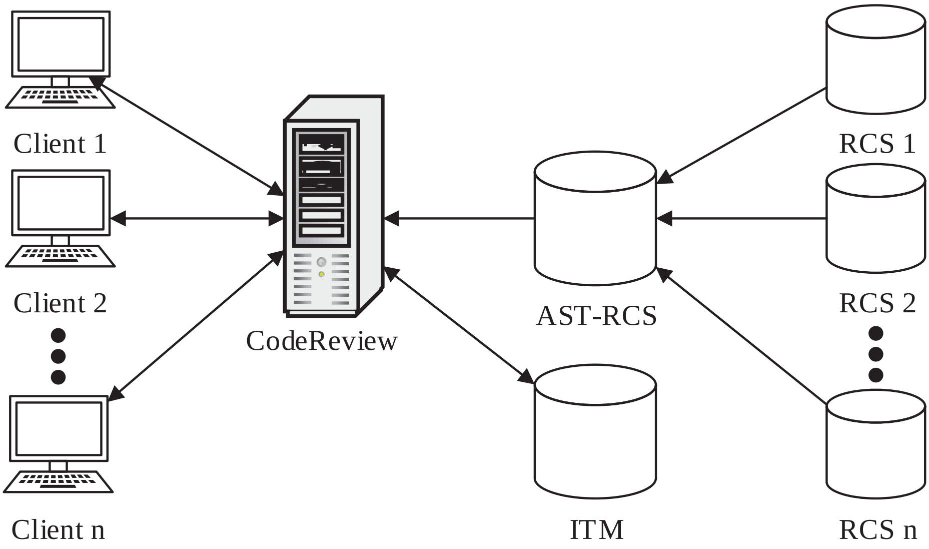


Figure 1. Architecture of the CodeReview system. AST-RCS represents database of revision control system with source code in format of abstract syntax trees. ITM is database of information tag management system.

3 Conclusions

All in all the system aims to provide easier means of communication between different members inside larger teams. The goal is not only to help them analyse their code, but also to offer a tool which helps them make their code reviews more useful and effective, which should ultimately enhance code as well as skills of individual team members.

References

[1] Bieliková, M., Polášek, I., Barla, M., Kuric, E., Rástočný, K., Tvarožek, J., Lacko, P.: Platform Independent Software Development Monitoring: Design of an Architecture. SOFSEM 2014, LNCS 8327, (2014), pp. 126-137.

Askalot: An Educational Community Question Answering System

Rastislav DOBŠOVIČ, Marek GRZNÁR, Jozef HARINEK, Samuel MOLNÁR, Peter PÁLENÍK, Dušan POIZL, Pavol ZBELL*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
askalot@fiit.stuba.sk

Often we find ourselves in a situation that we search for something that cannot be easily found anywhere on the Internet. One option to obtain such information is to ask a community and employ common knowledge of its members [2]. This kind of collaborative knowledge management gained attention lately and a number of novel systems based on utilization of collective intelligence were introduced. Since the content of such systems is created by the community, the systems provide a rich source of information, not available anywhere else. In our project, we specifically focus on Community Question Answering (CQA) systems, such as Stack Overflow, Yahoo! Answers or Quora.

The main purpose of CQA systems is to find or give answers to various questions on the open web. Since they are used to share knowledge, they have a great potential to be used also for academic purposes (e.g. [1]). Therefore, our main goal is to propose and develop a CQA system Askalot which is focused on the domain of education. We plan to implement functionality that supports the educational aim and specifics of universities, e.g. a strong role of teacher who is able to moderate the discussion about questioned topics. Our system is also a closed community system, therefore students can ask community specific questions.

Our goal is to create a system that could be widely used on the field of faculty by our fellow students and teachers. Lack of such system at our faculty and existence of obsolete forum that served such purposes (sharing knowledge among students), lead towards the thought to create an educational CQA system that could not only replace outdated forum, but even more, support real time questioning on lectures (our system is integrated with startup sli.do, which supports questions asking during lectures), simplify asking various questions about actual topics at school and make an organized and easily searchable database of knowledge for current and future students (especially, shift students' focus from asking questions in isolated Facebook groups to our CQA system).

Based on our assumption that community agrees upon their common knowledge, we believe that it is possible to collect right answers by the community of students. Since there are many of them, we can say that the answers to questions will be verified by others. On top of it, there is always a teacher who can answer or verify the student's questions and answers. The aspect of education in our system is reflected in the role of a teacher. The teacher has several opportunities to lead the collaboration among students: give feedback by evaluating questions and answers,

[*] Master degree study programme in field: Information Systems
Supervisor: Ivan Srba, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

comment on them or give the right answer. The first possibility is give a feedback by means of five grade scale on which the evaluation of question or answer quality can be performed. Another method of teacher's participation is commenting on answers or questions (the comment from a teacher is highlighted). By commenting, the teacher can lead the discussion into the right direction and can help students in their effort to answer the question. The teacher can also answer the question directly, which produces highlighted answer, so students can easily see the content added by the teacher. With these three improvements of the community answering process, we can obtain better answers, and the students can easily see the teacher's opinion about the particular content. The teacher can influence the problem-solving process in a way she desires. However, we do not want to create a system that has the most of the content generated by a teacher, since the teacher does not have time and capacity to answer everything. Therefore, the community of students is supposed to answer questions by themselves as it is in regular CQA systems. It is also meant to be the primary source of knowledge while teacher's collaboration is supposed to be a supporting one.

To encourage students in asking questions, there is an opportunity to ask a question anonymously. We believe that this feature is a motivating factor for some students, when he or she is not sure about the quality of the posted question, but still needs the piece of information.

As soon as the employment of concepts of CQA systems in educational domain represents an open research problem, our system has also advanced method of logging. In our application we log every action that is performed with corresponding data and application state. As a result, we are able to obtain a comprehensive dataset for further research use.

The proposed system is based on open source technologies. It is implemented in Ruby on Rails, a framework for creating web applications. We also use Bootstrap, a CSS framework that helps us to build responsive layout for our application. The quality of our code is assured by relying on test driven development and regular code reviewing.

Askalot has been already successfully employed as a part of educational process at four bachelor degree courses at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava. During the first five weeks of its production deployment, more than 500 students used our system. The fast adaptation of the question answering concept by students promises further positive results.

The main features of our system are: (1) Question asking support; (2) Notifications; (3) Followings; (4) Watchings; (5) Tags (search by tags);

In our future work, to encourage students in knowledge sharing process, we plan to develop a student motivational system which will include social elements, e.g. following other users or integration with social networking sites. Moreover, the student motivation will be enhanced also by well-known system of badges or achievements. Besides motivation, we plan to support creation of high-quality content, too, e.g. by means of an algorithm which will be able to compare the questions and filter out similar or even duplicated questions at the time of their creation.

The main contribution of our work is the proposal and implementation of the educational CQA system that is specifically designed for supporting of community question answering process at a university. Askalot is not only a tool to support students' learning, but it also provides a great possibility to collect a robust dataset with plenty of user interactions to be analysed subsequently.

References

[1] Barr, J., Gunawardena, A.: Classroom salon: a tool for social collaboration. In: *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, (2012), ACM Press, pp. 197-202.

[2] Liu, Q., Agichtein, E., Dror, G., Maarek, Y., Szpektor, I.: When web search fails, searchers become askers. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '12*, (2012), ACM Press, pp. 801-810.

