

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA  
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES  
INSTITUTE OF INFORMATICS AND SOFTWARE ENGINEERING

Study program: Programme Systems  
Field of study: 9.2.5 Software Engineering

Michal Tvarožek

# Exploratory Search in the Adaptive Social Semantic Web

DOCTORAL DISSERTATION

FIIT-10890-17567

Supervisor: Prof. Mária Bielíková

Bratislava, December 2010



# Exploratory Search in the Adaptive Social Semantic Web

Michal Tvarožek

Supervisor: **Prof. Mária Bieliková**

Reviewers: **Prof. Jozef Kelemen**  
**Dr. Michal Laclavík**

Date: **December 15, 2010**

Keywords: Semantic Web, exploratory search, faceted navigation, personalization, user modelling, user interface generation, graph visualization

ACM Subject Classification:

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces

H.5.4 [Information interfaces and presentation (e.g., HCI)]: Hypertext/Hypermedia



# Annotation

---

Slovak University of Technology in Bratislava  
Faculty of Informatics and Information Technologies  
Field of study: Software engineering  
Study program: Programme systems

Author: Michal Tvarožek  
Thesis: Exploratory Search in the Adaptive Social Semantic Web  
Supervisor: Prof. Mária Bielíková  
December, 2010

The thesis addresses the crucial issue of effective access to information resources on the Web with specific focus on aiding exploratory user experience in the Semantic Web environment. We identify end-user related issues that arise in present day information retrieval systems (e.g., difficult query construction/modification, information overload, limited exploration options/guidance) and analyze their impact in the Semantic Web environment.

A review of the current state of the art in the multidisciplinary exploratory search field is given with special focus on faceted browsing approaches, personalization, navigation and visualization approaches, and web information retrieval support systems. We present an extension and novel combination of existing approaches in these fields with the global aim of improving end-user experience in the Semantic Web environment.

The proposed approach extends faceted browsing with dynamic facet generation from ontological metadata with personalization and adaptive view generation based on estimated user preferences acquired from implicit user feedback. The issues originally identified are addressed by visual query construction via the faceted browser, personalized recommendation (of facets, views and content) and multi-paradigm exploration allowing the user to dynamically switch between searching via keywords/views/content and browsing the result views or individual results.

We discuss the extension of the existing web browser concept based on the proposed principles into a next generation web browser, with integrated support for both legacy and semantic web content. Facets and personalization are described as integral parts of the web browser rather than being a part of server-side applications as is the case today, effectively addressing issues of privacy, trust and widespread availability.

The thesis presents the evaluation of the proposed approaches in several application domains (job offers, digital images and scientific publications) with promising results. The partial approaches are evaluated quantitatively and qualitatively both separately and as a whole while comparing their individual usefulness and user acceptance.



# Anotácia

---

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
Študijný odbor: Softvérové inžinierstvo  
Študijný program: Programové systémy

Autor: Ing. Michal Tvarožek

Téma: Prieskumné vyhľadávanie v adaptívnom sociálnom webe so sémantikou

Školiteľ: prof. Ing. Mária Bieliková, PhD.

December, 2010

Dizertácia sa zaoberá kľúčovou otázkou efektívneho prístupu k informačným zdrojom na webe so špecifickým zameraním na podporu prieskumného vyhľadávania na webe so sémantikou. Identifikujeme problémy s ktorými sa stretávajú koncoví používatelia súčasných webových vyhľadávacích systémov (napr. zložitá tvorba a zmena dopytov so sémantikou, preťaženie informáciami, obmedzené možnosti prieskumného prehliadania) a analyzujeme ich dopad v prostredí webu so sémantikou.

V práci rozoberáme aktuálny stav v multidisciplinárnej oblasti prieskumného vyhľadávania so zameraním sa na prístupy pre fazetové prehliadanie, personalizáciu, navigáciu a vizualizáciu informácií, a na podporné prístupy pre vyhľadávanie informácií. Prezentujeme rozšírenie a originálnu kombináciu existujúcich prístupov v oblasti s cieľom zlepšiť prácu koncových používateľov v priestore webu so sémantikou.

Navrhnutý prístup rozširuje fazetové prehliadanie o generovanie faziet na základe ontologických metadát, personalizáciu a adaptívne generovanie pohľadov na základe odhadovaných charakteristík používateľa odvodených z implicitnej spätnej väzby. Pôvodne identifikované problémy riešime pomocou vizuálnej tvorby dopytov vo fazetovom prehliadači, personalizovaného odporúčania (faziet, pohľadov, obsahu) a viac-prístupového prieskumného prehliadania, ktoré používateľovi umožňuje dynamicky prechádzať medzi vyhľadávaním pomocou kľúčových slov, pohľadov, obsahu ako aj prehliadaním výsledkov.

Diskutujeme tiež možné rozšírenia existujúcich webových prehliadačov s využitím navrhnutých princípov do podoby prehliadača webu novej generácie s integrovanou podporou prehliadania tak klasického obsahu webu ako aj obsahu webu so sémantikou. V tomto prípade by fazety a personalizácia predstavovali základné vlastnosti prehliadača, namiesto ich súčasnej pozície ako funkcie webových aplikácií na strane servera, čo by umožnilo efektívne riešiť problém súkromia, dôvery a širokej dostupnosti.

Navrhnuté prístupy sme overovali vo viacerých aplikačných doménach (fotky, pracovné ponuky, publikácie) s perspektívnymi výsledkami v rámci viacerých výskumných projektov realizovaných na FIIT STU. Čiastkové prístupy sme vyhodnotili tak kvantitatívne ako aj kvalitatívne z pohľadu ich použiteľnosti pre koncových používateľov.





## About the Author

---

Michal Tvarožek is a doctoral candidate in Programme systems at the Slovak University of Technology in Bratislava. He holds a Bachelor's degree in Informatics (2005) and a Master's degree in Software Engineering (2007) from the same university. His research interests are in the areas of exploratory information retrieval, adaptive user interfaces, and personalized web-based systems and user modelling. In line with his research interests, Michal Tvarožek has worked as a researcher in several research projects conducted at the Slovak University of Technology whose focus was effective use of metadata and personalization to improve acquisition, search and exploration of information on both the Web and the Semantic Web.



He has published his research findings in journals and presented his work at several conferences including some supported by ACM, IEEE and IFIP. He received the Werner von Siemens Excellence Award 2007 for his diploma thesis and ranked as the 2<sup>nd</sup> place winner in the ACM SRC Grand Finals 2010 with his work on “Personalized Semantic Web exploration based on adaptive faceted browsing” presented at the Hypertext 2009 conference. He received the title “Student Figure of the Year 2008/2009” in the field of mathematics, physics and informatics from Junior Chamber International endorsed by the President of the Slovak republic. He also acquired a studentship grant from Intenda aimed to support finishing doctoral candidates in their last year of study and became a member of the Club of individualities endorsed by the Intenda grant agency. He is a member of the Slovak Society for Computer Science and ACM.



## Acknowledgements

The work on this dissertation has been a long and tedious one. It was a path lined with great expectations, hard work during long evenings and early mornings, many pleasant encounters all over the world and lastly outstanding achievements. I wish to thank all who accompanied me during this time, helped me stay motivated and always pointed me into the right direction.

Specifically, I want to thank my supervisor, Prof. Mária Bielíková, who helped me nurture and develop ideas, supported me during the more difficult times and kept me on track towards the final submission. I wish to thank my colleague Michal Barla, who I collaborated with on several research projects during my doctoral study and who has been a great initiator and critic of our common work. I also wish to thank people at our faculty and institute who by taking care of us and by acquiring research funding allowed me to do my research, publish my results and enjoy the pleasant encounters at conferences where I presented my results and represented our faculty.

Last but not least, I would like to thank my family and close friends who supported me during all this time and allowed me to keep focused on my work by relieving me from many daily tasks and cheering me up when things got dire.



# Table of contents

---

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	INFORMATION AS THE SOURCE OF CIVILIZATION ADVANCEMENT .....	1
1.2	THE WEB AS THE FOUNDATION OF THE INFORMATION AGE .....	2
1.3	SEMANTICS IN THE PRESENT WEB .....	3
1.4	GOALS AND OUTLINE .....	5
<b>2</b>	<b>CHALLENGES IN (SEMANTIC) WEB EXPLORATION .....</b>	<b>9</b>
2.1	CHALLENGES IN LEGACY WEB SEARCH AND NAVIGATION.....	10
2.2	CHALLENGES IN SEMANTIC WEB SEARCH AND NAVIGATION .....	14
<b>3</b>	<b>STATE OF THE ART IN EXPLORATORY SEARCH .....</b>	<b>19</b>
3.1	FACETED BROWSING .....	19
3.2	QUERY CONSTRUCTION AND SEARCH .....	22
3.3	NAVIGATION AND VISUALIZATION .....	25
3.4	REVIEW OF SELECTED EXISTING APPROACHES .....	28
3.5	SUMMARY OF CURRENT EXPLORATION APPROACHES .....	43
<b>4</b>	<b>FRAMEWORK FOR EXPLORATORY SEARCH .....</b>	<b>47</b>
4.1	EXAMPLE USER SCENARIO .....	47
4.2	DESIGN OBJECTIVES AND MAIN PRINCIPLES .....	48
4.3	SEMANTIC INFORMATION SPACE REPRESENTATION .....	49
4.4	MULTI-PARADIGM EXPLORATION .....	50
4.5	FACETED BROWSER EXTENSIONS .....	53
4.6	VALIDATION OVERVIEW .....	54
<b>5</b>	<b>PERSONALIZED RECOMMENDATION .....</b>	<b>61</b>
5.1	USER CHARACTERISTICS MODEL.....	62
5.2	MODEL FOR RELEVANCE EVALUATION .....	63
5.3	PERSONALIZATION METHOD OVERVIEW .....	66
5.4	FACET AND RESTRICTION RECOMMENDATION.....	67
5.5	SEARCH RESULT RECOMMENDATION .....	68
5.6	DISCUSSION AND EVALUATION .....	69
<b>6</b>	<b>ADAPTIVE VIEW GENERATION .....</b>	<b>75</b>
6.1	FACET GENERATION .....	75
6.2	DISCUSSION AND EVALUATION .....	80

<b>7</b>	<b>MULTI-PARADIGM EXPLORATION .....</b>	<b>83</b>
7.1	SEARCHING AND BROWSING .....	84
7.2	RESULT EXPLORATION .....	89
7.3	REVISITATION AND ORIENTATION SUPPORT .....	93
7.4	DISCUSSION AND EVALUATION .....	98
<b>8</b>	<b>CONCLUSIONS .....</b>	<b>101</b>
8.1	MULTI-PARADIGM EXPLORATION SUMMARY .....	101
8.2	CONTRIBUTION .....	102
8.3	DISCUSSION .....	103
<b>9</b>	<b>LOOKING AHEAD .....</b>	<b>105</b>
9.1	NEXT GENERATION EXPLORATORY (WEB) BROWSER .....	105
9.2	INTERACTIVE CONTENT EXPLORATION .....	106
	<b>REFERENCES .....</b>	<b>109</b>
 <b>APPENDIX A DISSERTATION OUTCOMES</b>		
A.1	LIST OF PROJECTS PARTICIPATED IN	
A.2	LIST OF AWARDS RECEIVED	
A.3	LIST OF PUBLICATIONS	
A.4	LIST OF CITATIONS	
 <b>APPENDIX B EVALUATION ENVIRONMENT</b>		
B.1	FIRST FACTIC PROTOTYPE (NAZOU, MAPEKUS)	
B.2	SECOND FACETED BROWSER PROTOTYPE	
 <b>APPENDIX C ONTOLOGICAL MODELS AND DATASETS</b>		
C.1	PROJECT NAZOU: JOB OFFERS DOMAIN MODEL	
C.2	PROJECT MAPEKUS: SCIENTIFIC PUBLICATIONS DOMAIN MODEL	
C.3	PROJECTS NAZOU, MAPEKUS: USER MODEL	
C.4	PROJECT NAZOU, MAPEKUS: USER LOG MODEL	
C.5	DIGITAL IMAGE DOMAIN MODEL	

# 1 Introduction

---

## 1.1 Information as the source of civilization advancement

The relationship between economic growth and various kinds of advancements has occupied the minds of historians for a long time. Many popular histories ascribe the (economic) growth that the human society has experienced to various technological advancements such as the development of the printing press, industrial revolution, the steam engine or even more cultural advancements such as renaissance or reformation. Still, some authors explore the relationship between *information access and processing capabilities* and the very same civilization advancement, and rightfully ask the question of causality. What were the “true” causes of advancement? Was the Industrial revolution truly the cause or rather merely the consequence of deeper changes occurring within the human society at the time?

In his article, Michael Bergman (Information is the Basis for Economic Growth, 2007) explores the relationship between key technological advancements and the economic wealth based on current global average per capita income (GDP) estimates from AD 1 onwards. To summarize, the economic growth was relatively flat (even declining) before 1000 AD when it changed into a continuous slightly upward trend albeit with several minor transient inflection points. This change somewhat corresponds with the introduction of raw linen paper around 1000 AD, which made using skins for writing and information transfer obsolete.

However, the first major inflection point in GDP estimates, and thus the first major change happened in the early 1800s, roughly corresponding to the Industrial revolution, ignoring previous advancements such as the printing press, renaissance or reformation. The historically flat income averages suddenly rose sharply indicating that something huge did in fact happen in the early 1800s. While historically this advancement was accredited to machines and industrialization, these may have been the result and not the cause of the change. By the early 1800s, advances in printing presses and paper production lead to mass media, which could bring *cheap* information to the masses as opposed to the original printing press, which while providing significant advantages could only be used by a select few of the wealthy who could afford books.

Additional increases in growth occurred in the early 20<sup>th</sup> century and in the post World War II era, which might be ascribed to electrification, and early electromechanical and electronic computers resulting from the war effort respectively further improving information processing and availability.

The conclusion Michael Bergman draws is that *information has been the source of economic growth* as “information technology” advances allowed ever more people to effectively *learn, share and innovate*: “...all prior discovered information across the entire species can now be accumulated and passed on to subsequent generations. Our storehouse

of available information is thus accreting in an exponential way, and available to all. These factors make the fitness of our species a truly quantum shift from all prior biological beings, including early humans.”

At the end of his article, Michael Bergman describes the Internet as “the latest example of such innovations in the infrastructural groundings of information”, which has the tremendous potential to allow *every individual to access, contribute to and take advantage of information*.

## 1.2 The Web as the foundation of the Information age

While Michael Bergman describes the Internet as the latest innovation in the infrastructural groundings of information, we believe that it is in fact the Web that allowed unprecedented access to information on a global scale. Since its inception in 1989 at CERN, Switzerland, the Web has grown to become a *global ubiquitous socioeconomic space* that is used by both private users and businesses alike.

The present information technology domain encompasses the access to, processing, organization and visualization of information together with the corresponding software and hardware infrastructure. Information technology has already become a vital and indispensable part of daily life in (developed) countries, shifting focus from the manufacturing of physical goods towards the creation, organization and sharing of information, thus giving rise to the *Information age*.

In this respect, the Internet and the Web play a principal role in information access and processing by providing a communication environment where information can be published, shared and accessed by everyone – anytime and anyplace. Already in 2001 about 10% of the world’s population or 550 million users had access to the Web. In 2005, the worldwide number of Internet users surpassed 1 billion with above 65% penetration in developed countries and 10-20% penetration in developing countries<sup>1</sup>. As of May 2010, the estimate was about 1.8 billion Internet users with a global penetration of 26.6% of the world population (Miniwatts Marketing Group, 2010). The online population *effectively tripled in less than 10 years time*, while already every fourth person on the planet has access to the Web as a global shared information medium unprecedentedly surpassing any and all of the previous information media.