IDEM
Programmer's Monitor

Ján PODMAJERSKÝ[1], Ivan KOŠDY[2], Michal JURANYI[1]
Jozef MARCIN[1], Tomáš MARTINKOVIČ[1], Juraj RABČAN[1]
Matej NOGA[1*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`tp-tim-8@googlegroups.com`

These days there is a tendency to use information technologies everywhere, or in other words there is a necessity to use information technologies in order to simplify way of work. Customers require having software delivered as soon as possible. Despite using agile programming software companies are usually unable to deliver product on time in demanded quality. Current tools which provides the opportunity to see the quality of code are - in the cases known to us - does not deal with possible evaluation of program code partially, thus according to the author. They only evaluate the overall quality of the program code. Associated with this is the fact of inability the comparison of programming skills within a specific project. We aim to offer software source code control of programmers in teams, or project respectively. The statistics are meant to display the quality of every programmer, or the complete view of the project.

The main goal is to offer monitor of programmer in IDE (Integrated Development Environment). We are developing an application using the best-known metrics to evaluate mainly the quality of object oriented source code (Java). Tracking of the users and measuring the core characteristics of keyboard usage and production of the source code enables to create user model, which is essential for personalization of the code. The rhythm of writing is used to detect users, as the most natural metrics. We can realize the author of the code, by storing time of typical writing style; we compare the changes of code times. To distinguish the users' source code we need to use software metrics mainly– control variables count, deep of inheritance and the copied code (re-usable code). Thanks to the users' stereotypes and PerConIK project applications, which captures important users' data, we can evaluate the code's author (and its specific parts), subsequently to measure its quality. The results are relatively compared to the team members and judge each programmer's quality and assessment. Another very interesting aspect, which can be explored of the data, is the feeling of programmer, which enables the rate of copied code and his own ideas.

The basic goal is to offer an application running on web, which provide availability for everyone, using client-server internet application. The design of our application is shown in Figure 1.

For logging the data from user we use PerConIK tools developed by Gratex company. PerConIK (Personalized Conveying of Information and Knowledge) is name of the research project co-funded

[*] Master study programme in field: Information Systems[1] / Software Engineering[2]
Supervisor: Assoc. Professor Daniela Chudá, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

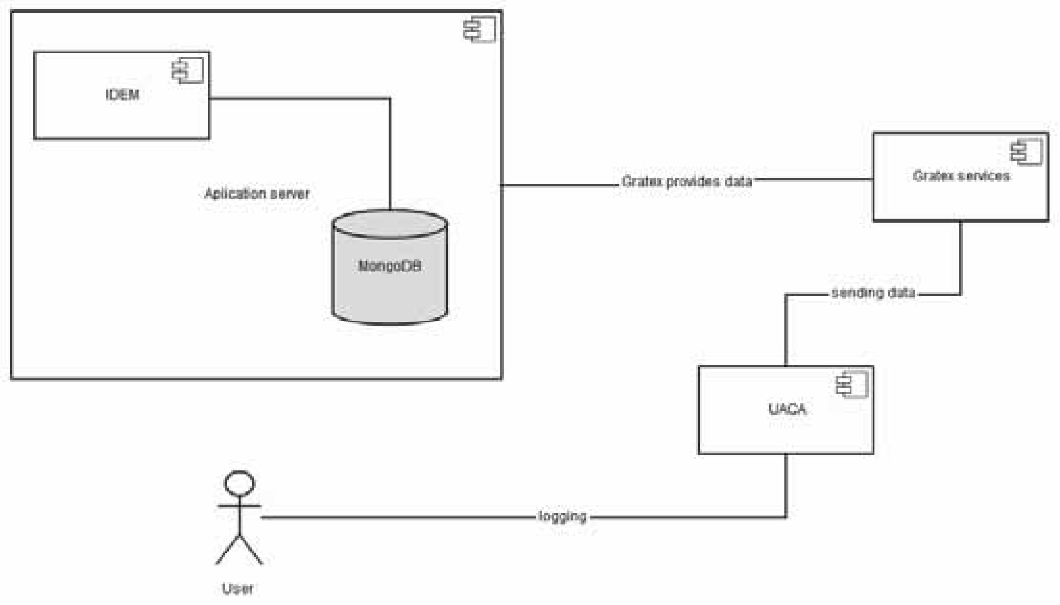


Figure 1. IDEM architecture scheme.

by the European Regional Develoment Fund. The reseach is being done by FIIT STU in cooperation with Gratex International a. s. One of the tools we use for logging user data is UACA (User Activity Client Application) which subsequently with defined frequency and data volume, sends this data via web service to server provided by Gratex. Data is then loaded and stored in Gratex database. From this database we obtain data directly into our application in which we process them, evaluate them a subsequently, we store them to our database. Results are then made available for viewing by user.

Core element of application is module of evaluating quality of program code which works by using various of code metrics. The most common used metrics are Cyclomatic complexity, LOC, CLOC etc. Another metrics we further want to use are Weighted Method Count - total complexity of a class, Average Method Weight - average complexity of a method and use the C.R.A.P and inheritance depth etc. Tools we currently provide are web application used for management and viewing logged data and Eclipse IDE plugin for logging data.

We use MongoDB database in which we does not need to have created a relational model. Next, we use the data serialization by JSON. This two data approaches enable to elude creating relational model of database. Next element of architecture is Glassfish server, Spring and we work in Eclipse EE development environment.

The major asset of our project is provide feedback to leads of programing teams and reveal lack of program code from perspective of programing quality of individuals in team. The unique on our solution is almost fully automated logging of programmers activity and source code metrics and viewing results in web application the way that programmers can view their metrics as they are changing in time and comparing with other programmers in team (or at least with project average values). Project managers / leads can additionally view and compare values of all project members. All data are displayed in tables or visualized using easy to understand charts.

Application for Funtoro Platform

Ján ONDER, Martin POLÁK, Tomáš TRÁVNIČEK, Dávid URBÁN, Lukáš ZEMANÍK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
funteam2@googlegroups.com

Many companies have a lot of cars or trucks which are difficult to keep track of and maintain. It is also hard and time consuming to make logbooks[1] manually for them. Logbooks are nowadays created after every ride even when it is just in one town. After certain period (usually a month), they have to be rewritten into digital form so this data can be used for tax purposes.

Our goal is to make this process easier, more enjoyable, accurate and better for employer than it is now. With series of interesting statistics it is possible to lower company expenses on vehicles, because drivers will be rated according to their driving style and also their average fuel consumption.

All this is possible with new complex devices, which provide very accurate position informations from GPS satellites and a wide range of telemetry data from vehicle (through OBD-II CAN bus, which is common in most of modern vehicles).

Another and also very important goal, is to protect people from various threats and to help them have less stress when something happens. For example using our product in families. Most parents know how stressful and scary it can be when they do not hear from their children for a long time and do not know where the children can be or what has happened to them. We believe that with modern technology this should be a history and with our product, parents can simply find out the location of their lost children quickly and safely on our website. This all is possible with monitoring devices, because of their accurate position tracking, and also SOS button, which all of our personal devices have.

And last but not least, people who are enjoying hiking, skiing, snowboarding or other outdoor adrenaline sports are also exposed to greater risks, for example there could be an avalanche, or they could wander out of marked trails and get lost. With our system composed of a personal monitoring device, its proper configuration and a web portal, it is possible to localize people who are lost in these extreme situations and help save their lives.

Nowadays, there are several competitive solutions available that are able to achieve some of our goals. One of the best solutions is ”ONI system[2]“ with real time tracking of vehicles and with good support. Problem of this solution is in the design of its page – it’s difficult to find some tools and also generally working with this application is not intuitive. Also this product doesn’t support interaction with mobile devices. These are the things we were focusing on during our implementation to make it more user friendly and more available in terrain.

[*] Master degree study programme in field: Computer Engineering
Supervisor: Peter Pištek, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

[1] http://www.pp.sk/6577/Motorove-vozidlo-vo-firme--vzory-internych-smernic_A-PMPP30847.aspx, http://www.epi.sk/odborny-clanok/Cestovne-nahrady-v-praxi.aspx

[2] http://www.onisystem.cz/

The most important component which will be physically located at the customer, is GPS tracking devices. These devices send important information to our server via GSM network. Quantity and actual content of this information depends on type and configuration of the particular sending device. In case of devices with ODB-II CAN connector, devices send in addition to GPS position data from ECU (electronic control unit) of the car. These data may include actual fuel consumption, fuel level and much more. Information sent by devices to our server are processed by parsing program, which analyzes them and inserts them into database. The most important part of our architecture for the customer is web portal. This portal is also optimized for mobile devices.

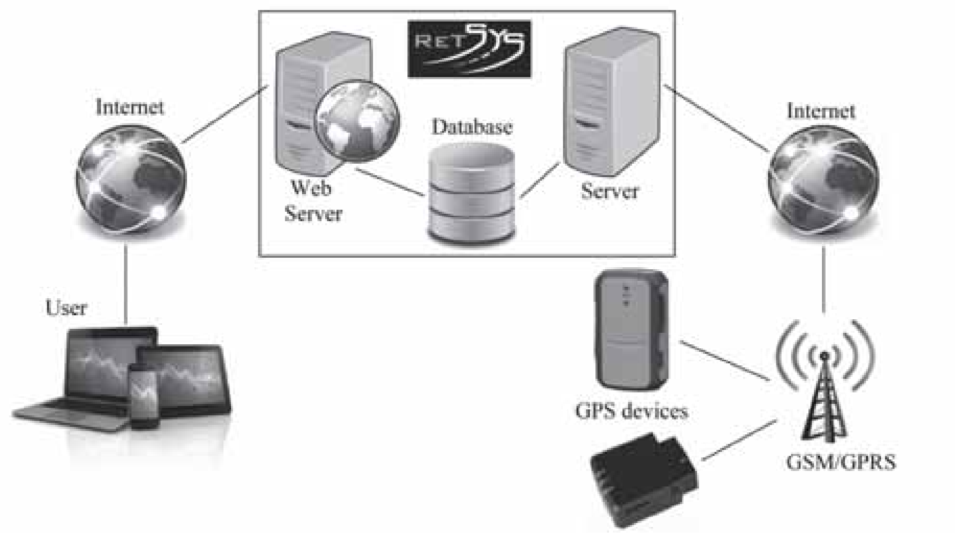


Figure 1. System overview.

On figure 1, we can see overview of our system. If going from right to left, our first units are GPS devices, which send all collected information via GSM/GPRS network and through the Internet to our main server. Our server then processes these data and stores them in database for future use. Last unit of our system, which communicates directly with our customer (represented by personal computer, smartphone or notebook) is Web server. After logging in, the customer can track movement of his own cars, people, or he/she can add information to logbook, which cannot be inserted automatically, e.g. purpose of ride. The customer can also display various statistics, create his own warnings, for example imminence of technical control or emission control. It is also possible to change configuration of devices, directly from our web portal.

To make the configuration of devices easier, our system is connected to our own SMS gateway, which sends configuration profile directly to device. Without this feature, it would not be possible to configure devices from portal – it would be done only by connecting the device directly with a computer via USB cable and configure it through configuration utility.

When logbook entry is created and the ride is over, people that were participating in that ride receive an e-mail with link to our portal, where they can add details about their ride.

Every larger customer have the opportunity to add his own users with custom permissions. These permissions specify what user can see and what he can modify – for example when someone doesn't have access to certain vehicle, he/she cannot add details about its ride. But when accountants need data from every ride for every vehicle in fleet – they can see logbooks. But they cannot see details of these rides.

All these features are covered in one complex system named RetSys (Real Time Tracking System).

Carlos - Car Entertainment System

Patrik POLATSEK[*], Martin PETLUŠ[*], Jakub MERCZ[*] Lukáš SEKERÁK[*],
Peter HAMAR[*], Róbert SABOL[†]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
team03.1314@gmail.com

Entertainment and information systems become the important part of our lives. One of the most modern and natural human-computer types of interaction is an *augmented reality* (AR), where a real-world environment is supplemented by virtual data, such as visual, textual and audio data.

Our aim is to create a prototype of an interactive system with a user-friendly interface for fellow travellers in a car designed for entertainment and educational purposes. The proposed system called *Carlos* changes a car side window into a *transparent projection screen* to supply the surrounding reality with the virtual information. The system creates an AR on a car window, due to which it can inform travellers about the immediate environment in real-time.

Carlos visualises the information on a side window with a transparent film using a small LED projector. The whole system is controlled using a mobile phone with gestures, voice commands or rotation of the device.

Carlos detects interesting objects such as sightseeing, restaurants and hotels on images captured by a camera mounted on a car. In the detection phase it compares the images using the actual GPS position and the internal database of objects of interest. The detection starts with the selection of potential objects close to the GPS position automatically received from a mobile phone. Subsequently the object detection is performed using the feature extraction and matching methods. After the successful detection the system computes the location of objects using the homography [1]. Due to proper displaying the information, Carlos works with Kinect device to detect the user's head position. Then the location of detected objects is recomputed in order to the precise placing of virtual information for the actual user's gaze at the window. Finally, Carlos projects on a window the basic tourist textual and visual information for detected objects.

Carlos is not only the information system, but also the entertainment system. It uses the object detection also for an educational game based on the answering a question related to the object. Another AR game is a flight game which aim is to keep a plane above the horizon as long as possible. In order to detect the horizon the system detects sky regions with an edge detection algorithm. The flight of a plane is controlled by simple gestures on a screen or rotation of a smartphone.

Most AR car systems create an AR on a front window to display navigation information or increase the safety by detecting objects close to the car. An example of an entertainment system which aim is closer to our system is a project called *Touch the Train Window* by

[*] Master study programme in field: Information Systems / Software Engineering
Supervisor: Dr. Vanda Benešová, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

Salad[1]. The system just allows a user to place virtual objects on a window in a train using GPS position and Kinect device which tracks the user's hand and window.

Our system is implemented in C++ and Java language using OpenCV, OpenGL and Freenect library. It consists of several modules which structure is presented in Figure 1:

- **Control module**: Each module is controlled by this single central module. The module manages other modules and their inputs and outputs, receives images captured by camera and displays virtual information using a projector.
- **Image processing module** implements the object detection algorithm using feature descriptors. The module returns the position and name of detected objects on an image captured by a camera. The module also detects the horizon using the edge detection.
- **Kinect module** processes video and depth information from Kinect device. It detects the face and computes its actual position.
- **Module of text position computing** calculates the correct position for projected virtual information according to the traveller's actual gaze using the position of detected objects and the position of the face.
- **Android module** is implemented in a Java application for a smartphone with OS Android. It contains the interface for controlling all Carlos applications. The module implements 3 types of interaction – gestures, voice and rotation of the device and records the GPS location.
- **Augmented reality module**: The primary aim of AR module is to create the visual information projected using the projector. This module implements 2 types of games. The first one is a quiz game which asks questions about detected objects. In another game a user controls a plane and tries to keep it above the detected horizon.
- **Database module** manages an internal database which contains photographs, GPS positions and essential information about objects.