The enormous growth, dynamics and diversity of the Web resulted in several issues such as ineffective information retrieval, orientation problems and information overload, which pose new research challenges. In practice, these turn into the need for novel search and navigation approaches and better information retrieval support approaches. Furthermore, the evolution of the Web in terms of novel usage means and user expectations such as exploratory search (Marchionini, 2006), social networks, interactive

---

<sup>1</sup> Computer Industry Almanac Inc. Press release: Worldwide Internet Users Top 1 Billion in 2005, <http://www.c-i-a.com/pr0106.htm>



applications or user created content together with the impact of the Web on the human society resulted in the emergence of Web Science as a new research field (Hendler, Shadbolt, Hall, Berners-Lee, & Weitzner, 2008).

Similarly, current web initiatives attempt to address the aforementioned issues in their respective fields (e.g., information retrieval, information visualization, human-computer interaction), requiring novel and ever improving methods for quick and easy information retrieval, presentation and understanding:

- *Semantic Web* aims for a machine readable representation of the Web with support for semantics and reasoning (Shadbolt, Berners-Lee, & Hall, 2006),
- *Adaptive Web* stresses the needs of individual users via personalization and user adaptation (Brusilovsky, Adaptive Hypermedia, 2001),
- *Web 2.0* or Social Web focuses on interaction, social aspects, collaboration of individual users and the sharing of content (O'Reilly, 2005).

Although each of these initiatives addresses different problems and aspects of the Web's development, if successfully combined together they are likely to produce synergetic effects, which are already starting to surface. Ultimately, their combination would transform the current Web as we see it today into an entirely new and mature Web of tomorrow.

Based on our research and these foundations, we believe that efficient information access, processing and usage are key prerequisites for sustained economic growth and advancement of the human society and civilization, thus requiring conscious persistent research and development effort.

### 1.3 Semantics in the present Web

A key aspect of the present Web is the increasing amount of semantics put into resources, be it multimedia (photographs, audio or video files), data repositories or simple web pages. Oftentimes, the goal is simply to provide more information, however the aim to simplify usage, discovery or sharing of information is becoming ever more important with increasing competition between service providers.

It is important to note that, so far there is no common concept of what exactly semantics on the Web are. For example, simple *tagging of web sites* on a social bookmarking site can be considered a form of user supplied semantics sensible either for the specific user or possibly useful for a broader range of users. The emergence of *folksonomies* based on user tags and their successive use for annotation of web resources is a somewhat stronger form of shared semantics between multiple users. Similarly, using *microformats*<sup>2</sup> or other agreed upon notations (e.g., hCard, hCalendar) to tag information on the Web can provide additional semantic metadata about resources. All these user-driven approaches correspond to so called *lightweight semantics* and usually represent small

---

<sup>2</sup> Microformats: <http://microformats.org/>

compact pieces of information distributed on the Web (e.g., contact information, calendar information). They are, for the most part, not strictly regulated nor formally represented in a predefined (managed) framework.

Another direction of research focuses on the *description of strong semantics via ontologies*<sup>3</sup> in the Semantic Web environment. W3C<sup>4</sup> formally defines standards for ontology representation such as RDF, RDFS and OWL, and related query languages such as SPARQL. Although the initial promise of wide-spread semantically described information on the Web has yet to be realized (Shadbolt, Berners-Lee, & Hall, 2006), there are already many significant sources of semantically enriched information on the Web in the form of *Linked Data*<sup>5</sup>.

While the availability of semantically enriched metadata on the Web is still far from satisfactory, the currently available amount of semantics on the Web has, in our view, already reached the critical mass necessary to provide added value to end-users.

The semantics on the Web form an additional layer of metadata describing the resources already present on the Web. Semantic Web ontologies go one step further and allow us to describe web resources (e.g., a web page or photo), abstract concepts (e.g., love) or real-world objects (e.g., people, buildings) which need not have a corresponding representation on the Web (see Figure 1). Thus the Semantic Web is a web of abstract resources and their semantically described relations and attributes, with optional references to existing resources on the legacy Web.

The semantics of links between legacy Web resources could be best described as *relatedTo*, although in many cases even this could be disputed (e.g., a link to Google). On the contrary, semantic metadata can provide detailed information on resource relations, e.g. *hasSupervisor* between a student and a teacher. These semantics can in turn be used by search and navigation approaches to offer better services to end-users.

Thus, the challenge is to take advantage of semantic metadata available in the Semantic Web in order to *devise novel metadata-based approaches for search, navigation and exploration* that would address the original problems of the Web (e.g., information overload, the navigation problem). We focus on the use of *strong semantics via ontologies* since we believe they can provide more added value to end-users as opposed to lightweight semantics approaches. We also outline the possibilities of including lightweight semantics in our approach (see chapter 9), where we propose means for automatic acquisition of semantic metadata from legacy content for the population of ontologies with strong semantics.

---

<sup>3</sup> Ontology is formally defined as an *explicit formal specification of a shared conceptualization* (Studer, Benjamins, & Fensel, 1998), or informally as a set of concepts and their relations.

<sup>4</sup> World Wide Web Consortium (W3C) is the main organization shaping the development of the Web.

<sup>5</sup> The Linked Data initiative aims to semantically link distributed related information on the Web using URIs and RDF, <http://linkeddata.org/>

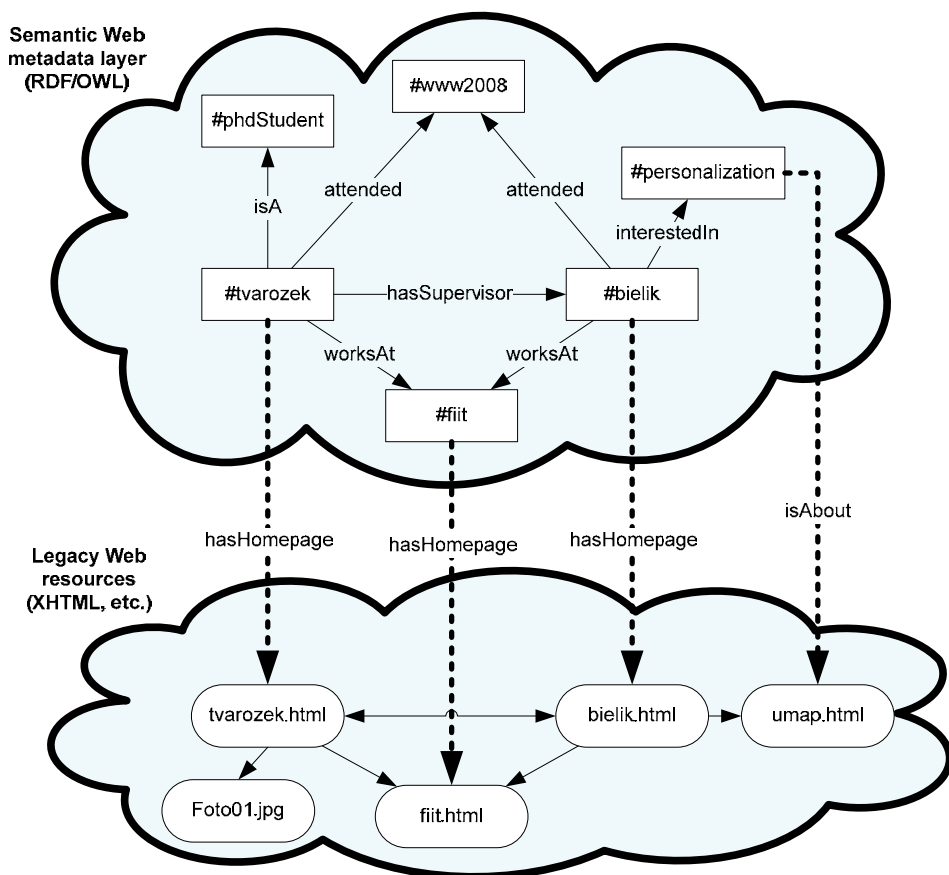


Figure 1. The Semantic Web metadata layer (top) overlays the resources already present in the legacy Web (bottom) and provides additional semantics to their relations.

## 1.4 Goals and outline

Our high-level goal is to *improve and maintain the usefulness of the evolving Web as a global information space*, and thus to provide end-users with effective access to information on the Web including both Semantic Web and Legacy Web resources. Our underlying research goal is to *devise an end-user grade exploratory search approach* for seamless exploration of both semantic and legacy web content with specific focus on user guidance, orientation support, intuitive visualization and personalization for individual users.

The major issues that presently affect end-user exploratory search experience on the Web are analyzed in chapter 2 and include complex query construction, query ambiguity, information overload and the underlying problems with orientation and understanding of complex information spaces. These issues also transfer into the Semantic Web

environment where new issues arise, such as the lack of default visualization of semantic resources and the intricacy of semantic queries.

In chapter 3, we explore the current (Semantic) Web exploration options which are mostly limited to query builders and table-based browsers of metadata. These however are often not user friendly and offer only limited exploration capabilities to end-users. More advanced approaches include the use of faceted browsers for specific information domains, such as mSpace for music collections, but offer support for neither personalization nor user interface generation. Specifically, *effective end-user grade construction of semantic queries and exploration and visualization of search results is presently a major problem.*

We take advantage of the aforementioned web initiatives, combine their respective approaches into a *highly interdisciplinary solution* and in chapter 4 present our framework for exploratory search based on:

- *Faceted browsing model extensions* – effective and expressive query construction via new facets and facet types, usability and presentation improvement via suitable visualization and adaptation.
- *(Dynamic) facet generation* from metadata accommodating for various data types with optional domain specific enhancements.
- *Integration of several search paradigms* including keyword-based search, view-based search, content-based search and social search.
- *Integration with visual presentation and navigation approaches*, based on graphs constructed from concept relationships or hierarchical clusters.

Since our ultimate goal is to *seamlessly merge both legacy Web and Semantic Web exploration*, in our current work we *focus on Semantic Web exploration based on strong semantics* (i.e., RDF(S)/OWL ontologies) with the assumption that ontologies have enough expressivity to capture all required semantics (description logics used by ontologies have high expressive power), and that other semantics can eventually be transformed into ontologies.

We provide details of the individual aspects of our approach in chapters 5, 6 and 7, where we describe means for personalized recommendation of facets, restrictions and search results, adaptive view generation with focus on facets and results overviews, and multi-paradigm exploration integrating different approaches to search, navigation and visualization respectively.

Since exact analytical evaluation of (personalized) exploratory search approaches is not possible, also due to the general immaturity of evaluation methodologies (Kules, Capra, Banta, & Sierra, 2009), we validate our approach via a software prototype of our exploratory search approach and evaluate its operation via user studies and proof of concept validation in these application domains:

- digital images,
- job offers,
- scientific publications.

During evaluation, we focused both on quantitative and qualitative aspects of the proposed approach both with respect to individual proposed methods and to the overall evaluation of the whole solution and its usefulness. Our evaluation results are presented along with the corresponding methods in chapters 5, 6 and 7 respectively.

We discuss the exploratory search improvements provided by our approach in chapter 8 where we also summarize our contributions to *end-user grade exploration of the Adaptive Social Semantic Web*. Furthermore, we have already identified several extensions of our approach which due to the scope of this thesis were left for future work. We outline these extensions with focus on legacy and Semantic Web exploration integration and interactive content exploration in chapter 9.

A major part of the presented results was achieved within several projects conducted both individually by the author and in research teams at the Institute of Informatics and Software Engineering (see Appendix A for a complete list of projects). These projects also strongly influenced the two realized software prototypes which were a major part of projects NAZOU and MAPEKUS (first prototype), and PeWePro (second prototype) respectively, as they had to fit in line with the overall software evaluation framework in these projects (see Appendix B for a description of the evaluation environment). The projects also determined the selected evaluation domains, as NAZOU was conducted in the job offers domain and MAPEKUS was conducted in the scientific publications domain (see Appendix C for a description of the used domain and user ontologies).



## 2 Challenges in (Semantic) Web Exploration

---

The Web serves both as a global ubiquitous repository of information and a shared social environment for communication and interaction between businesses and private individuals alike. The evolution of the Web into a global socioeconomic space resulted in several major issues that must be continuously addressed to maintain its usefulness for society:

- *Information availability* – the issue of making information technically available somewhere in the surface/deep web also available to end-users. I.e. how to find and index all available data, how to make it searchable, how to provide users with access to it. E.g., the required information might be available in some deep web database, but there is no practical way how an interested user would access it let alone know it was there.
- *Information overload* – the issue of having too much information available and making sense of it, understanding it and selecting the relevant portion of it. E.g., too many results in a search engine, too many products to choose from in a web shop, too many articles to read on a news page.
- *The navigation problem* – the “lost in hyperspace problem” where users lose track of what sites they were browsing, how they got there and how they should return back due to the complexity of the web information space. E.g., a user browsing a large corporate web site looking for some form suddenly finds himself lost as he was unable to find the page with the form for download and has no clue what to do next.
- *User diversity* – includes different preferences, expectation and requirements on tools, cultural differences, age differences and various means used to access information. E.g., older users might prefer larger fonts, young users might prefer more sophisticated tools.

Users typically access the Web as an information repository to satisfy their individual needs, which according to Broder can be classified into three categories (Broder, 2002):

- *Informational*, when users seek specific information (e.g., what is the Web?).
- *Navigational*, when users seek a starting point for further exploration of information on the Web (e.g., what is a good web site about Beethoven?).
- *Transactional*, when users wish to use a service often provided by a web site (e.g., buy a digital camera).

In practice, the aforementioned major issues are “just” consequences of the lack of support for the four primary actions that users iterate through during typical web search sessions to satisfy their needs (Levene & Wheeldon, 2004):

- 1) *Query*, where users submit a search query describing their respective need.

- 2) *Selection*, where users select one of the returned links (search results) and browse the web page displayed after following the link.
- 3) *Navigation*, when users start a navigation session, which involves the browsing of web pages and the following of highlighted links. This step also involves the *exploration and understanding of web page content*.
- 4) *Query modification*, which occurs when users interrupt a navigation session, e.g. due to unsatisfactory results, and decide to update the original search query and resubmit it to the search engine. The users then continue with step 2.

Alternatively, users may know the destination web site through other means and directly choose a URL address, in which case they iterate over step 3 until their need is satisfied or they decide to use a search engine (step 1). For example, the destination site can be selected from bookmarks, browsing history or passed to them via an email from a friend.

While typical search engines such as Google or Bing are prime examples of information retrieval systems, from an end-user perspective, finding relevant information is but the first step towards satisfying user needs. The second step concerns the finding and understanding of relevant information in the target web page, while the third one involves the processing of the found information. Specifically, this last step might not always be present if only the sole knowledge of something was the user's goal, however it will be necessary if the acquired information is to be used for something else (e.g., writing a report, collecting data).

Although contemporary systems offer some support for the information retrieval step, they offer little support for the understanding and processing of information. For example, users often need to scan through whole documents to find relevant information or manually convert the acquired data into usable form.

## 2.1 Challenges in legacy Web search and navigation

A lot of work has been done in various fields concerning the search and navigation on the Web. Most of the focus lies with better acquisition of data from the Web (i.e., crawling and indexing), and with improving the relevance of search results (i.e., via information retrieval methods with better precision and recall or via better query formulation). Most information retrieval work for the Web environment is done by research team affiliated with major search engines such as Google, Yahoo and Bing, while smaller (academic) teams work on personalization, query formulation and related aspects. In this respect, the Semantic Web field is still somewhat immature, although a lot of work has already been done mostly on theoretical foundations, standardization and querying. We analyze legacy Web approaches, because they provide a good foundation for approaches for the Semantic Web and can also be used as a baseline either for enhancement or comparison.



### 2.1.1 Querying and searching

From an end-user perspective, the querying stage is the first and currently perhaps the most crucial step in fulfilling a user's information needs. The available querying means must be expressive enough so that users can specify exactly *what information* is desired while also being *easy to use* and *simple* enough such as not to interfere with the overall user experience.

At present, the most prevalent approach is *keyword-based search*, where the search query is a set of keywords and optional modifiers (e.g., AND, NOT, language and domain restrictions). Many contemporary web search engines (e.g., Google, Bing) support keyword-based querying, while also offering advanced search capabilities that allow users to specify further restrictions. However, studies indicate that advanced search features are complex and impractical to use for most users and usage scenarios (Technical Advisory Service for Images, 2006).

Keyword-based query formulation is difficult for many users as they are unable to figure out good keywords describing their needs. Studies indicate that most search queries are short (up to two-three words) (Jansen, Spink, Bateman, & Saracevic, 1998) and that the use of advanced search forms is deemed impractical even by experienced users (Technical Advisory Service for Images, 2006).

Furthermore, query formulation depends on the application domain for which it is targeted – the average query length was below two keywords for video, two to three keywords for general and audio search, yet above four for image search. Effective query formulation and modification are also necessary, because most users view only the first results page – 50-70% based on the domain (Jansen, Spink, & Pedersen, 2003). As a result, these partial problems must be addressed:

- *Ambiguity of keyword-based queries*, where the meaning of individual keywords is not explicit and the relationships between keywords are not specified. For example, searching for “John Brown” as brown is also a colour and the person John Brown can be mentioned in different ways – J. Brown, John, Mr. Brown, or by his nickname etc.
- *Difficult construction of suitable keyword-based queries*, i.e. the selection of proper keywords that best describe the information need while also preventing irrelevant search results. This requires that users have good knowledge of what keywords are likely to be present in suitable web sites while not being present in other web sites. Thus a generic user must exhibit great knowledge about both what is relevant and what is not relevant.
- *Low expressive power of keyword-based queries*, which cannot be effectively used to describe complex information needs. For example, finding all notebooks with at least 1GB of memory and a 2.0GHz+ dual core CPU priced below \$1500 is nearly impossible via generic search engines.

- *Open-ended search tasks*, where in contrast to typical queries, the user does not know what exactly his information need is or how to specify it beforehand. The exact need only becomes known or specific enough throughout the search session after the possible options are explored.
- *Insufficient support for query modification*, which is necessary if the search results are unsuitable (e.g., bad query, too many irrelevant results, too few results). Only limited assistance for query modification is available, such as the correction of spelling errors. Limited or no practical assistance is available in terms of query suggestions and their possible consequences on search results (the best solutions probably being annotated faceted browsers).

Once a set of suitable search results is returned, users need to select some for further exploration. Presently users must decide based on very limited information such as the title, URL address or a short snippet from the target website. No or limited information is available about the actual content and context of the target web sites, or their type, reliability, and trustworthiness, making the selection process a lengthy “try, fail and try again” exercise.

### **2.1.2 Searching by means of navigation**

Different usage scenarios also result in different user interface needs. For example, typical keyword-based query interfaces might be suitable when searching for specific items but are highly unsuitable if users need to explore the available information space or if they wish to gain an overview of its content, size, scope and structure.

Query-by-example approaches (Geman, 2006) enable users to search by means of navigation, e.g. creating the query by clicking example photos, where the results will be similar photos. More specifically, they allow users to construct search queries via navigation with the immediate evaluation of the query and subsequent optional query modification and/or refinement.

Similarly, *view-based search approaches* provide users with combined search and browsing capabilities, usually via a set of different views of the entire information space and the properties of individual information artefacts. View-based search approaches guide users during query construction by displaying the available options (examples), while the query itself is constructed via navigation instead of writing keywords. View-based search is typically facilitated by faceted browsers (Yee, Swearingen, Li, & Hearst, 2003), which combine the search and querying process with the browsing of search results. Several studies have indicated that combined search and browsing interfaces are required for seamless user experience (Fox & Flanagan, 2003).

### 2.1.3 Navigation and visualization

Ideally, the last step of information retrieval is the navigation in the search results, the browsing of the associated documents and the visualization of the resulting information. Thus, the goal of navigation and browsing is to locate suitable documents with the desired information, while visualization serves as means to improve user orientation and understanding of the respective information.

Search engines typically employ linear navigation in search results with limited or no navigation and orientation support. Individual web sites offer improved navigation support via hierarchical navigation schemes and the following of best practice design guidelines for web sites, which improve user orientation and experience but fail to provide enough support for first time visitors.

Faceted navigation via faceted browsers is often used in online shops, digital libraries or other advanced information retrieval systems. It integrates search and query construction with navigation in search results and individual documents, while also providing navigation support thus improving user experience (Fox & Flanagan, 2003).

Most approaches utilize textual visualization, which employs links, lists or tables of data described via text. Examples include lists of search results in traditional search engines (Google, Bing); for faceted browsers, textual lists of restrictions in facets and tables of search results with their respective attributes.

The IGroup image search engine (Wang, Jing, He, Du, & Zhang, 2007) presents results in semantic clusters. Other approaches include various graphical and graph visualizations suitable for the presentation of the structure of the information space, for the presentation of search results, or for the presentation of changes and trends. These include CropCircles (Wang & Parsia, 2006), various graph visualizations (Schulz & Schumann, 2006), and trend visualization (Ishikawa & Hasegawa, 2007) respectively.

Despite continuous advances in navigation and visualization approaches, the size and complexity of present day systems often results in the infamous navigation problem ("Lost in hyperspace") where users get lost in the information space due to insufficient navigation aids, which is further shown by the high level of navigation recurrence estimated at about 60-80% (Levene & Wheeldon, 2004). More recent estimates indicate a lower recurrence rate of about 46% due to additional navigation options such as opening a page in a new window/tab, introducing new navigation and history tracking issues. Another reason is the increased number of form submissions, indicating a change from a static Web towards a more interactive one (Weinrich, Obendorf, Herder, & Mayer, 2006).

In practice these issues correspond to the following navigation problems:

- *Insufficient information about individual search results/links*, which results in high navigation recurrence as users follow links without knowing what information "lies ahead" and often return immediately because it was not what they expected it to be. This also results in query modification if browsing search results or in leaving a web site in case of general navigation.

- *Lacking navigation guidance and orientation support*, which would allow users to navigate more effectively by providing cues about interesting and/or relevant information, or about their position with respect to local and global landmarks respectively. In this respect, the support for navigation trails – sequences of links (Levene & Wheeldon, 2004) seems vital, as simple links proved to be insufficient. For example, many web search engines favour homepages, but fail to address the issue of further navigation on the target homepage, which might be facilitated by navigation-aided retrieval (Pandit & Olston, 2007).
- *Explorative tasks*, which instead of narrowly focusing on a single “I want this” type of goal assume free navigation best described by “I want to know what's out there and how it all links together”. Thus, explorative tasks require that users focus on a broader set of goals, gain an overview of the information space and the relations between individual information artefacts.
- *Navigation and browsing history*, which are important for users who need to revisit individual sites or get a better overview either in the current session or over a longer time period. Current browsers employ bookmarks and simple history lists. However, bookmarks grow impractical over time as their number increases and they become outdated or obsolete (Weinrich, Obendorf, Herder, & Mayer, 2006). Furthermore, the history list is used only infrequently (Levene & Wheeldon, 2004), thus being of little practical use. While the advantages of even simple graphical history trees were confirmed by a study (Nadeem & Killam, 2001), which compared global trees and domain trees against history lists, mainstream web browsers still lack proper history support despite continuous work on history tools (Mayer, 2009).

Oftentimes, finding a relevant document is not enough, because many documents tend to be long or hard to understand thus making the discovery of relevant information within document difficult. Thus proper navigation within a document and suitable visualization options are necessary in order for users to understand the document contents quickly.

Furthermore, the overall usability of web-based systems depends on the quality and features of its user interface. Even the best information retrieval system in terms of precision and recall will be useless if its user interface is confusing, hard to use and not feature rich enough to enable users to use the functionality of the query, navigation and visualization engine behind it.

## 2.2 Challenges in Semantic Web search and navigation

The Semantic Web initiative aims to provide better search and browsing capabilities by enabling machine readability of information on the Web taking advantage of ontologies and metadata (Shadbolt, Berners-Lee, & Hall, 2006). Semantic Web approaches allow us

to infer new knowledge automatically based on existing knowledge via reasoning on ontologies, and also improve interoperability due to shared semantics of ontologies.

Most Semantic Web content is part of the Deep Web and stored in publicly accessible semantic repositories (e.g., accessible via SPARQL endpoints), in the form of distributed Linked data or as metadata associated with legacy web documents. Despite continuous progress in semantic search engines such as Sindice.com, the original promise of the Semantic Web still remains unrealized because there are few real-world applications that allow end-users to access, view and process Semantic Web information (Shadbolt, Berners-Lee, & Hall, 2006).

Semantics can be used to improve search in different ways. Instead of simple keywords, semantic search queries employ URIs, which explicitly define concepts (“keywords”) and properties (relations between concepts) thus eliminating the ambiguity of keyword-based search. However, the writing of semantic search queries is difficult as it requires the knowledge of the respective query language and concept or property URIs.

Initial approaches augmented traditional keyword-based searches with semantic metadata harvested from web pages. Some more advanced approaches provide hybrid solutions combining traditional full-text search with semantic search if metadata are available or try to identify entities in keyword queries (Guha, McCool, & Miller, 2003).

Semantic Web approaches must address new issues that arise from the principal differences between Semantic Web and legacy Web content, and the way it must be accessed (Ding, Pan, Finin, Joshi, Peng, & Kolari, 2005). These include the use of Semantic Web repositories and query languages to store and access information, distributed Linked data that are spread across multiple locations, and the metadata-like non-visual nature of Semantic Web content.

These vast differences between the Web and the Semantic Web in terms of information granularity, visualization and navigation possibilities require us to define a new perspective on navigation and exploration of the Semantic Web.

From a technical standpoint, the Web is a network of documents interlinked via hyperlinks. It can thus be represented as a *directed graph* where nodes represent documents (information artefacts) and edges represent hyperlinks. The Semantic Web is a network of resources linked via relations, which can also be represented via a directed graph. Thus, web navigation defined as “*the activity of following links and browsing web pages*” (Levene & Wheeldon, 2004) corresponds to the process of moving via edges from one node to another.

Typical web navigation involves the presentation of a single graph node (web page) at a time. However, in the Semantic Web, the presentation of multiple resources at once is more practical due to the different granularity of information and the availability of both data and metadata as opposed to the Web. For example, a job offer page on the Web contains all data about a specific job offer, while in the Semantic Web, the job offer would be represented as several related instances, e.g. one for the job offer, one for the employer, one for each requirement, and one for contact information.

Consequently, in Semantic Web navigation we move or modify a *window*, which defines the presented resources and effectively corresponds to a web page in legacy web navigation. In the trivial case this can be reduced to moving the centre of the window, between graph nodes via edges. In the job offer example, the window would be centred on the job offer instance and also contain the other directly associated instances corresponding to a detail page in traditional search engines (Figure 2). Exploring the properties of e.g., the employer instance, would centre the window on the employer.

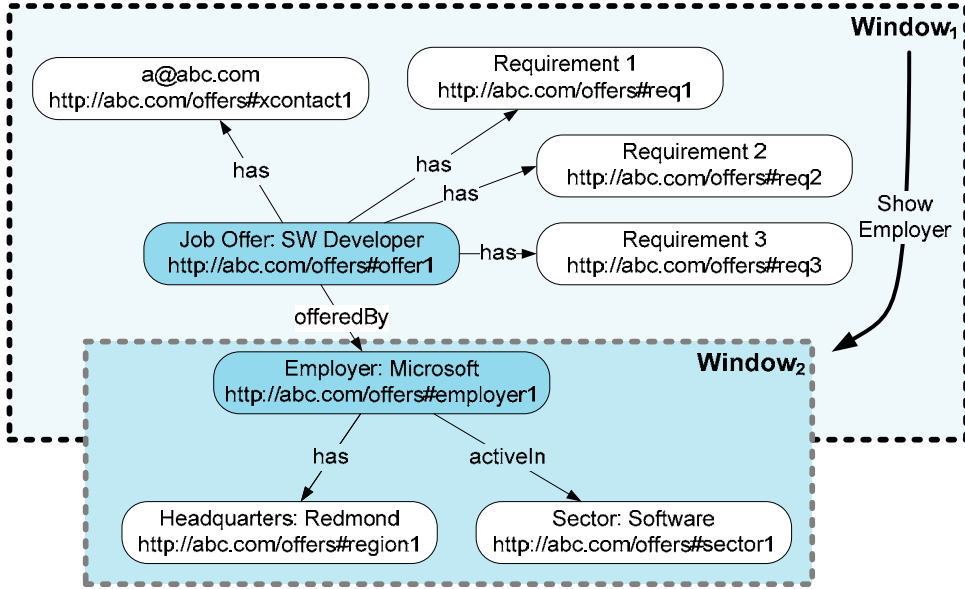


Figure 2. Window movement in the Semantic Web, window centres shown in blue.

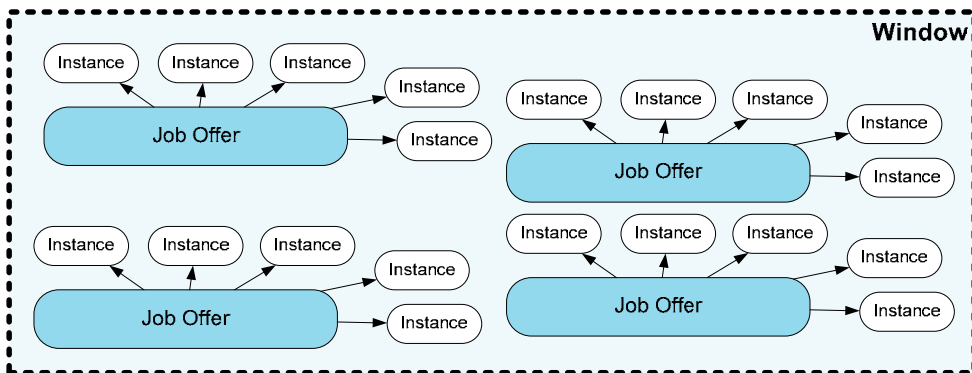


Figure 3. Presentation window without a specific centre.

If however we consider a set of job offers presented simultaneously (e.g., search results), there is no clear node, which might be the window's centre (Figure 3).

Furthermore, the Semantic Web effectively contains both data (e.g., job offer instances, employers and requirements), metadata (e.g., the class *JobOffer*, *Employer* and *Requirement*), and a set of inference rules that can be used to reason on the available information and infer new information. Thus, relations between resources need not be explicitly asserted, but can be inferred based on available metadata and rules enabling additional navigation options compared to traditional web navigation.