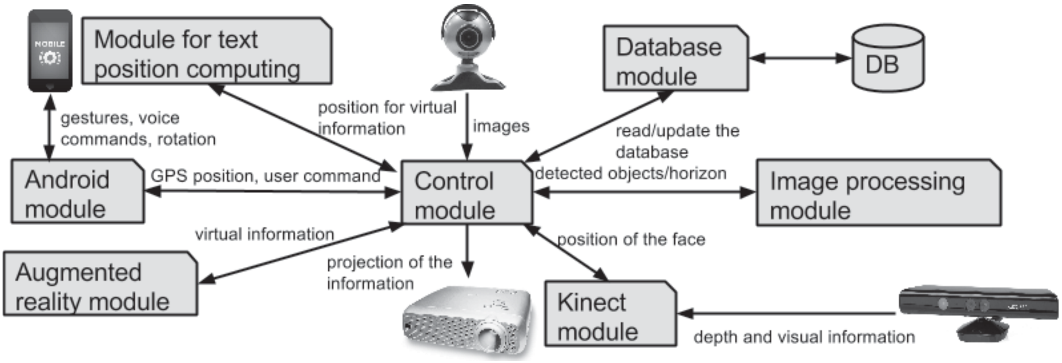


Figure 1. Structure of our system.

Acknowledgement: This project is partially supported by Volkswagen foundation and VEGA 1/0625/14.

References

[1] Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

[1] http://csp-salad.com/

FIIT Grid – Distributed Computing Network

Juraj VINCÚR*, Juraj PETRÍK*, Pavol PIDANIČ*, Ján KALMÁR*, Ondrej JURČÁK*,
Radoslav ZÁPACH[§], Martin TIBENSKÝ[§]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
tp-12@googlegroups.com

Nowadays public and organisations such as universities or small research centers possess quite large amount of computers, which are mainly used for administrative work. These computers are often idle and their performance limits are far further from their actual usage. Volunteer computing, or under special conditions grid computing, can turn these collations of computers into powerful tool for handling computational problems with high complexity. This approach is both financially beneficial for researchers and friendly for nature, because there is no need for additional specialised super computers to solve these tasks in reasonable time.

In terms of our project, distributed computing can be defined as collaboration of many nodes on processing computationally complex problem. In comparison to alternative understanding of distributed systems, nodes does not share their resources in traditional way, they offer themselves as independent entities. This fact denotes some restrictions for set of problems, which can be efficiently solved in distributed way – it must be possible to break down these problems into subproblems. These subproblems are distributed to nodes, which performs necessary computations and returns results to central node for further processing.

Distributed computing can be divided into two main categories mentioned above –volunteer computing and computing using clusters. Both approaches uses the philosophy of distributing problem to nodes, but each differ in characteristics determined by environment of distributed network. Computing using clusters relies on nodes owned by organisation. These nodes can be controlled by organisations IT professionals, they are typically connected with high-bandwidth network links and available whenever they are needed. Under this level of control, projects can be performed on nodes silently, completely hidden from node user and results returned from the nodes can be explicitly marked as trusted, so malicious behaviour is neither expected nor handled. Opposed to computing using clusters, volunteer computing networks consist of nodes with many heterogeneous users, operation systems, network connections and resources. Node owner is fully aware of participating in project. Under these conditions, systems used for volunteer computing must handle situations as frequent connecting and disconnecting of nodes, efficient distribution of work based on nodes resources and possible malicious behaviour. One of such systems which we use for our project is BOINC system.

BOINC (Berkeley Open Infrastructure for Network Computing) is software system that offers researchers tools for easy creation and management of distributed computing projects. Although BOINC can be used also for grid computing, it was originally designed for volunteer

* Master degree study programme in field: Software Engineering

[§] Master degree study programme in field: Information Systems
Supervisor: Dr. Peter Lacko, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 629–630.

computing projects. BOINC was first time widely introduced with SETI@home project which is focused on finding extra terrestrial life and today has more than one million participants with total performance of 500 teraflops. For comparison, the average computer with Intel i3 processor has around 50 gigaflops, which is ten thousand times lesser. Main goals of our project is to introduce BOINC system to researchers on our university, provide them with useful extended user guides and find the initial set of participants for their projects [1].

To fulfil these goals, we deployed BOINC system on one of university servers. To make user guides, which will possess additional value we chose to make our own projects. Experiences and skills obtained during the creation of these projects will be transformed into materials for university researchers, which are in need for greater computing performance for their own projects.

Our first project has ultimate goal of finding ultra weak solution for symmetrical Reversi 8x8 game. In case of success, we will be the first team worldwide, which found the solution for board of this size. Reversi game is perfect candidate for volunteer computing project, because the natural distribution of game tree allow us to generate subtrees as subproblems and send them for processing to project participants, everything with help of BOINC system.

During the preparation for this part of project we implemented two prototypes for solving 6x6 board. Solving board of this size was really helpful, because the correct solution is already known and performed tests allowed us to validate the accuracy of our algorithms. Testing of our first prototype written in Java also showed us pitfalls of using this language in BOINC system. Java is not fully supported, therefore programmer must use wrapper and access to BOINC API is complicated. Also for this kind of computation Java virtual machine’s garbage collector caused excessive consumption of system resources. We decided to solve these issues by making our second prototype written in pure C language. With this prototype we gained easy access to BOINC API and was able to find ultra weak solution for Reversi 6x6 remarkably faster.

One of the main advantages in distributional approach of solving problems like Reversi game tree, is not only greater computing performance, but also distribution of required storage, which a researcher with single device would find almost impossible to acquire with limited funding.

Our second project focuses on developing a tool for DNA researchers. The main aim of this tool is to take simulated DNA reads from systems like Illumina and then test the capability of various parameters in the process of DNA sequencing. Due to the amount of possible variations of such parameters, it would take a very long time for a single node to reach sufficient result. Our design solves this problem by dividing variations of parameters into subsets and distributing these subsets among nodes. The final goal is to find the best parameters for sequencing human genome, but due to the relatively large amount of data, which would be sent to each node, we chose to test our system on complete genomes of bacteria and other lower organisms or on an individual human chromosome. Projects like this present a great example of how can volunteer computing return favour to the volunteers by helping to advance the research beneficial for society.

Since single BOINC project can hold many applications, it is suitable for educational purposes in larger groups. With our documentation and user guides, students of courses focused on solving the problems in distributed way, will be able to easily test their own algorithms and applications in heterogeneous environment, which is provided by widely used BOINC system.

References

[1] C David P. Anderson. 2004. BOINC: A System for Public-Resource Computing and Storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing* (GRID '04). IEEE Computer Society, Washington, DC, USA, pp. 4-10.

PINTA.SK - Feedback Providing Community

Filip BEDNÁRIK, Róbert ČERNÝ, Miroslav MOLNÁR
Marek LENČÉŠ, Patrik ŠTRBA, Miroslav VOJTUŠ
Martin TOMA[*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
`contact@pinta.sk`

Even today, with all the existing technology, there still is a problem with obtaining fast, high quality feedback from real people. We all know that the constructive feedback is a crucial element in many businesses nowadays. Companies hire marketers which do their best to sell their products. Marketers struggle finding people providing valid feedback. Usual approaches like phone marketing, interviews and surveys lack motivation for people to answer. These ways are not entertaining nor do they make any profit for feedback provider.

We extend this problem by looking on term feedback as not only answer to a question or opinion in that matter but also evaluation, some action or micro procedure. Our main goals we aim to achieve:

– build online environment for exchanging quality feedback,

– help marketers aim their campaign for specific group of people (age, location, gender, qualification, interests),

– motivate feedback providers to get involved by rewarding them and by making the process of feedback leaving simple and entertaining,

– provide evaluated feedback results back to marketers.