Hence, we define Semantic Web navigation as the movement and modification of presentation windows containing resource visualizations, based on the following of embedded links corresponding to relations between resources.

As a result, all the aforementioned issues are even more pronounced in the Semantic Web environment, where the main challenges for its exploration are:

- *Querying* – semantic queries resemble relational database queries rather than typical keyword queries used in web search engines making them impractical even for experienced users. Manual construction of semantic queries (e.g., in SPARQL) is a highly complex task, which in addition to query language proficiency requires prior knowledge about domain concepts in the respective information space (e.g., URIs of classes or properties).
- *Visualization* – Semantic Web contains raw information without any associated presentation templates thus offering no default way to render it in human readable form making end-user grade visualization difficult. Furthermore, resources can be associated with legacy web content (web pages, images, videos, etc.) or have many attributes and relations to other resources causing information overload.
- *Exploration* – the Semantic Web is essentially a graph of resources and their attributes and relations, and also associated legacy web documents (e.g., web pages or multimedia). Exploratory search principles (Marchionini, 2006) stress open ended tasks, learning and understanding of information in context, not just finding a specific resource e.g., with a traditional search engine. Here orientation support, multiple navigation and/or visualization options and the ability to move towards a goal from different directions are needed to provide satisfactory user experience.





## 3 State of the Art in Exploratory Search

---

To facilitate our goal of providing and improving end-user grade exploration of the Semantic Web, we need to address querying, visualization and exploration of Semantic Web resources via a combination of approaches from different fields. Thus our work has a strong multidisciplinary background ranging from information retrieval to adaptive web-based systems (The Adaptive Web: Methods and Strategies of Web Personalization, 2007) and human-computer interaction with focus on faceted browsers and facet generation (Sacco & Tzitzikas, 2009), exploratory search (Marchionini, 2006), information visualization, the Semantic Web (Shadbolt, Berners-Lee, & Hall, 2006) and the Social Web (Staab, et al., 2005).

We focus on exploratory search whose primary aim is to offer support for open-ended exploratory tasks such as learning, investigation, knowledge acquisition, comparison, evaluation as opposed to traditional lookup tasks (Marchionini, 2006). Although exploratory search approaches aim to support all steps of the search process as defined in chapter 2, there is no common or prescribed means of doing so.

We specifically explore the faceted exploration paradigm, which has already been shown to be very promising and generally accepted amongst end-users (Kules, Capra, Banta, & Sierra, 2009). We also take advantage of personalization and look at querying support techniques offered by query expansion, disambiguation and recommendation approaches in order to provide navigation and orientation support. In this respect we also take advantage of history tracking and visualization approaches to support the revisiting of resources (Mayer, 2009), and domain specific and graph-based visualization of information (Schulz & Schumann, 2006).

### 3.1 Faceted browsing

The faceted navigation model is based on the faceted classification scheme of an information space. Originating in library sciences, “*a faceted classification differs from a traditional one in that it does not assign fixed slots to subjects in sequence, but uses clearly defined, mutually exclusive, and collectively exhaustive aspects, properties, or characteristics of a class or specific subject. Such aspects, properties, or characteristics are called facets of a class or subject, a term introduced into classification theory and given this new meaning by the Indian librarian and classificationist S.R. Ranganathan and first used in his Colon Classification in the early 1930s.*” (Wynar & Taylor, 1992, p. 320). For online information retrieval and navigation however, the library definition of faceted classification can be somewhat relaxed as e.g., the exhaustiveness is not strictly necessary.

A detailed (theoretical) introduction into dynamic taxonomies and faceted search is given in (Sacco & Tzitzikas, 2009), thus we present only the basic principles necessary to

understand our work and elaborate on the extensions we have made over the baseline approaches also with respect to the Semantic Web.

Faceted browsers are a view-based search approach that takes advantage of faceted navigation, which is often used in practical applications including online shops or information retrieval systems built around databases, e.g. for product or job search. As opposed to hierarchical navigation, faceted navigation is almost exclusively used for dynamic systems, which generate all views at runtime, due to the exponential number of possible facet and restriction combinations<sup>6</sup>.

Faceted browsers allow users to easily select the desired information by accessing one or more facets available in the used faceted classification and selecting one or more restrictions in those facets. Users visually create faceted queries by navigating and selecting *metadata* (i.e., facets and restrictions respectively), thus specifying the *data* (i.e., results) that should be retrieved. This effectively translates into multidimensional hierarchical navigation in metadata describing a particular information domain or, in graph terms, simultaneous navigation in multiple tree hierarchies (i.e. a forest) as individual facets are often hierarchically organized. The combined navigation state from all facets then defines the global navigation state (i.e., the faceted query) and the presentation window, which shows the search results.

Figure 4 outlines a typical browsing session in a faceted browser, which corresponds to the steps performed during information search:

- 1) *Query* – users typically select facets and restrictions as long as they match their perceived (and known) information needs.
- 2) *Selection* – once the set of available options is exhausted or the users cannot think of any more criteria, they examine the search results and select prospective results for further navigation.
- 3) *Navigation* – detailed information about “good” results can be retrieved and a navigation session via their properties or associated resources can be initiated (e.g., showing associated resources or comparing similar ones).
- 4) *Query modification* – users can relax the query by removing restrictions and repeating the process from step 1.

Advantages of faceted navigation include its flexibility and expressivity – users can navigate the information space in many different ways and combine elements from various facets to specify their information need. This corresponds to the true strength of *faceted navigation* which lies in the fact that it *corresponds to view-based search* and natively provides users with integrated search and navigation capabilities thus alleviating several disadvantages of traditional search approaches (e.g., difficult query construction, unsuitability for open-ended tasks).

---

<sup>6</sup> Each facet consists of a (hierarchical) set of its values – restrictions. For example, Seattle and Washington are both restrictions in a facet describing location with Seattle being a child restriction of Washington as it is located in the state of Washington.

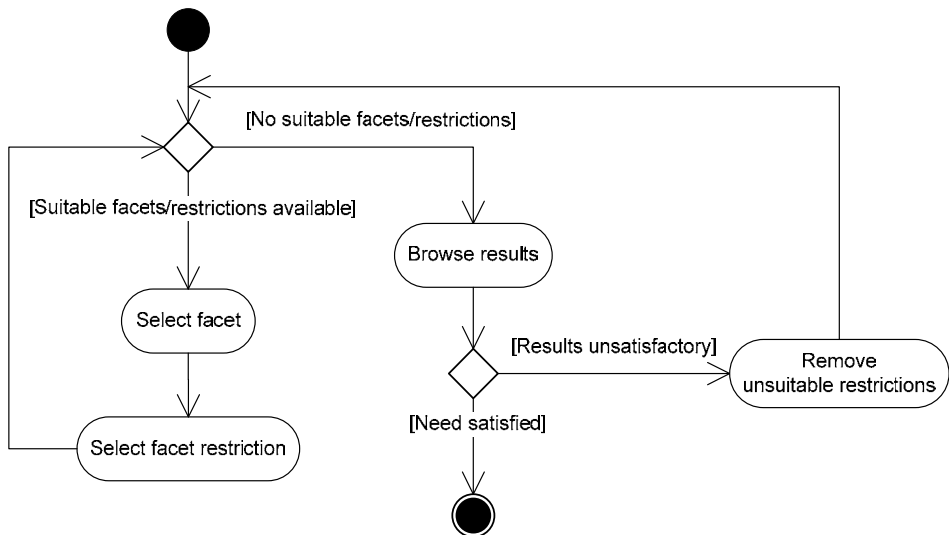


Figure 4. The navigation process in a simple faceted browser from an end-user perspective.

Disadvantages of faceted navigation originate mostly from properties of faceted classifications, which do not provide quick access to popular topics and at first might be difficult to understand due to their scope. Furthermore, a faceted browser interface is somewhat more complex which might result in cognitive overload if too much information is available.

Faceted browsers also typically require the existence of a (manually) predefined faceted classification scheme that is used to construct the faceted browser interface. This can usually be easily facilitated in closed information spaces, where the structure is known and typically does not change. In the wild Web environment, the existence of a static faceted classification cannot be guaranteed which historically resulted in the lack of faceted solutions in generic web search engines. The situation started to change recently with the addition of domain independent facets to Google (e.g., for result freshness) and integration with domain specific search in Bing (e.g., hotel search).

Many contemporary faceted browsers also provide additional usability features in addition to faceted navigation such as:

- Simple sorting of instances based on one given attribute (e.g., name, price or weight, screen size, popularity).
- The comparison of several selected instances and their attributes in a table.
- Different views which are either more or less detailed, with or without images and with a selectable number of simultaneously display results.
- Different actions with search results, such as bookmarking, adding to the shopping cart or rating.

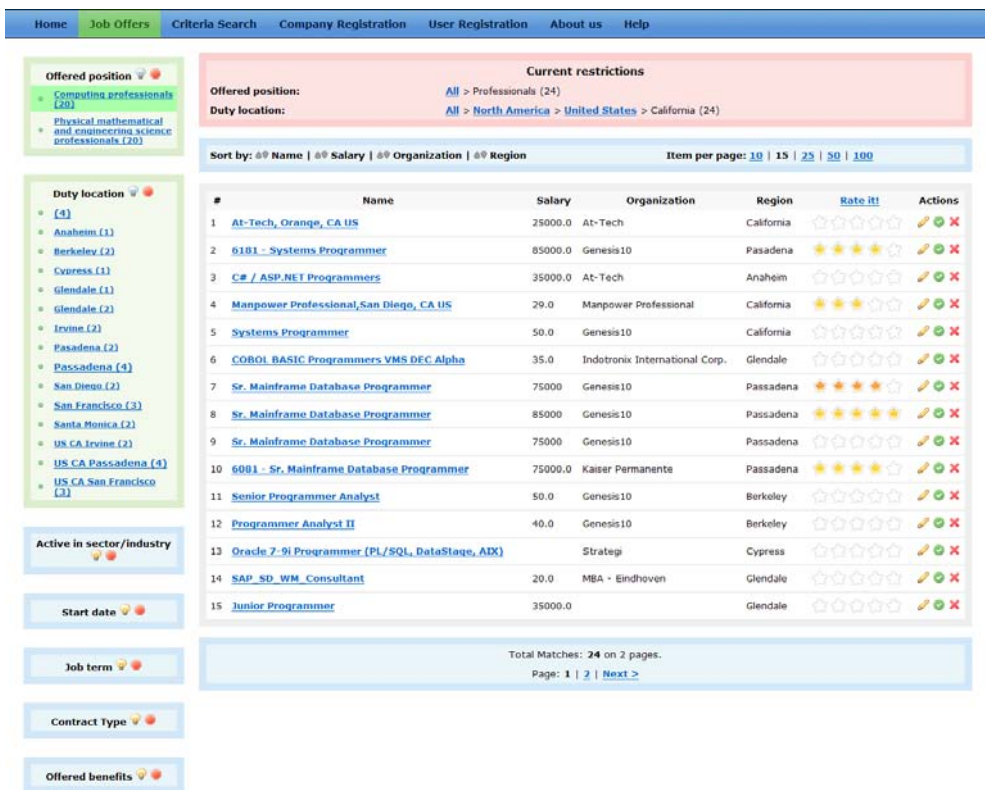


Figure 5. Sample GUI of the faceted browser *Facetic*. Facets are shown on the left (green and blue backgrounds), results in the centre (white background), current restrictions and available actions at the top (pink and blue background).

Figure 5 shows an example graphical user interface of our faceted browser *Facetic* in the domain of job offers (Tvarožek & Bieliková, Personalized Faceted Navigation in the Semantic Web, 2007). It is a faceted browser employing primary and secondary facets with multilevel content. It supports both nominal and ordinal facet values – enumerations and intervals respectively. Individual facets for the type of the offered position, location, industry sector, start date, job term and contract type are shown on the left. The current query is shown at the top, while the results of its evaluation are displayed in the centre. For each search result, the title of the job offer and its main attributes are shown. Additional operations with results include their sorting, rating and optionally editing.

### 3.2 Query construction and search

Search and thus query construction constitute the first part of most user sessions as described in chapter 2. Approaches which can be used for exploration include:

- *Keyword-based search*, where queries constitute a set of keywords and optional modifiers (e.g., NOT). Search results usually correspond to an ordered set of web documents, which contain the specified keywords, ranked via different metrics, e.g. PageRank or HITS.
- *View-based search*, which allows users to visually create queries via navigation. The most prominent practical examples of this approach are faceted browsers, which allow users to create queries by selecting the desired properties of the search results in a set of facets usually shown on the left side of a user interface.
- *Content-based search*, which allows users to select one or more positive or negative examples of information artefacts in the information space. Search results correspond to an ordered set of resources, which are similar to the positive examples (and dissimilar to negative examples). Content-based search is often used in the digital image domain due to the lack of other metadata and keywords that could be used to perform search. Instead, similarity based on low-level image characteristics is used (e.g., luminance).

Typical examples of keyword-based search are web search engines (e.g., Google, Bing) or other open-source search frameworks such as Apache Lucene<sup>7</sup>. While these solutions historically offered little to no orientation and navigation support, being fully focused on lookup, and leaving orientation support to the specific web sites they link to, present day web search engines recently started to offer more advanced features for query refinement. For example, Google now supports query suggestions via auto-complete, while also allowing users to refine the query with generic faceted properties of the results such as modification date or page size. In addition, Bing in the US now also provides faceted domain specific search options such as hotel search.

More sophisticated approaches offer query expansion and disambiguation solutions performing click-stream analysis and data mining (Bordogna, Ronchi, & Psaila, 2009), (Braak, Abdullah, & Xu, 2009), while support for result browsing and selection is still much more limited to snippets, simple ratings of results or more scarcely clustering of search results (e.g., Clusty.com).

Although both querying and query expansion and disambiguation are often performed or supported by a web search engine and thus their support is fairly widespread, typically these features are rather simplistic with their advanced versions remaining only as laboratory research prototypes.

View-based search and content-based search are similar in that they present users with views or examples of what is present in the information space and allow them to select subsets of information based on these examples. In the case of faceted browsers such as mSpace (schraefel, Smith, Owens, Russell, Harris, & Wilson, 2005) users search by specifying properties of the desired results (e.g., a classical musical piece, composer J.S. Bach), which correspond to metadata describing the results and need not necessarily be in

---

<sup>7</sup> Apache Lucene, <http://lucene.apache.org/>

the target resource, which may be an audio recording. Similarly, in query-by-example, users select good examples of results and expect to get similar documents effectively performing a similarity search.

The presently limited use of Semantic Web technologies in most existing systems is caused in part due to the unavailability of metadata especially in open information spaces, which is perhaps also a reason for the limited deployment of faceted browsers in generic applications. While metadata for some applications might be readily available (e.g., digital libraries, online shops), others lack sufficient metadata. Thus automated means of metadata generation from text, tags or images were proposed (Dakka, Ipeirotis, & Wood, 2005), (Diederich & Balke, 2007). Furthermore, the creation of ontologies based on communities and social networks was outlined in (Staab, et al., 2005), which also covered many other aspects common to both the Semantic Web and Social Web.

More recent approaches focus strongly on the automated creation of faceted interfaces for existing data sources such as Wikipedia (Li, Yan, Roy, Lisham, & Das, 2010) or for a search results returned by a web search engine (Zwol & Sigurbjornsson, 2010). Facetedpedia takes advantage of Wikipedia's collaborative vocabulary and user created hyperlinks to construct a hierarchical faceted categorization of the Wikipedia articles. In order to construct a usable interface (i.e., not overcrowded with too many facets), the authors propose individual and aggregate metrics for facet goodness based on pair-wise similarity and navigational cost (Li, Yan, Roy, Lisham, & Das, 2010).

Yahoo! Research recently demonstrated their MediaFaces system that enables faceted exploration of media collections. The guiding principle is the automatic construction of a faceted interface over a media (e.g., photo) collection taking advantage of image search query logs and user tags in Flickr, which alleviates the typical lack of metadata for media searches. Due to the overall size of the collection, all the facets are effectively dynamic, generated at runtime given the initial search query (i.e., query dependent) (Zwol & Sigurbjornsson, 2010).

Neither of the aforementioned approaches address Semantic Web querying and exploration. Consequently with respect to the Semantic Web, current approaches focus on several primary issues:

- Definition, extension and realization of semantic query languages (e.g., SPARQL and its extensions such as SPARUL) and the corresponding ontological repositories and query engines (e.g., Jena<sup>8</sup>, Sesame<sup>9</sup>), including the development of the underlying reasoning engines (e.g., Pellet, Owlrim).
- Semantic Web search engines (e.g., Sindice<sup>10</sup>), with focus on the acquisition, collection, organization and effective querying of various semantic web data, including local repositories, harvested web resources and Linked data.

---

<sup>8</sup> Jena Semantic Web Framework, <http://jena.sourceforge.net/>

<sup>9</sup> Sesame: RDF Schema Querying and Storage, <http://www.openrdf.org/>

<sup>10</sup> Sindice - The Semantic Web index, <http://sindice.com/>

- Practical realization of federated querying over the entire distributed Linked data cloud, which includes the dereferencing and querying of Semantic Web content.

The Swoogle semantic search engine (Ding, Pan, Finin, Joshi, Peng, & Kolari, 2005) addresses the issues of sparse references and distributed files by providing some of the infrastructure required to find, parse, index and query the Semantic Web materialized on the Web in the form of distributed Semantic Web documents (SWD). Swoogle also provides several rankings/metrics for SWDs and detects navigational paths between them.

Perhaps a more simple approach is to use Semantic Web repositories as centralized repositories of Semantic Web data, which can be accessed via standard database like query interfaces and specialized semantic query languages. Presently, many repositories can be considered standalone, meaning they contain data and metadata about all resources they reference.

This alleviates many of the issues associated with finding and matching SWDs on the Web automatically, yet it also reduces the amount of available information. Already today, work is steadily progressing on interlinking repositories with each other so that queries are distributed, thus querying a greater subset of the Semantic Web, while hiding the complexity of the underlying infrastructure.

Similarly, Sindice is another Semantic Web search engine, which indexes RDFS and other semantic (meta)data on the Web and allows users to submit either keyword-based or semantic queries. Still, the results for both Swoogle and Sindice are a list of results, which often correspond to RDF/OWL files, microformats or other embedded data in web pages, which unfortunately are not readable and thus not useful to end-users. To illustrate the issue, a simple search for ‘tim berners lee’ results in many dead links on the first page of the search results, the remaining results being either empty RSS feeds or OWL files with no practical means of visualization.

One practical existing search application of Semantic Web technologies is the GoPubMed<sup>11</sup> search engine for research articles in the medical domain, which employs several integrated medical taxonomies and ontology reasoning to augment search results.

### 3.3 Navigation and visualization

The next step after executing a query is the navigation in the search results or the exploration of the individual search result(s) details. While current search engines have limited support for the navigation in search results (e.g., snippets), advanced result exploration support is virtually non-existent as it would require individual web sites to have been designed and developed to provide user support.

Although some online shops (e.g., Amazon) augment navigation by showing related products via collaborative recommendation (“users who bought this also bought”) or

---

<sup>11</sup> Transinsight GmbH – GoPubMed: <http://www.gopubmed.org/>

augmenting revisitation and orientation by showing recently visited products, this support is not present in the vast majority of (corporate) web sites.

Faceted browsers generally provide strong orientation support for the query construction and result navigation steps as they provide users with additional information which enables them to make informed decisions. They also partially support the successive result exploration step if coupled with a decent content browser.

Wilson and schraefel performed a study comparing three prominent exploratory browsers – Flamenco, mSpace and RelationBrowser++ (Wilson, schraefel, & White, Evaluating Advanced Search Interfaces using Established Information-Seeking Models, 2009). While Flamenco and RelationBrowser++ are more traditional faceted browsers, mSpace takes advantage of RDF data (native to Semantic Web) to provide users with a set of customizable filters that can be used to visualize a subspace of a high dimensional information space. The RelationBrowser++ is tailored to exploration of large statistical data and persistently displays all facets at the top unlike Flamenco, which hides exhausted facets (Zhang & Marchionini, 2005).

In order to better understand user behaviour in faceted browsers, Kules et al. performed a user study examining how searchers interact with individual parts of a faceted browser. The study discovered that users primarily explore the results list and the facets, while mostly ignoring the current query. In fact, the study has shown that facets were an integral part of the exploration experience accounting for about one half of the time spent on actual search results. Kules also argued that the design of exploratory search tasks as well as methodologies for evaluation of exploratory browsers was still in an early stage of development making thorough evaluation difficult (Kules, Capra, Banta, & Sierra, 2009).

VisGets is an advanced visualization and querying solution for legacy web data (Dörk, Carpendale, Collins, & Williamson, 2008). It crawls the Web and gathers news articles, and in turn enables users to interactively explore the data based on three dimensions – time, location and topic. It does not however provide any kind of social recommendation support nor supports navigation or orientation after selecting a search result (i.e., once the user leaves the original search engine). Moreover as VisGets uses its own crawling and indexing engine it cannot be effectively used for general web search or Semantic Web exploration.

Stewart et al. presented an alternative approach called Idea navigation, in which they extract subject-verb-object triples from a predefined news article corpus (Stewart, Scott, & Zelevinsky, 2008). They build hierarchical faceted categories based on the extracted triples, taking advantage of Wordnet term relations, and allow users to via interactive (faceted) selection of subjects, verbs and objects somewhat resembling a natural language query. This approach is in principle very similar to the RDF triple model, but its advantage lies in its support for unstructured textual information due to the use of their custom parsing and pre-processing engine.

The BrowseRDF faceted browser provides elementary facet generation capability over simple RDF data (Oren, Delbru, & Decker, 2006). BrowseRDF automatically identifies facets in source data based on several statistical measures, but offers only very



limited interaction options and does not consider semantic metadata provided in the more expressive RDFS and OWL formats. Other approaches include automatic multifaceted hierarchy generation from textual collections (Dakka, Ipeirotis, & Wood, 2005), and middleware solutions posing as proxies between a databases and users providing a faceted interface by dynamically suggesting a number of facets using precomputed decision trees (Roy, Wang, Nambiar, Das, & Mohania, 2009).

Similarly in the Semantic Web context, Tabulator enables users to browse Linked Data (Berners-Lee, et al., 2006). While Tabulator enables users to take advantage of different visualizations (e.g., map, calendar), it offers only very limited search support. Other Semantic Web browsers / query builders such as Disco Hyperdata browser or Zitgist Dataviewer offer even less user support and are thus useful only to experts.

The practical visualization of Semantic Web resources has so far been very problematic. Probably the best solutions so far were wiki-like applications. DBpedia, the semantic version of Wikipedia visualizes semantic data in a huge table of triples, which is still far from being practical for end-users. Another examples of existing applications also in the Web 2.0 context are the OntoWiki (Auer, Dietzold, & Riechert, 2006), which provides inline RDF authoring support and semantically enriched full-text search, and the Semantic Wikipedia (Völkel, Krötzsch, Vrandečić, Haller, & Studer, 2006), which extends the existing Wikipedia system with lightweight semantic annotations.

Neither of these approaches can be effectively used for complex interactive exploration of Semantic Web content, which in addition to advanced (faceted) querying needs to support interactive information visualization and exploration of graphs (Semantic Web being a graph). Here, also graph visualization and interaction approaches must be considered as described in (Schulz & Schumann, 2006). Other visualization approaches such as CropCircles (Wang & Parsia, 2006) and TagSphere (Aurnhammer, Hanappe, & Steels, 2006), which focus on the presentation of metadata can also be used to visualize link structures and thus support navigation in exploratory search interfaces.

CropCircles is a topology sensitive approach to visualization of OWL class hierarchies inspired by treemaps (Wang & Parsia, 2006). Since it visualizes (class) hierarchies, it might be ideally suited for the visualization of facets, which contain restriction hierarchies (Figure 6). CropCircles support orientation by providing quick overview of the topology (i.e. the size, depth and complexity of a hierarchy), while also providing a visually pleasing nested presentation of nodes.

Circles represent nodes, their size corresponds to the size of the respective subtree rooted at a particular node. Child nodes are sorted in descending order based on their size. Different layout strategies are employed based on the size distribution of child nodes (e.g., dominant child node, equal sized children).

While typical orientation support approaches focus on support during a user session, from an exploratory perspective, providing user support between multiple sessions is just as crucial. Revisitation support typically includes various browser tools such as bookmarks and a history list, which have been shown to be of little practical use to end-users due to the associated overhead and the limited capabilities of searching within history.

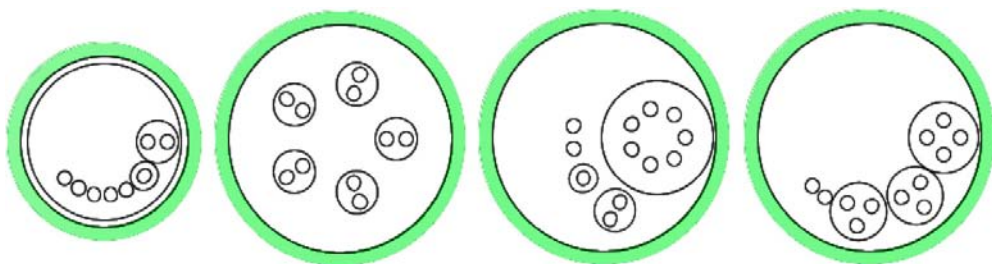


Figure 6. Topology sensitive visualization via CropCircles, taken from (Wang & Parsia, 2006). Layout strategies from left to right – single child, equally sized children, dominant child, no dominant child.

Mayer provides a broad survey of existing history and revisitation approaches, along with open problems including acquisition, search and visualization of history entries and metadata (Mayer, 2009). While current browser and search engine extensions support features such as full-text search in history (e.g., the Firefox plug-in WebMynd) or tree-based history visualization more suited to the recursive nature of web navigation (e.g., the Firefox plug-in HistoryTree, Pad Tree or WebView (Mayer, 2009)), users still encounter issues with keyword guessing, disorientation and dead links.

### 3.4 Review of selected existing approaches

It has been shown that faceted browsing approaches are highly suitable for exploration of various information collections including both structured semi-structured and unstructured information. Their main advantages are intuitive user friendly exploration interfaces, high expressivity via faceted classification and overall suitability for large data collections. However, few of these approaches have so far been used for Semantic Web exploration, and even fewer included advanced support features such as personalization, orientation and revisitation support.

In this section, we present a more in-depth survey of existing exploration approaches from which we drew inspiration for our Semantic Web exploration approach:

- *Flamenco*: FLexible information Access using MEtadata in Novel COmbinations (Yee, Swearingen, Li, & Hearst, 2003)
- *Ontoviews*: A tool for creating Semantic Web portals (Mäkelä, Hyvönen, Saarela, & Viljanen, 2004)
- *Relation Browser++* (Zhang & Marchionini, 2005)
- *mSpace* (schraefel, Smith, Owens, Russell, Harris, & Wilson, 2005)
- *BrowseRDF*: Faceted RDF browser (Oren, Delbru, & Decker, 2006)
- */facet*: Browser for heterogeneous semantic repositories (Hildebrand, van Ossenbruggen, & Hardman, 2006)

- *Tabulator*: Generic data browser (Berners-Lee, et al., 2006)
- *TagSphere* (Aurnhammer, Hanappe, & Steels, 2006)
- *IGroup*: Image search engine (Wang, Jing, He, Du, & Zhang, 2007)
- *VisGets* (Dörk, Carpendale, Collins, & Williamson, 2008)
- *Microsoft Pivot* presented by Microsoft LiveLabs in 2010<sup>12</sup>

## Flamenco

Flamenco was the pioneering faceted browser originally devised for exploration of image collections in digital libraries (Yee, Swearingen, Li, & Hearst, 2003). Due to its pioneering role, Flamenco stressed interface design and the HCI aspects of faceted browsing over traditional information retrieval systems. Furthermore, Flamenco divided the exploration process into three steps:

- The *opening*, where users are presented with a broad overview of the entire information content (Figure 7). This shows all the facets, i.e. the structure of the information space thus giving users a good understanding of what information can be found and explored.
- The *midgame*, where users can refine their search via additional facets while simultaneously exploring the results of their search (Figure 8). While individual results are not specifically ranked (faceted results have no default ordering), they are grouped based on their attributes. Empty or exhausted facets, which would lead to no results, are hidden from the user interface.
- The *endgame*, where users explore the properties of an individual search result with options for query refinements to find similar items (Figure 9).

At its time, Flamenco worked likely with relational data corresponding to static manually predefined facets. Similarly, the linking to related objects in the detailed view was limited to asserted relations based on similar attributes, i.e. searching for other items that had the same attribute values. The browser also had little support for user customization or personalization. Nevertheless, it proved its point that faceted browsing was suitable for exploratory tasks in structured information spaces.

## Ontoviews

OntoViews (Mäkelä, Hyvönen, Saarela, & Viljanen, 2004) is a comprehensive tool for the creation of Semantic Web portals based on the Apache Cocoon framework<sup>13</sup> and a service oriented architecture using Ontogator as a view-based search service provider (Mäkelä, Hyvönen, & Saarela, 2006). Ontoviews supports faceted navigation over RDFS ontologies and link recommendation services via Ontodella.

---

<sup>12</sup> PivotViewer for Silverlight, <http://www.microsoft.com/silverlight/pivotviewer/>

<sup>13</sup> Apache Cocoon: <http://cocoon.apache.org/>

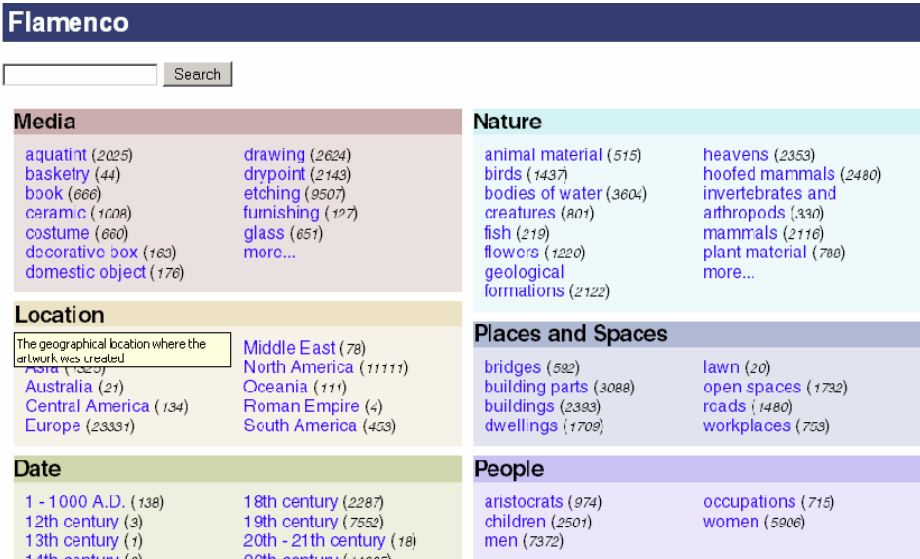


Figure 7. The opening showing all the available facets in the information space thus giving users a broad global understanding of the information content and its structure, taken from (Yee, Swearingen, Li, & Hearst, 2003).