To achieve these goals we propose web-based application for on-line feedback exchange called "Pinta.sk". This application should provide powerful, yet easy to use tools which will help us to achieve these goals. In order to be able to achieve these goals, we must create the ability to use and combine multiple filters to exactly specify target group, implement some of the gamification principles, which are very popular nowadays and particularly set up an effective and expendable validation mechanism which will greatly improve feedback processing process.

There are two types of users in our application. Requesters are asking for high quality feedback and providers are supplying this need. Each of them travels through a specific process. In case of feedback requester the basic flow will be:

– build online environment for exchanging quality feedback,

[*] Master study programme in field: Software Engineering / Information Systems
Supervisor: Dušan Zeleník, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

- help marketers aim their campaign for specific group of people (age, location, gender, qualification, interests),
- motivate feedback providers to get involved by rewarding them and by making the process of feedback leaving simple and entertaining,
- provide evaluated feedback results back to marketers.

Additionally in case of worker, or feedback provider, basic work-flow will look like this:

- build online environment for exchanging quality feedback,
- help marketers aim their campaign for specific group of people (age, location, gender, qualification, interests),
- motivate feedback providers to get involved by rewarding them and by making the process of feedback leaving simple and entertaining,
- provide evaluated feedback results back to marketers.

There are some other applications like us for example Amazon Mechanical Turk[1]. We identified several weaknesses we believe this system is suffering from.

- basic user interface which is not user friendly,
- lack of motivation and involvement in community,
- lack of social interaction and other competitive elements,
- lack of good result presentation,
- lack of support for European countries including Slovakia.

There are also others micro-work providing portals like MicroWorkers[2] or EliteWorkers[3], but they all either do not focus primary on marketing and academic layers as we do, or are also operating only on local districts. So, we have focused to set our features priorities based on AMT weaknesses. Therefore, our unique value will come from:

- very intuitive and user friendly UI,
- support for Slovak market,
- gamification features,
- unique live result presentation.

From technical point of view our application uses modern web technologies like HTML5, CSS3, JavaScript, SASS and is built on Ruby on Rails. Our main server runs on Linux distribution Debian. We use Git for version control and we automated the process of integration with use of Jenkins. Issue tracking is provided by Atlassian Jira with module Jira Agile.

[1] Amazon Mechanical Turk, `https://www.mturk.com/mturk/welcome`

[2] Micro Workers, `https://microworkers.com/`

[3] Elite Workers, `http://eliteworkers.org/`

High School Students at IIT.SRC Junior 2014

Jakub ŠIMKO and Mária BIELIKOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
{maria.bielikova,jakub.simko}@stuba.sk

Abstract. The IIT.SRC Junior track is a platform for talented high school students interested in informatics and information technologies to present their innovative ideas and projects to their senior colleagues – university students and staff. During poster sessions, works accepted to the IIT.SRC Junior track have been presented by their authors, who subsequently received feedback on their projects throughout discussions.

1 Seeking the talent

Seeking for the talented high school students is essential for maintaining quality of future IIT.SRC conference submissions as well as the life of the faculty. Therefore, we repeatedly started the IIT.SRC Junior track – a platform for high school students to present and discuss their innovative ideas and projects in the field of informatics and information technologies. Previous years of IIT.SRC showed to be promising since we managed to involve several talented high school students who recently became our students.

Student works accepted to this track have been presented by their authors during regular poster sessions. Here, the authors had the opportunity to receive valuable feedback from the faculty members as well as from their older colleagues. The authors had also the opportunity to view and discuss other works presented at the conference to gain experience and inspiration for their future projects.

2 IIT.SRC Junior 2014 Projects

This year, three submissions were selected. All of them presented as extended abstracts for more detailed explanation of proposed ideas and realized prototypes. The first project, authored by Martin Pavelka is aimed onto effective re-use of hardware in a high school by “scavenging” older computers. Three steps were defined and performed considering actual state of hardware and software infrastructure: (i) inventory, (ii) building up computers and making changes, and (iii) looking for the best software solution. Purpose of this work is to improve the level of education and services in schools for students and staff, as well.

In the second project, Miloš Prokop devised a device supporting moderate handicapped people in using of the mobile phones. Although it contains very simple user interface, it provides many useful features. The features include access to stored contacts, adding new contacts, making

* Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

and receiving a call, writing and receiving an SMS message and reading stored SMS messages. It is built as a working prototype, which consists of large display and PS/2 port for interfacing and PC keyboard for input.

The third project was authored by Miroslav Šedivý and involved a creation of intelligent content management system for web environments (iCMS project). Special attention is paid to user interaction with the system, which is critical to its acceptance. Current realization of iCMS offers full compatibility with existing mobile devices and web-browsers. The aim is to make the changes on the website easier comparing with existing CMS systems, so that the end user has the option to change the whole design by one click.

More information about the IIT.SRC Junior track can be found on the Web:
`http://junior.fiit.stuba.sk/`

Rational Usage of Information Technologies in our School

Martin PAVELKA*

Evangelical Lyceum
Vranovská 2, 851 03 Bratislava, Slovakia
martinpavelka28@gmail.com

Abstract. In this paper we will describe our ideas how to improve usage of digital technologies in our school. The main point is to reuse older computers by setting them up to run a suitable edition of operating system. One computer will be combined with hardware from other ones to improve the performance. These refurbished computers may be used in the library or in the cafeteria. Advantage of this solution is, inter alia, software support from Microsoft for non-profit organizations in Refurbishers program.

1 Introduction

We had several opportunities to visit the storerooms of information technologies in some organizations. The conclusions of those visits are amazingly positive – we found there many pieces of hardware that could be reused with minimal expenses. In the storerooms were many older lower performance computers running old versions of operating systems – Windows XP or older. By rational combinations of still working parts we may create upgraded computers, for example from 2 computers with 512 MB RAM, 40 GB HDD, Pentium 4 processor we may create upgraded workstation with 1 GB RAM, 2x40 = 80 GB HDD. This is enough to run the well-established Windows 7 with appropriate settings. Described computer is able to run basic programs such as Mozilla Firefox web browser, Microsoft office and Media Player Classic.

2 The First step – Inventory

The first step for rational using of IT in the organization is the inventory of all available components which may be used. In some cases we may find devices, which may be repaired in easy and not expensive way, but also devices that can be sold in specialized bulk-buying center.

The special category of devices in storerooms are laptops – if their cooling pipes and fans are cleaned then they may be used instead of classical desktop, because the laptop needs less space. The main point of reusing older laptops is removing the non-functioning battery and storing it in the safe place or rather keeping it, we may send it to ecological liquidation. The result of the inventory should provide the list of all peripherals and devices that can be used.

[*] IIT.SRC Junior contribution
Mentor: Pavol Helebrandt, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

3 The Second step – building up computers and making changes

When building/repairing computers or workstations several configuration variants can be used:

I: Computers with configuration less than 1024 MB, 30-40 GB HDD, processor Celeron or P4 may be used as internet terminals for students in public places of school; in cafeteria for it specialized software or in library for displaying a catalogue of books.

II: Computers with configuration 1024 MB, 80 GB HDD may be used as computers for staff or computers in classrooms which are connected with projector.

Reusing of older computers is supported by Microsoft - Microsoft Registred Refurbisher program [3]. Microsoft Company provides free articles about refurbishing computers in form of PDF files [2]. The software solutions for these variants of configurations are described below.