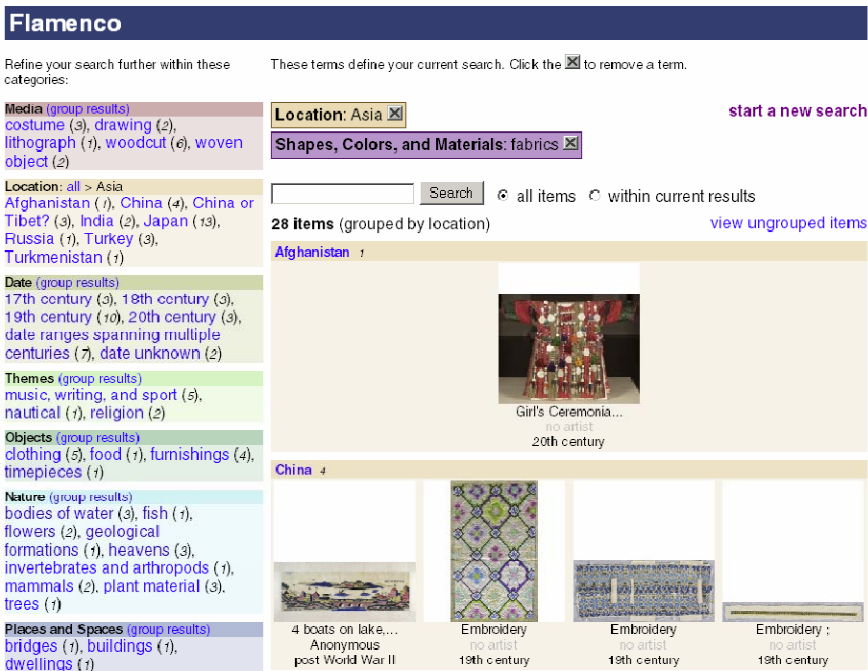



Figure 8. The midgame showing the available facets on the left, current query at the top and the grouped search results in the centre, taken from (Yee, Swearingen, Li, & Hearst, 2003).

**Flamenco**

item 1 of 22 ([back to results](#))

**Bag (ch'uspa)**



[next](#) ▶

**Current search ([start a new search](#)):**  
**Shapes, Colors, and Materials:** [fabrics](#) > [wool](#)

**Select any link to see items in a related category.**

more general categories	information about this item
<b>MEDIA</b> <hr/>	<b>MEDIA</b> <a href="#">costume</a> (60)
<b>LOCATION</b> <hr/> <ul style="list-style-type: none"> <li><a href="#">South America</a> (453)</li> <li><a href="#">Bolivia</a> (153)</li> <li><a href="#">Cochabamba</a> (25)</li> <li><a href="#">Tapacari</a> (19)</li> </ul>	<b>LOCATION</b> <hr/> <a href="#">Challa</a> (10)
<b>DATE</b> <hr/> <ul style="list-style-type: none"> <li><a href="#">20th century</a> (74205)</li> </ul>	<b>DATE</b> <hr/> <a href="#">1950 - 1959</a> (1027)
<b>SHAPES, COLORS, AND MATERIALS</b> <hr/> <ul style="list-style-type: none"> <li><a href="#">fabrics</a> (245)</li> </ul>	<b>SHAPES, COLORS, AND MATERIALS</b> <hr/> <a href="#">wool</a> (22)
<b>ARTISTS</b> <hr/>	<b>ARTISTS</b> <hr/> <a href="#">unknown</a> (2745)
<b>RECORD NUMBER:</b> <b>OBJECT TITLE:</b> <b>DESCRIPTION:</b>	205128 Bag (ch'uspa) Small bag, two pieces, patterned double cloth, multicolored, strap, tubular edge binding on sides and top, multicolored wool trim and tassels on bottom.
<b>CULTURE OR PEOPLE:</b>	Quecha/Aymara

Figure 9. The endgame showing detailed information about a search result with options for query refinement to similar items (right), taken from (Yee, Swearingen, Li, & Hearst, 2003).

A demonstration application of OntoViews is publicly available in the domain of digital libraries (museums) as MuseoSuomi<sup>14</sup>. Figure 10 shows the user interface of OntoViews, which copies the typical faceted browser layout with facets on the left and content on the right. Search results are presented in groups corresponding to the last used facet. The detailed instance view (Figure 11) shows instance attributes at the top, followed by a list of faceted categories to which the instance belongs. Recommended links to related instances are shown on the right.

Furthermore, OntoViews has a mobile user interface, which retains the functionality of the original desktop interface albeit with minimal screen size.

Link generation is based on predicates in the form  $p(\text{subjectURI}, \text{targetURI}, \text{explanation})$ , which succeed when two resources ( $\text{subjectURI}$ ,  $\text{targetURI}$ ) should be linked together with label *explanation*. Individual rules/predicates are processed by the Ontodella service for link generation.

The use of XSLT in the user interface and query transformations provided high interface flexibility, yet resulted in complicated templates that are tied to a specific RDF/XML representation.

<sup>14</sup> MuseoSuomi: <http://www.museosuomi.fi/>

**MuseoSuomi**  
- Suomen museot semanttisessa webissä -

University of Helsinki

Uusi haku | Ohjeet | Näytä kaikki kategoriat | Tietoa ohjelmasta | MuseoSuomi-palautte | English Tutorial | About MuseumFinland

**Käsitteet:**   ☐ tarkenna

**Hakuehdot**

**Kategoria:** Esinetyyppi > työvälineet (ryhmittele kohteet) (poista)

**Kohteet ryhmiteltyinä kategorian työvälineet mukaisesti**  
(näytä ilman ryhmittelyä)

**tekstiilikäsityövälineet**, kohteet 1-4/219 (ryhmittele kohteet)

**Esinetyyppi:** [kakki](#) > [työvälineet](#) (koko luokittelu)

[tekstiilikäsityövälineet](#) (219),  
[kansanlaakinnän työvälineet](#) (1),  
[muut työvälineet](#) (36), [maataloustyövälineet](#) (7),  
[metallityövälineet](#) (1),  
[pilkkomis- ja hienontamisvälineet](#) (4),  
[kirjoitusvälineet](#) (9), [metsätyövälineet](#) (4),  
[työkalut](#) (22)

**Materiaali** (koko luokittelu) (ryhmittele kohteet)

[materiaalit](#) (241)

**Valmistaja** (koko luokittelu) (ryhmittele kohteet)

[henkilöt](#) (9), [tuotemerkit](#) (2),  
[viritykset](#) (38)

**Valmistuspaikka** (koko luokittelu) (ryhmittele kohteet)

[Afnika](#) (2), [Etela-Amerikka](#) (1),  
[Eurooppa](#) (84)

**Valmistusaika** (koko luokittelu) (ryhmittele kohteet)

[aikakaudet](#) (90), [vuosisadat](#) (89)

**Käyttäjät** (koko luokittelu) (ryhmittele kohteet)

[henkilöt](#) (54), [laitokset](#) (1),  
[viritykset](#) (3)

**Käyttöpaikka** (koko luokittelu) (ryhmittele kohteet)

[Eurooppa](#) (71)

**Käyttötilanne** (koko luokittelu) (ryhmittele kohteet)

[kansalais-, harrastus- ja vapaa-ajantoiminta](#) (4),  
[kohteelle tehtävät toimenpiteet](#) (17),  
[maatalous ja karjanhoito](#) (2), [ruuan- ja juomanvalmistus](#) (3),  
[elollisten olentojen perustoiminnat](#) (2),  
[elinkieinot](#) (9), [valmistustekniikat](#) (179)

**Kokoelma** (koko luokittelu) (ryhmittele kohteet)

[Espoon kaupunginmuseon kokoelmat](#) (54),  
[Kansallismuseon kokoelmat](#) (193),  
[Lahden kaupunginmuseon kokoelmat](#) (50)

**kehräpuni**, kuosali (NBA SU4527 50)

**kehrunlanta**, kehräpuni, kuezzel, kuosali (NBA SU5069 26)

**rukinlapi** (ECM 100 1)

**sneldde**, väärtinänluppio, väärtinäpyörä (NBA SU2449 7) (edellinen) / (seuraava)

**suonirauta**, suonieniskentärauta (ECM 2711 1) (edellinen) / (seuraava)

**nappikoukku**, nappikoukku (ECM 3594 264)

**kietkamlabdzi**, komssiohanna (NBA SU4922 32)

**palohosat**, palohosat (ECM 614 1)

**hontillasta** (NBA SU4135 166)

Figure 10. Example of the OntoViews GUI, facets shown on the left, search results shown on the right.

Moreover, OntoViews does not take advantage of OWL metadata, it must be manually configured to use facets and link recommendation (e.g., via aforementioned rules), and has no support for personalization based on user preferences.

Overall, although OntoViews maintained the primary functionality, layout and limitations of Flamenco, it also added several novel aspects such as support for Semantic Web data in RDFS form and some support for dynamic link generation between similar/related resources based on knowledge asserted or inferred from the underlying knowledge base.

## RelationBrowser++

Zhang and Marchionini present a slightly different approach to faceted browsing in their RelationBrowser++, which is aimed at large statistical collections of data (Zhang &



Marchionini, 2005). Its main focus is to provide interactive and dynamic exploration of the collection and thus support the user in better understanding its contents. To this end, the interface visualizes the properties of information artefacts in several columns (i.e., facets) while also supporting dynamic previewing of the next query results (Figure 12).

The screenshot displays the MuseoSuomi web application. At the top, there is a header with the Helsinki Institute for Information Technology logo and the title "MuseoSuomi - Suomen museot semanttisessa webissä". Below the header is a navigation bar with links like "Uusi haku", "Takaisin hakusivulle", "Ohjeet", "Tietoa ohjelmasta", "MuseoSuomi-palaute", "English Tutorial", and "About Museum Finland". A search bar contains the text "(<<) tekstiilikäsityövälineet (219) (>>) kansanlääkinnän työvälineet (1)". Below the search bar, there are links for "(kehruslauta, kehräpuu, kuezzel, kuosaiti <) rukinlapa (> sneldde, väärtinänlumpio, väärtinäpyöri)".

The main content area is divided into three columns:

- Left Column (rukinalapa):** Displays a photograph of a wooden artifact, a "kehruslauta" (weaving board).
- Center Column:**
  - Valmistuspaikka:** Suomi
  - Valmistusaika:** 1793
  - Käyttöpaikka:** Suomi, Bembole, Espoo, Suomi, Vanhakartano, Espoo, Suomi
  - Asiasana:** KEHRUU, KORISTEVEISTO, PUUMERKKI, VUOSILUKU
  - Museokokoelma:** Museokokoelma
  - Vastuumuseo:** Espoon kaupunginmuseo
  - Asiasanasto:** Espoon kaupunginmuseon sanasto
  - Esineen numero:** ECM:100.1
  - ID:** 1001
  - Esinetyyppi:**
    - työvälineet (298) > tekstiilikäsityövälineet (219)
    - > kehrum ja langanvalmistuksen työvälineet (63) > kehrusvälineet (59)
    - > kuoritalonpöydät (3) > rukinlavat (1)
  - Valmistuspaikka:**
    - Eurooppa (2541) > Suomi (2239)
  - Valmistusaika:**
    - aikakaudet (3024) > historiallinen aika (3023) > uusi aika (3013)
    - vuosisadat (3012) > 1700-luku (123)
  - Käyttöpaikka:**
    - Eurooppa (2232) > Suomi (2227)
    - Eurooppa (2232) > Suomi (2227) > Etelä-Suomen lääni (1999)
    - Uusimaa-Nyland (670) > Espoo (512)
    - Eurooppa (2232) > Suomi (2227) > Etelä-Suomen lääni (1999)
    - Uusimaa-Nyland (670) > Espoo (512) > Bembole (14)
  - Käyttötilanne:**
    - valmistustekniikat (1587) > tekninen työ (39) > veisto (32)
    - > koristeveisto (8)
    - valmistustekniikat (1587) > tekstiilitö (886) > kuivutö (74) > kehruu (64)
  - Kokoelma:**
    - Espoon kaupunginmuseon kokoelmat (1190) > Museokokoelma (1129)
- Right Column:**
  - Sama käyttöpaikka:**
    - Bembole:
      - lämsivuolin
      - opetusväline-peli
      - opetusväline-peli
      - opetusväline-peli
      - opetusväline-peli
    - Espoo:
      - kuvakirja, kuvakirja, kangasta
      - lennikkilapsen lyhytuhainen lenikki
      - neuletakkinaisen neuletakki
      - hartiavaate-naisen pitsinen hartiavaate
      - puvun vlaosa, jakkunaisen puvun vlaosa
    - Suomi:
      - ruokailinruokailina, damasti
      - kaitalinalakaitalinala, etupistokirjontaa
      - pöytälinapöytälinala, kirjoitu
      - pöytälinaristikokirjontainen pöytälinala
      - kaitalinala batistilina, kirjoitu
  - Esineeseen liittyvään paikkaan liittyviä muinaismuistoja:**
    - Espoo:
      - Rovikkiöt
      - Puolustusvarustukset
      - Rovikkiöt
      - Rovikkiöt
  - Samaan aiheeseen liittyviä esineitä:**
    - ajan käsitteet:
      - hevosloimi
      - arkkivaatearkku
      - takki-vampite

Figure 11. Example of the OntoViews GUI for presentation of instance details with Instance attributes (top), other facet categories (centre) and related instances (right).

While showing several promising features, such as dynamic query previews and visual cues to assess the size of individual facet restrictions, RelationBrowser++ was fairly limited in its interface design user friendliness and the fact that it used predefined static facets. As such it resembled more of a database exploration tool for professional users and large statistical collections than an end-user grade tool for web exploration.

## mSpace

The mSpace<sup>15</sup> browser is based on a set of successive columns, which correspond to dimensions in the information space (schraefel, Smith, Owens, Russell, Harris, & Wilson,

<sup>15</sup> Demo available at: <http://demo.mspace.fm>

2005). Overall, mSpace presents a significant improvement over the previous approaches in terms of user friendliness and customization. The columns can be rearranged by the user to fit their preferences, while their order also affects query evaluation, which is performed from left to right (Figure 13). In addition to the columns, the mSpace interface includes an info view which shows result details and preview cues that show what the results of a given action (selection in column) would be. To further customize user experience, users can save their preferred browser arrangement or add favourites for future reference.