4 The Third step - Looking for the best software solution

For older computers with configuration less than 1024 MB RAM is better to use operating system Lubuntu [1], which is – in our opinion - the best OS for older computers. The previous statement is based on results of our survey that looked for the following OS parameters:

- Increasing computer speed by choosing appropriated OS
- Finding similar GUI to the most used OS – Microsoft Windows
- OS with easy administration and usage for users – no need to learn plenty new things
- Appropriate network abilities – web browser, LAN configuration

For computers with performance for Windows 7 [5], we suggest to run Windows 7. School can obtain this OS for special discount prices [4].

In both cases it is necessary to prepare installation medium with settings for optimization and pre-installation of basic software programs – text processor, table calculator, presentation design program, web browser, audio and video player, website bookmarks on the desktop (the organization´s homepage, online ordering lunch system, etc.). The renewed computers should be connected to the part of the LAN isolated from other computers, because they may be much more vulnerable to infections according to the fact that they are shared by many users.

5 Conclusion

In this article we outline our methods and procedures for reuse of older functioning computers in organizations – concretely in our school – by refurbishing. We prepared short manual what to do with older computers, how to sort them and what operating system is best to use. When speaking about refurbishing in educational spheres - final expenses of my solutions would be minimal because schools can get OS Windows for special prices.

We hope that our solution may improve the level of education and services in our school for students and staff, as well.

References

[1] Lubuntu system requirements https://help.ubuntu.com/community/Lubuntu

[2] Manuals for refurbishing computers – TechSoup resources http://www.techsoup.org/support/articles-and-how-tos/manuals-for-refurbishing-computers

[3] Microsoft Registred Refurbisher http://www.microsoft.com/refurbishedpcs/RRP.aspx

[4] Microsoft support for highschools http://www.microsoft.com/slovakia/ education/schools/

[5] Win7 Req. http://windows.microsoft.com/sk-sk/windows7/products/system-requirements

Device Designed for Handicapped People to Facilitate the Use of Mobile Phones

Miloš PROKOP*

Slovak Michal Miloslav Hodža Grammar school in Liptovský Mikuláš
Hodžova 860/9, 031 36 Liptovský Mikuláš, Slovakia
`prokop.milos@gmail.com`

Abstract. This paper presents a prototype of a device intended to help minority of people, who are a little handicapped, with the use of mobile phones. It can help people who are living alone to let their relatives know they are well. Our system has large display and a PC keyboard port so the use is much simplified. It must be connected with a smartphone via Bluetooth to use its features.

1 Introduction

People often do not care about handicapped people and instead of helping and supporting them they usually ignore them. Our objective was to find a way to help them. We decided to use our experience to create a device which can help them to connect with their relatives. Fortunately, many handicapped people who live alone can use their mobile phone. But they are also many of them who cannot because of their health status. This is the group of people, for which our device is intended. It includes people with visual impairment disorders or trembling hands, who at the same time, posses only limited IT skills. This group often includes seniors.

We wanted to make the device to be as simple as possible. Although it contains very simple user interface, it provides many useful features. The features include access to stored contacts, adding new contacts, making and receiving a call, writing and receiving an SMS message and reading stored SMS messages.

The device we have created is just a prototype. It needs to be connected with a smartphone via Bluetooth. The purpose of creating it was not to make a perfectly working device which can be immediately used by the target group of people. We have built it to present our ideas and to find out if there are any people interested in such device. In that case we will design new one, determined for everyday use without any problems. The biggest limitation of our system is that the device needs a smartphone to be paired with. It was the cheapest and the easiest way to build it.

[*] IIT.SRC Junior contribution
Mentor: Maroš Ďuríček, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 637–638.

2 Proposed system

By user view the device consists of large display and a PS/2 port for interfacing a PC keyboard for input. But it can be internally divided into several parts by their function. These parts are processing units, display, display controllers, Bluetooth component and voltage regulators.

The device consists of two processing units ATmega32-16PU [1] and ATmega162-16PU [2]. The first one takes care more or less only about display and communicates via UART with the second microprocessor which communicates with smartphone, encrypts and decrypts the data for the smartphone and tells the ATmega32 processor what to display. Both microprocessors run on external 16MHz crystals.

The display is one of the very important parts of the device. It contains 12 LED Dot Matrix Displays LM-88HR23-CC [3] glued together in a row. It has totally 96 LEDs in a row and 8 LEDs in a column.

Display controllers part includes ATmega32 microprocessor, serial-parallel convertors 74HC595, 8-bit addressable latches 74HC259 and protecting resistors the process of displaying content is described in [3].

We decided to use BTM-112 Bluetooth Class 2 Module in the device. The module uses Bluetooth 2.0 + EDR, so it might be connected with any smartphone which supports Bluetooth. It provides communication between a smartphone and ATmega162. The protocol used between ATmega162 and the module is UART.

The device is supplied by 5V DC, so there is no need to convert it to 5V logic voltage, but the Bluetooth Module needs to be supplied by 3.3V so we used LT1086CT-3.3 to change 5V to 3.3V.

We have designed a special application which is running on the paired smartphone. It is called SMS2, because the device is designed to be used for writing and reading SMS messages at most. It has been developed for Android platform, but we do not exclude developing the app also for iOS in the future. It has been written in Java programming language. It has access to many smartphone repositories and runs system functions like sending SMS messages, making calls, etc. It also encrypts and decrypts data which is transmitted between the smartphone and the device via Bluetooth.

3 Conclusion

The device has been successfully built and it is working properly. We drew a scheme, designed a Printed Circuit Board (PCB), developed software for microprocessors and smartphone and soldered electronic components to PCB. When we tested everything and everything worked correctly, we put it into a cable trough to protect electronic parts.

If people are interested in this device, we arrange to design new one with similar to this, but it would use GSM module so there would be no smartphone needed, just a SIM card. The reason, why we did not use GSM module in the device we have created is a price and higher difficulty. We hope, that this device will also appeal to some sponsors to sponsor us with creating it.

References

[1] Atmel: ATmega32. Datasheet. http://www.atmel.com/Images/doc2503.pdf

[2] Atmel: Atmega162. Datasheet. http://www.atmel.com/Images/Atmel-2513-8-bit-AVR-Microntroller-ATmega162_Datasheet.pdf

[3] LM-88HR23-CC. Datasheet. http://www.tme.eu/sk/Document/a2c7dabbed811375b571c73631668cb9/LM-88x23-Cx.pdf

[4] Writing and debugging the software. http://members.ziggo.nl/electro1/avr/scroll5.htm

Intelligent Content Management System

Miroslav ŠEDIVÝ*

Grammar School in Nové Zámky
M. R. Štefánika 16, 940 61 Nové Zámky, Slovakia
mirkosed@gmail.com

Abstract. In this paper you can find short description referring my project as a solution for the end users dealing with website creation. It includes argumentation which explains my motivation to work on the iCMS project. The project is divided into a three parts: Introduction comprising general information about the project, design description comprising design changes compared to three standard products and third part, the functionality itself, as a significant system change on website creation.

1 Introduction

I am student of Grammar School in Nové Zámky and free-time website programmer. I have been actively working with web site programing since 2012. Usually, every time I try to create a website I do not manage to do it perfectly at the first try. I constantly get ideas which bring different solutions and I am forced to make an update. This process is not easy and to avoid difficulties, I decided to make a system which would help me be more effective. Designing of such a system is the main content of the proposed project.

The project is necessary to hold up to date. If I tried to change any features on the website, I still needed to find a proper script which was responsible for the parameters. The scripts are usually located in several files and the workload to change it was still high. The system, which will make the updating process and programing much easier and faster inspired me to create the proposed iCMS (Intelligent content management system).

2 Design description

The proposed Intelligent content management system is composed of two main parts:

1. design
2. core

I am sure that in this case the design is as important as the system itself. The first impression can be done only once. First user experience with the iCMS is critical for its acceptance. I also see the importance on the fonts which will be used for communication, for example the thin font sans-serif is very modern and impressive. The icons may make the usage easier and bring faster orientation in tools which are available to developers.

[*] IIT.SRC Junior contribution
Mentor: Michal Holub, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

Another important aspect of the system is its simplicity. The user must intuitively control it and it should have friendly user interface.

The design must be adopted on the usage by mobile devices so that the traditional errors (like menu bar hiding the content or very small letters which are not readable anymore) can be easily avoided.

The big advantage compare with another systems is that iCMS offers full compatibility with all existing mobile devices and also web-browsers. No further adoptions will be needed and the optimal result achievement is very easy to get.

2.1 Extending the content with plugins

The main task is to make the changes on the website easier, so that the end user has the option to change the whole design by one click. Every part of the website is interpreted in the source code as plugin. There are different types of plugins, for example menu plugin, text plugins. Each plugin is stored on the server in the folder plugins/ and it contains PHP functions displaying particular content. The plugins are visible also in administration mode, where they can be customized.

Making the system modular using plugins is very convenient for future development. Moreover, it allows other developers to write their own plugins, and thus the main functions of the system can be extended by the community.

Each plugin function has two inputs, the first is the user preference and the second is the information about the current location.

User preferences are stored in the database with a unique ID, which also contains information about the home plugin. The fact that the presets are saved in the database allows us to easily reuse them and apply them on variety of pages without the need for making them from scratch.

In the information about the current location will get the plugin input where the user actually is located. Also, the plugin will get the information about the user, for example whether the user is logged in or not, etc. The last feature is the template.

2.2 Template – color variations

The user can chose from different color variants. However, the user is not limited by any pre-defined templates, he can define his own templates. For this system there is no need for pre-defined templates to exist, which gives unlimited options to the end user. The end user can create customized layouts with thousands of variations. This was the reason why I decided not to use any templates. The system is working on the base of grid where the user can put their plugins.

3 Conclusion

The proposed iCMS system will help the end users, companies or any professional website developers to get a user-friendly interface by creating the unlimited and creative webpages, without any restrictions and very low workload. The webpage variations can be easily done without any previous programing skills. This should allow for wide public adoption and it should make a revolution in the preprogrammed website management systems which are using templates towards the unlimited plugins programing systems with easy control and no limitations.

Programming Contest at IIT.SRC 2014

Peter TREBATICKÝ, Mária BIELIKOVÁ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
maria.bielikova@stuba.sk

Abstract. Programming contests has a long tradition at the Slovak university of Technology in Bratislava. As the student research conference offers an open day without any lectures for all our students, we are looking for ways how to attract them. Now, seventh year we have prepared an accompanying event – the programming contest for all our students.

1 Background of the Contest

Programming contests have a long tradition at our university and the faculty. From the beginning in 1998 local contests were organized for our students in order to form teams to represent the Slovak University of Technology in Bratislava at the ACM International Collegiate Programming Contest (ICPC) for Central Europe region. Since 2002 our faculty participates in organization of Czech Technical University Open, which is joint event where universities of Czech and Slovak Republic compete with the aim to select their respective representatives for ACM ICPC Central Europe region.

We prepare our students for this type of programming contest already before they enter the university. We organize for our future students the ProFIIT contest since 2004. It consists of two rounds. In the correspondence round the contestants compete in solving several (around 10) programming problems. They are allowed to compete either on their own or in pairs. The best teams advance into onsite round organized at our faculty. They compete on their own in this round as they can gain bonus points into the admission process. This year is the third time we moved the final round of ProFIIT to coincide with the IIT.SRC in order to show our potential future students exciting research opportunities awaiting them at our faculty. The main reason for this move was that many high school students have only hazy idea what are the projects they will be able to work on during their university study.

Students at our faculty can choose an elective course *Construction of Effective Algorithms* which further develops the algorithmic thinking in them and teaches them the more advanced techniques specifically usable in programming contests. We prepare four 3 hour contests during this one semester course. Participants gain bonus points in them, but these contests are not limited to course participants, everyone can compete for fun. Moreover, our bachelor students selected for the research track have more possibilities in algorithms training, mainly in seminar on advanced algorithms.

[*] Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

2 Structure of Programming Contest at IIT.SRC

The structure of the programming contest at IIT.SRC closely copies the structure of ACM ICPC. Contestants compete on their own onsite in our computer labs. They have two hours to solve four problems. Problems contain a basic description of what should be solved, exactly specify the format of textual input to the program as well as the format of output and the end of problem statement is the sample input and corresponding sample output.

The task of contestants is to create a program in either C/C++ or Pascal that transforms test input, which has described format but is unknown to contestant, into correct output according to problem statement and in correct format. They submit the source code through our system for programming contests, which compiles the code, runs it against test input, evaluates the given output and informs the contestant of the result. Result is only in the form of simple statement, e.g. “Accepted”, “Wrong answer” or “Presentation error” which means the output is not formatted correctly but otherwise appears to have given the correct answer.

The order of contestants is primarily determined by the number of solved problems and in the case of tie, by the sum of the times taken to solve each problem since the beginning of the contest. There is also a 10 minutes penalty for each submitted incorrect solution, but only for the eventually solved problems. This type of order determination favors of course primarily those who solve more problems, but secondarily those who first solve easier problems and also those with lower number of incorrect submissions. The ability to decide fast which problem is the easiest one and to create solution without bugs is also very important apart from the ability to come up with working idea. These skills are mainly trained by practice and learning that is where we help the students through activities mentioned here.

The contest is made more attractive for participants by the fact that during last 45 minutes the preliminary results are not updated. This way, one cannot be sure about her final standing until the awards ceremony. The time interval of not displaying preliminary results was chosen in accordance with conference schedule, because there is another contest ending right before the second poster presentations in which the other conference attendants can tip the winner.

More information about our programming contests can be found on the Web:

- ACM programming contest – `http://www.fiit.stuba.sk/acm/`
- ProFIIT programming contest – `http://profiit.fiit.stuba.sk/`

FIITAPIXEL Exhibition at IIT.SRC 2014

Pavol NÁVRAT, Mária BIELIKOVÁ, Ján LANG*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
{name.surname}@stuba.sk

Abstract. FIITAPIXEL is an initiative of the Faculty of Informatics and Information Technologies that brings together its members (both students and staff) as well as its potential students and alumni in an effort to create, share and judge pictures. It is organized as an ongoing event, where anyone can contribute pictures to certain categories of photographs. The submitted photographs take part in a contest that is organized annually. Besides best photographs, also best photographers are announced based on their success with their photos. The contest has an expert panel of jurors who give their lists of best photos in each category. In parallel, visitors vote for any photo they like and their votes are counted to result in list of best photos according to popular voting. For the fifth time we organized at the IIT.SRC an exhibition of the best pictures this year contest.

1 FIITAPIXEL as an inspiration

FIITAPIXEL is an initiative of the Faculty of Informatics and Information Technologies to contribute in providing to its members, students and staff alike, an inspiring, creative, stimulating environment to study or to work in. Studying is mostly demanding and hard, and so is working at an institution which faces such a level of competition as is the case in the higher education sector in informatics and information technologies related fields in this region of Europe. From Budapest to Prague, from Vienna to Brno, in a relatively close proximity of Bratislava there several respected institutions with a similar scope of interest. Moreover, in the city itself, there are several other competing institutions.