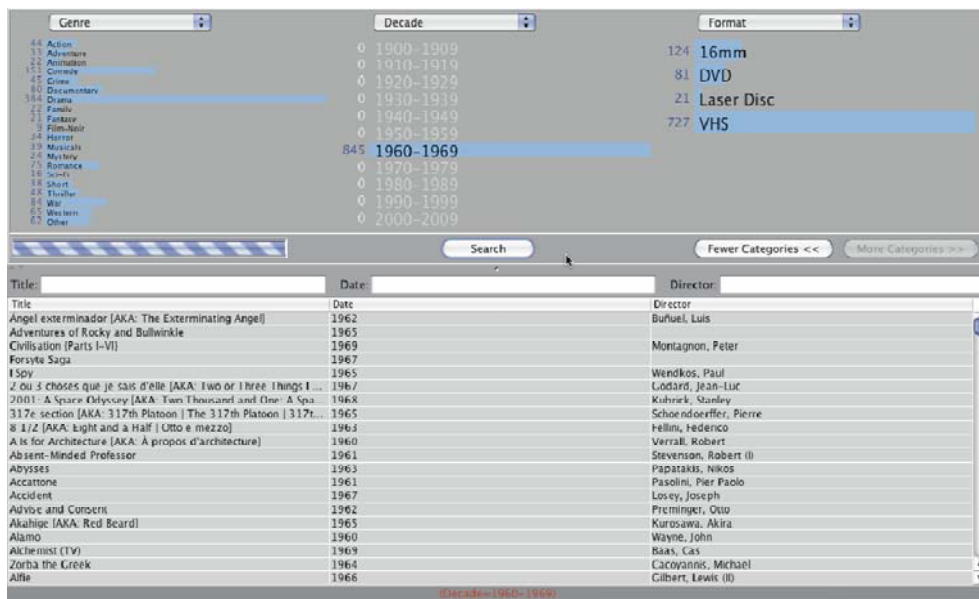


Figure 12. Example of the RelationBrowser++ interface. Hovering over facet restrictions previews the resulting item distributions; taken from (Wilson, schraefel, & White, *Evaluating Advanced Search Interfaces using Established Information-Seeking Models*, 2009).

Since mSpace takes advantage of RDF data representation with an SQL back-end it could likely be used for Semantic Web exploration with some adjustments. While it is unclear whether the columns are manually predefined or automatically generated, the fact that RDF is used as the underlying data representation should make it relatively easy to adjust for new information domains. Thus the major points brought forth by mSpace can be summarized in:

- Intuitive and user friendly interface with support for user customization.
- Multiple, customizable ways to query and explore the data set also using multiple views and navigation cues.
- Use of ontologies and RDF data representation, although not OWL.





Figure 13. Example of mSpace, taken from (Wilson, schraefel, & White, *Evaluating Advanced Search Interfaces using Established Information-Seeking Models*, 2009).

### BrowseRDF: Faceted RDF browser

BrowseRDF (Oren, Delbru, & Decker, 2006) is a faceted browser for Semantic Web data in RDF format. BrowseRDF can automatically generate a faceted interface from arbitrary RDF data with little manual configuration.

BrowseRDF extends typical faceted queries with RDF semantics, e.g. existential selection, inverse selection, non-existential selection and others. Furthermore, it defines statistical metrics from automatic facet ranking and adaptation, such as predicate balance, object cardinality and predicate frequency.

Figure 14 shows the GUI of BrowseRDF in the domain of wanted FBI suspects. Individual facets with new selection types are shown on the left, instance details are shown in the centre.

Similarly to OntoViews, BrowseRDF does not take advantage of OWL data and automatically generates facets for all available RDF predicates, even those with little sense for the end users. Moreover, it only employs statistical metrics computed from the supplied RDF data and thus supports no personalization, nor link recommendation.

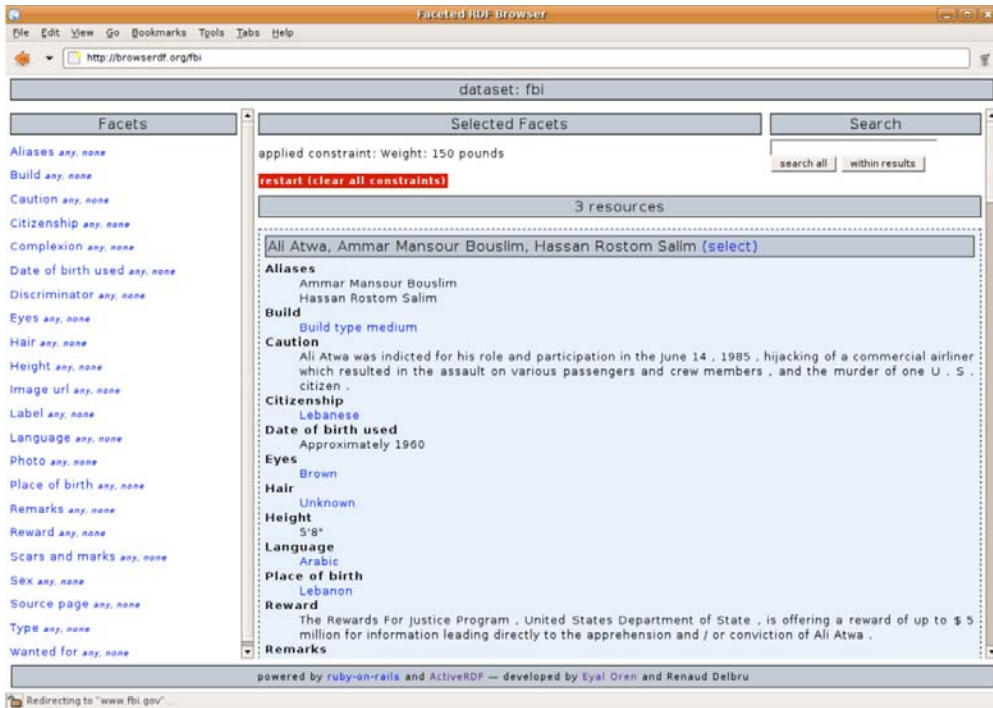


Figure 14. Example of BrowseRDF GUI with existential facet restrictions (left), taken from (Oren, Delbru, & Decker, 2006).

## /facet: Browser for heterogeneous semantic repositories

/facet (Hildebrand, van Ossenbruggen, & Hardman, 2006) is a faceted browser for heterogeneous information spaces consisting of distributed semantic repositories represented in RDFS. It takes advantage of both the *rdfs:subClassOf* property and the *rdfs:subPropertyOf* property in order to process facet restriction hierarchies.

Furthermore, /facet supports multi-type queries and runtime facet specification thus greatly increasing flexibility and support for heterogeneous repositories. The multi-type capability effectively translates into an additional facet, which is used to specify the target data type. Based on the selection in the type facet, other facets are made available.

Figure 15 shows the /facet GUI. The selected type *vra:Work* corresponds to facets *Creator*, *Date* and *Material.Medium*. Moreover, /facet supports semantic keyword search, which allows users to perform keyword-based search on

- all instances (helps find a suitable instance type),
- individual facets (improves movement and restriction selection),
- and across all facets (improves orientation).

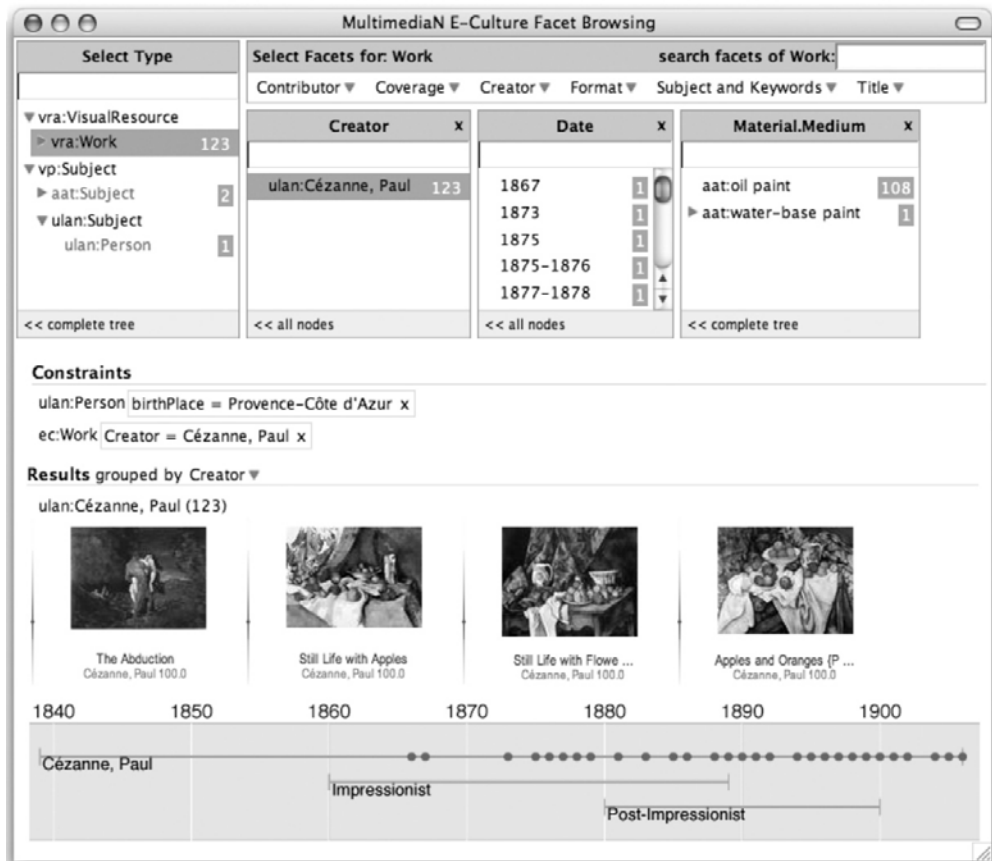


Figure 15. Example of the /facet GUI with multiple facets (top), constrained search results (centre) and timeline display (bottom), taken from (Hildebrand, van Ossenbruggen, & Hardman, 2006).

Lastly, /facet supports the grouping of search results based on individual properties and timeline visualization of dates. However, it does not support personalization nor advance link generation and recommendation techniques.

### Tabulator: Generic data browser

The Tabulator project aimed to create a generic data browser for Linked data, which would be capable to visualize distributed Linked data and allow users to explore individual resources (Berners-Lee, et al., 2006). The prototype itself features several standard views (e.g., table, map, calendar, timeline) which allow users to explore the data collection while also providing basic SPARQL query construction capability.

Figure 16 shows an example of the Tabulator interface showing the description of the Tabulator project itself in a long nested table of textual descriptions and URIs, similarly to the public DBpedia web interface. Users can click and expand the blue/green icons which indicated the availability of further information in the Linked data cloud.

Similarly to DBpedia, the visualization or maybe the information itself is accessible to end users in that far that it is not plain XML, but hardly user friendly enough to be used by casual users, unlike previous browsers such as mSpace.

About	<ul style="list-style-type: none"><li>▶ Thing</li><li>▶ n0</li><li>▶ <a href="http://dig.csail.mit.edu/2007/wiki/tabulator">http://dig.csail.mit.edu/2007/wiki/tabulator</a></li><li>▶ Table Of Contents</li></ul>
Bug database	<ul style="list-style-type: none"><li>▶ <a href="http://dig.csail.mit.edu/2007/wiki/tabulator">http://dig.csail.mit.edu/2007/wiki/tabulator</a></li><li>▶ Tabulator Issue Tracker</li><li>▶ <a href="http://dig.csail.mit.edu/issues/tabulator/">http://dig.csail.mit.edu/issues/tabulator/</a></li></ul>
Created	<p>---</p> <p>2006-01-26</p> <p>2006-01-27</p> <ul style="list-style-type: none"><li>▶ <a href="http://dig.csail.mit.edu/2007/wiki/2006-01-26">http://dig.csail.mit.edu/2007/wiki/2006-01-26</a></li></ul>
Description	<p>The Tabulator is a generic data browser. It provides a way to browse and query RDF data in a variety of formats. Outline, table, map, calendar, and timeline views come standard. Adding new views is a snap. The Tabulator also has features for the power user wanting to export data or edit their queries by hand. The Tabulator is open source and written in Javascript. The source can be easily combined with custom web pages to add data browsing functionality. It currently runs with Firefox, and requires Firefox preferences to be set -- see the tabulator help page. The Tabulator is open source under the W3C software license.</p>
Developer	<p>hjjghghghj</p> <p>Dany thomas</p> <p><a href="http://huemer.lstadler.net/role/uc1/foaf.rdf#black">http://huemer.lstadler.net/role/uc1/foaf.rdf#black</a></p> <ul style="list-style-type: none"><li>▶ n0</li><li>▶ n1</li><li>▶ Ilaria Liccardi</li><li>▶ Kenny Lu</li><li>▶ Melvin Carvalho</li><li>▶ Oshani Seneviratne</li><li>▶ Data on location of libraries mostly in the UK (sparql, slow)</li><li>▶ Michael Hausenblas</li><li>▶ Albert Au Yeung</li></ul>

Figure 16. Example of the Tabulator interface showing the semantic description of the Tabulator project in a long nested table (shortened here).

Although Tabulator technically allows users to build SPARQL queries, its means of doing so are not end-user friendly. Also it has only limited capability to actually perform a search as it works with Linked data and thus has no underlying query engine to rely on. Thus it mostly renders all available textual information in alphabetical order including (inconsistent) metadata, and since it performs no personalization of the content nor exploration experience it easily overwhelms casual users. Still, professional users are likely to take advantage of some of its SPARQL querying and RDF visualization capabilities.

TagSphere

TagSphere is an exploratory approach for augmented content-based image retrieval using collaborative tagging (Aurnhammer, Hanappe, & Steels, 2006). Its main principle lies in navigation and orientation support via tag-based visualization of search results – users can explore an image collection by selecting positive and negative examples (Figure 17).

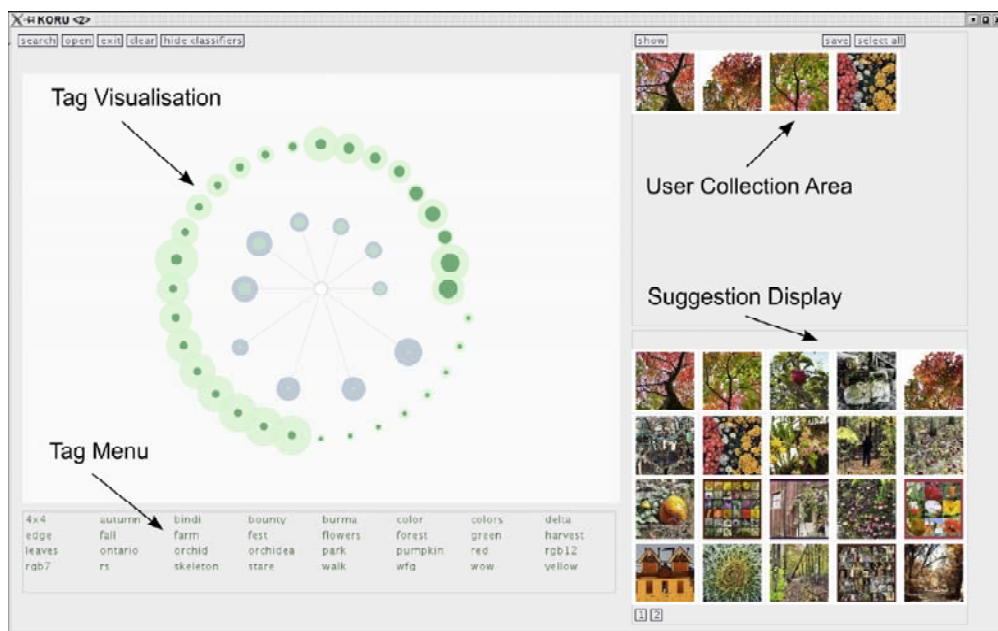


Figure 17. TagSphere interface supports content-based search by selecting images from a set of suggestions (bottom right), which can be access from the search result overview (top left). The current query includes positive and negative examples (top right); taken from (Aurnhammer, Hanappe, & Steels, 2006).