We try to offer something that may make a little difference. By providing a platform and other forms of support, the Faculty creates an environment that allows expressing its members in a completely different way as it is usual in their professional work. Instead of writing programs or designing chips, they get a chance to express themselves by way of pictures. The language of pictures is intended as a language of artistic expression, even when respecting all the limitations given by the simple fact that these professionals in one (informatics related) field are complete amateurs in another (photography) and similar limitations apply when e.g. elements of journalism are involved.

* Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2014, Bratislava, April 29, 2014, pp. 643–644.

2 FIITAPIXEL Organization

FIITAPIXEL started in 2009 and it has been organized ever since then. It takes place as a contest organized annually. The final results are usually announced and prizes awarded around the time of our student research conference IIT.SRC. Immediately after one year of the contest is closed, themes for the next one are published and the contest is open again. The contest is organized in two legs during one year that last approximately half a year each.

There are usually four themes open for each particular leg, but some of them may be adopted for the next period. For example, in academic year 2013/14 contest there were these four themes for the first leg (Summer and Autumn):

- *Enchantment of tininess*
- *Images from the street*
- *Something's wrong, something has changed here*
- *The place where I am right now*

with *Colourful nature* replacing the third one for the second leg (Winter and Spring).

Each participant can submit up to five pictures to each category both in the first and the second legs. These up to 40 pictures are published on the contest portal, where they are freely visible from anywhere in the world. Anyone can express her/his likes which are treated as votes for the particular picture. At the end of each period, votes are simply counted and the best dozen pictures are announced as winners, according to a popular vote, in each category.

There is also an expert jury formed by experts in visual arts which gives its opinion resulting in another set of lists of dozen winning photos in each category. Results of both opinions, expert and popular, are then used to determine a list of best photographers based on how their photos are placed in particular results.

In the 2013/2014 contest, we have had 947 pictures taken by 132 authors. They received nearly 2 500 votes from visitors. Pictures and wining photos are available on the contest portal: `http://foto.fiit.stuba.sk`.

3 IIT.SRC Exhibition

Annual evaluation of the best photographers of the FIITAPIXEL Contest takes place at the student research conference award ceremony. Moreover, we give conference participants the opportunity to enjoy an exhibition of the winning photos of each category in both legs, i.e. we exhibit two dozens of winning pictures, in 2014 in nice new building of the Faculty. IIT.SRC participants can cast their vote for the best photo during the conference. At the end of the day, winner of the participants'vote is announced and awarded.

FIITAPIXEL brings new dimension into our living space at the Faculty together with much inspiration for our activities. The selected best photos will decorate our environment in new building.

RoboCup Presentation at IIT.SRC 2014

Ivan KAPUSTÍK, Pavol NÁVRAT [*]

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
{kapustik, navrat}@fiit.stuba.sk

Abstract. RoboCup is an attractive project theme with a free participation, designed to support education and research in artificial intelligence, robotics and information technologies. During the last few years, our students achieved some interesting results, which were presented during our student research conference.

1 Motivation

RoboCup is an international joint project to promote research in artificial intelligence, robotics and information technologies. It is an attempt to advance artificial intelligence and intelligent robotics study and research by providing a well-known and attractive problem where wide range of technologies can be integrated and examined. RoboCup chose to use soccer game as a central topic of research. The ultimate goal of the RoboCup project is to develop by 2050 a team of fully autonomous humanoid robots that can win against the current human world champion team in soccer.

In order for a robot team to actually perform a soccer game, various technologies must be incorporated, including design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, multi-level decision making, robotics and sensor-fusion. RoboCup is a task for a team of multiple fast-moving and skilled robots within a dynamic environment. It offers also a software platform for research on the software aspects. RoboCup is divided into four main fields: RoboCup Soccer – defined by the original domain of soccer, RoboCup Rescue – intended to do search and rescue in large scale disaster area, RoboCup Junior – aimed to child education and motivation and RoboCup @Home – oriented to provide various help not only at home.

From our point of view, the main goal of RoboCup is to promote research in areas of artificial intelligence and information technologies, especially in the area of multi-agent systems. This is a benefit for the students, making their studies more interesting and attractive. Students can meet with robotic soccer in courses like Artificial Intelligence, Team Project and others. Students are facing an interesting problem, which demands invention as well as use of modern artificial intelligence approaches. Teams of students have the possibility to directly compare their results in tournaments. This encourages the students to even higher effort and motivates them for better results. More fundamentally, achieving progress requires tackling

[*] Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2012, Bratislava, April 25, 2012, pp. 645–646.

serious open research problems in artificial intelligence, such as planning of cooperation of multiple agents etc. That is why this area is of interest also for our doctorate students.

We have been organizing this tournament regularly for several years. Starting as a local event in 2000, it has grown to a regional contest under the official RoboCup authorization. Our Faculty organizes tournaments in the simulated category only, but we gradually include other categories. Our current contest event has three parts.

First part is a tournament of two-dimensional (2D) simulated player teams, where students try to make their own players win soccer game. 2D players are simple entities, ready to follow any possible action in their virtual environment. Students' main research is aimed to team tactics and autonomous player decision. It includes team formations and planning, player communication, use of a team coach and decision skill improvement. Methods here cover planning and player's action selection based on diverse sources – success evaluation of similar situation, teammate decision model and prediction of opponent behaviour. This contest part is currently more an exhibition of new approaches than a tournament, because interest of our students shifted to more complex three-dimensional (3D) robotic simulation.

Second and third parts of this tournament involve three-dimensional (3D) robotic simulation. These robots are true copies of their real master. They have limbs and joints. Primary students' task was to teach robots to reliably walk, turn, stand up and kick the ball. It was followed by design of a proper composition of these basic skills to achieve simple goals, like walking to the best game position or getting the ball. Then, the training support framework has been developed and test modules for robot learning were created. This academic year, new layered software architecture was designed. It facilitates separation of different robot skills into few layers – simple skills, complex skills, tactical decisions and strategic planning. After some code refactoring, students' effort is currently oriented to full development of layers for complex skills and tactic. They want to end this term with robots trained to compose adequate complex skills from simple ones in accordance to tactic needs.

Any soccer player must be good with physical skills and must make good and fast decisions during the game. So the second part of our tournament contains skills match. Robots compete in speed and accuracy of given tasks. They can get a few points for "unusual" useful skills as well. Finally, third and most valued part of this tournament holds soccer contest, where both skills and decision making are verified in real-time game.

2 Results presentation

For this student conference we decided to hold an exhibition of results achieved in 3D soccer simulation. Two student groups work on code refactoring, complex skills development and decision making for 3D soccer robotic players. Both groups presented details of their own ideas and methods. These methods involve composition based on annotation, inter-layer communication, simple skill control, complex skill handling, soccer situation recognition, situation based decision, tactical planning and others.

Presentations were enhanced by show of robot skills performance. Our students improved some of old movement sequences and added few complex movements. New skills included mainly optimised movement to a chosen place, faster robot orientation and better work with ball. Improved player actions were also more attractive for audience.

The extension of the soccer game simulation to the third dimension shows the continuous progress in RoboCup and in our students' skills, too. Decision making of these robots is very complex and brings new challenge to everyone concerned. We hope that exhibition of robotic simulation will attract many present and future students and give them motivation for their study and research work.

More information about our annual tournament can be found on the web page `http://www.fiit.stuba.sk/robocup/`.

Index

Mária Bieliková (Ed.)

IIT.SRC 2014: Student Research Conference in
Informatics and Information Technologies

1st Edition, Published by
Slovak University of Technology in Bratislava

670 pages, Available at http://iit-src.stuba.sk/
Print Nakladateľstvo STU Bratislava
2014

ISBN 978-80-227-4153-8

ISBN 978-80-227-4153-8

9 788022 741538