The white circle in the centre denotes the user's image collection corresponding to a query-by-example (Figure 18). The tag sphere represents the search results, i.e. sets of search results corresponding to tags associated with images from the query samples shown in the user collection area. The size of circles in the tag sphere denotes the number of images, distance to the white circle denotes the number of overlapping images and the circles in their respective centres denote the overlap returned by an image classifier, which evaluates low-level image properties against the query. The outer Classifier sphere works the same way, yet describes a different set of results, which are returned by the classifier instead of a tag search. E.g., for sets *leaves* and *park* the tags seem to match the low-level image properties quite well, while having high overlap with the user's collection.

While TagSphere does not work with semantic metadata nor faceted classification, it presents an interesting way to visualize results and support content-based search, which neither of the previous approaches supported. TagSphere shows users many visual examples of possible results and thus gives them a better understanding of the query and contents of the information space. Moreover, TagSphere also gives users a global overview of all search results instead of just listing the first K results as typical search engines (Figure 18).

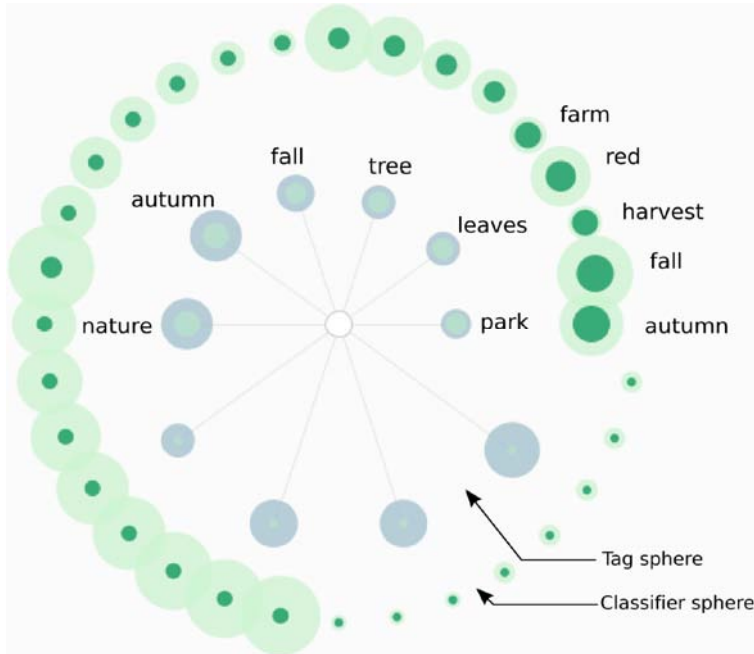


Figure 18. Tag visualization via TagSphere where different circles indicate overlap between the query and search results corresponding to the given tags; taken from (Aurnhammer, Hanappe, & Steels, 2006).

### IGroup: Image search engine

IGroup (Wang, Jing, He, Du, & Zhang, 2007) is a typical keyword-based search engine in the image domain. However, it presents search results in semantic clusters that users can use for search via query-by-example thus effectively expanding their query options.

Figure 19 shows the IGroup interface, with a list of identified clusters on the left. These correspond to different “tigers” identified in the use collection and allow users to select specific subspaces of the information space as in view-based search. Individual search results are presented in a matrix in the centre of the GUI with descriptive information.

The clustering algorithm takes as input the results of a standard keyword-based search and gives a list of annotated clusters as its output. It takes advantage of text, which is available for individual images and selects top-ranked phrases via n-gram analysis (phrase frequency, document frequency, phrase length, etc.).

Advantages include a wider coverage, where some minor, previously hidden, subsets are now visible. Furthermore, individual clusters are annotated thus allowing users to refine the query based on the displayed images instead of writing keywords.

Disadvantages include no support for personalization and link generation and no direct support for Semantic Web data as the source data results from a traditional keyword-based query to some other search engine.



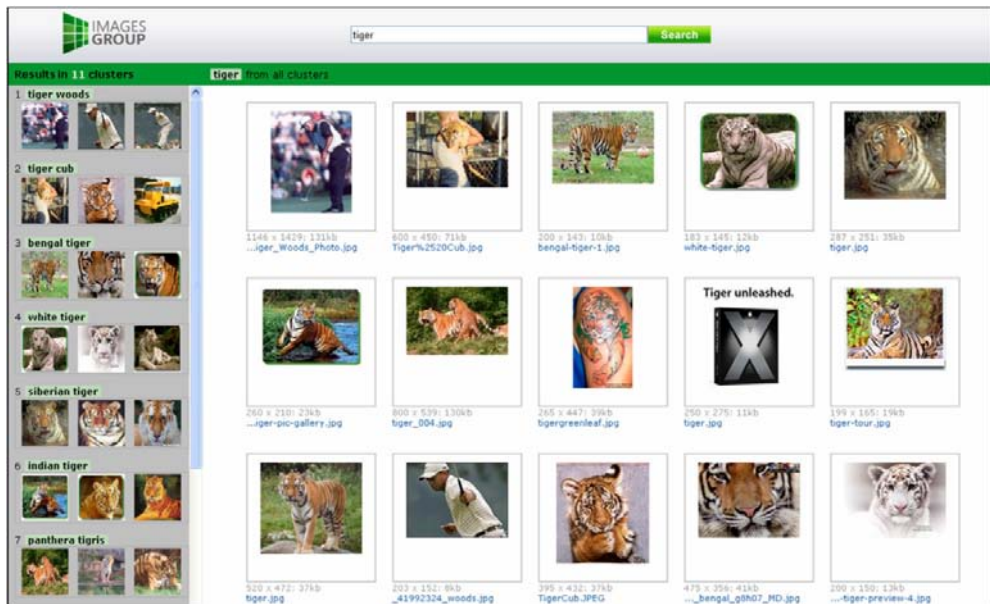


Figure 19. Example of the IGroup GUI with identified clusters (left), taken from (Wang, Jing, He, Du, & Zhang, 2007).

## VisGets

The VisGets exploration interface allows users to explore information based on three prominent dimensions in the news domain – time, space and topic (Dörk, Carpendale, Collins, & Williamson, 2008). The interface effectively corresponds to an advanced faceted browser where users can select the time using a histogram timeline widget, location via an interactive map and topic via a tag cloud (Figure 20).

Advantages of VisGets include intuitive visualization of individual facets, which enable users to select restrictions in a way natural to a given facet type (e.g., time on a timeline, location on a map, topic via a tag cloud). VisGets also provides orientation support in line with mSpace by providing highlight previews of queries and improves user understanding of the information domain by showing many samples of results.

However, VisGets works with a custom made back-end which crawls the Web and collects news articles and is thus not usable for Semantic Web exploration as it does not use RDF metadata and also provides little in terms of user customization or personalization.

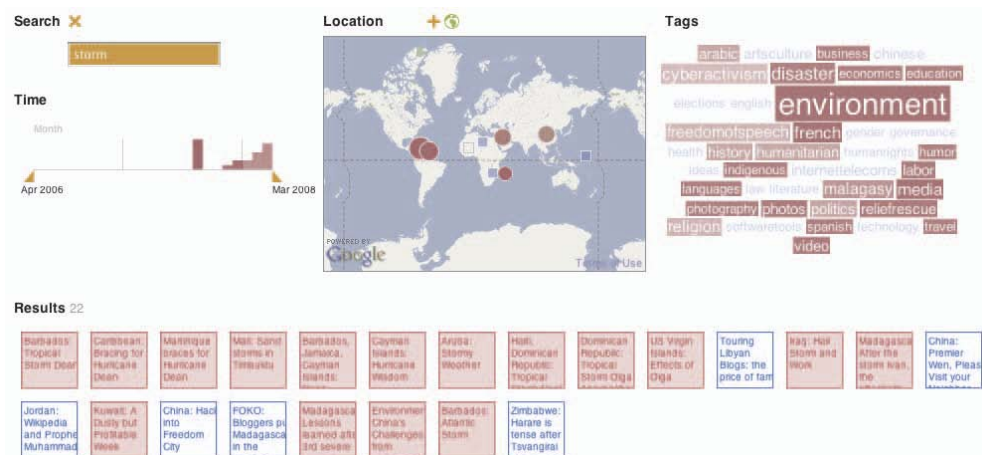


Figure 20. Example of the VisGets interface showing the timeline (left), location (centre) and topic facets (right), taken from (Dörk, Carpendale, Collins, & Williamson, 2008).

## Microsoft Pivot

Microsoft Live Labs presented Pivot in 2010 and described it as “an experimental technology that allows people to visualize data and then sort, organize and categorize it dynamically”<sup>16</sup>. Pivot is a view-based search tool, which takes advantage of exploratory and visual search principles to provide dynamic information organization and visualization capabilities. Microsoft demonstrated Pivot via multiple showcase applications<sup>17</sup> in various domains (e.g., car search), which enable users to select items from several simple categories (facets) and render the results interactively via Deep Zoom which supports quick previews of even detailed images (Figure 21).

Despite presenting Pivot as a practical user interface widget with highly interactive and animated transitions, the query engine and the logic behind is less practical. Pivot normally expects data to be described by a CXML file and the associated image collection, both available on the Web as static files. As such, the CXML file must contain the whole collection and only a flat item categorization is supported (i.e., each item is directly associated with a set of faceted categories).

Although Pivot also supports dynamically generated collections (described as hard to create in its documentation), this requires complex server infrastructure not provided by Pivot itself. Consequently, Pivot is a practical client-side user interface widget which can serve as a nice front-end to a compatible server-side query engine, but cannot provide any meaningful (Semantic Web) browsing experience by itself. This can only be provided in conjunction with a sophisticated back-end service.

<sup>16</sup> Pivot press release: <http://www.microsoft.com/presspass/features/2010/feb10/02-11pivot.mspg>

<sup>17</sup> Pivot showcase applications: <http://www.microsoft.com/silverlight/pivotviewer/>



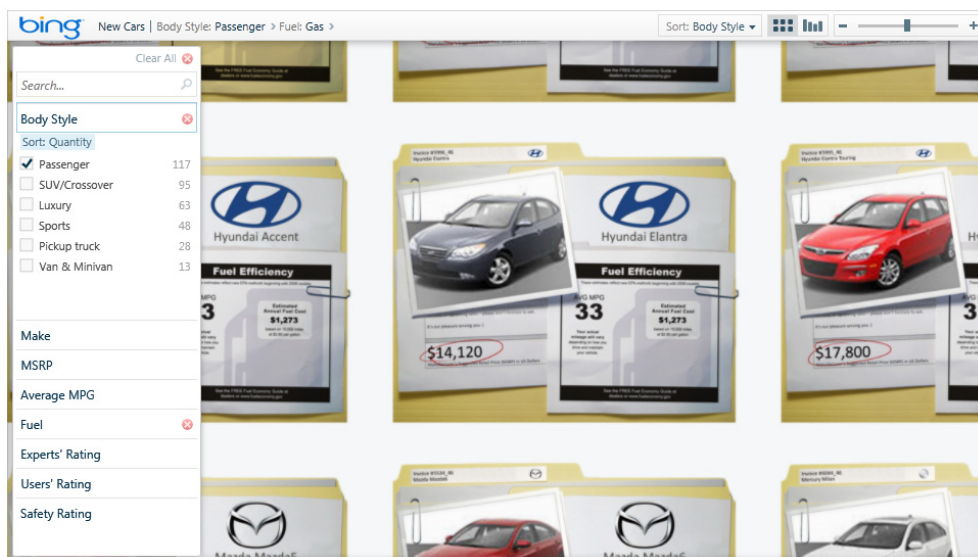


Figure 21. Example of the Microsoft Pivot interface showing simple facets (left) and search result thumbnails with properties (right).

### 3.5 Summary of current exploration approaches

The work described in the previous sections captures only a portion of the most relevant research done in the field of faceted browsers and related exploratory search approaches. We have shown the progress made in effective information exploration by showing a timeline of approaches from 2003 to 2010 (see Table 1):

- Flamenco, a faceted browser for images, pioneered view-based search even before the advent of Exploratory search and major Semantic Web standards;
- Ontoviews and mSpace, faceted browsers for multimedia collections, which took advantage of emerging Semantic Web technologies (e.g., RDF) and exploratory search to provide end-user exploration of data repositories;
- Tabulator, a generic table based data viewer, which built upon Semantic Web technologies and extended browsing support to distributed Linked data;
- TagSphere and VisGets focused on visualization and user interaction to provide advanced information exploration capabilities in dynamic collections;
- Pivot, effectively a user interface widget, providing a product-grade faceted data browser for arbitrary information collections conforming to a given format.

Most of the examined solutions offered at least limited support for Semantic Web data in the form of RDF/RDFS ontologies, though several had no support at all. This were most notably Flamenco and RB++ as older approaches, and the newer exploratory approaches

TagSphere, IGroup, VisGets and Pivot as generic approaches aimed at the general Web. Still this support was more in line with the internal representation of data rather than capability to browse arbitrary RDF data available on the Web with the exception of Tabulator, which was specifically designed to browse the distributed Linked data cloud.

Table 1. Summary of main properties of selected exploration approaches.

	Flamenco	OntoViews	RB++	mSpace	/facet	BrowseRDF	Tabulator	TagSphere	IGroup	VisGets	Pivot
Overview											
Created	2003	2004	2005	2005	2006	2006	2006	2006	2007	2008	2010
Browser type	faceted						generic	query-by-example		faceted	
Semantics	no	RDFS	no	RDF	RDFS	RDF	RDFS	no	no	no	no
Domain	multimedia digital library					generic data		images	images	news articles	generic data
Dynamic content support											
Link generation	✓	related links	✗	✗	✓	✓	✓	✗	result clusters	✗	✗
Facet ranking	✗	✗	✗	✗	✗	statistical	✗	✗	✗	✗	✗
Facet generation	✗	✗	✗	✗	✗	static, direct	✗	✗	✗	✗	✗
View adaptation	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Search support											
Keyword search	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Faceted search	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
Query by example	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
Result exploration support											
Faceted navigation	✓	✓	✓	✓	✓	✓	✗	✗	clusters	✓	✓
Graph navigation	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Related results	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Visualization support											
Text	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Images	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓
Tables	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓
Timeline	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗
Maps	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗
Graphs	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Content-based	✗	✗	✗	previews	✗	✗	✗	✗	✗	✗	✗
Advanced feature support											
Editing	✗	✗	✗	tags	✗	✗	✓	✗	✗	✗	✗
History tracking	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Personalization	✗	✗	✗	manual	✗	✗	✗	✗	✗	✗	✗
Recommendation	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Revisitation	✗	✗	✗	favourites	✗	✗	✗	✗	✗	✗	✗
Social networking	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗

Many of the examined approaches were aimed at exploration of closed information domains, such as digital libraries with multimedia content (e.g., video, audio, images), and only later approaches worked with open information spaces and generic web data.

From an exploratory search perspective, we examined the properties of individual approaches with respect to search, result exploration and visualization support also considering their handling of dynamic content and optional advanced features presently expected by users. Only about one half of the approaches supported dynamic link generation during result browsing, mostly corresponding to related results as asserted in the dataset. Only IGroup computed result clusters based on estimated similarity at runtime. With the exception of BrowseRDF, which supported statistical facet ranking and static facet generation, none of the examined advanced faceted browsers supported facet ranking or generation. Moreover, all approaches used static views as none supported view adaptation based on user context.

Search support was straightforward, except Tabulator all approaches supported keyword-based search; faceted approaches also supported faceted search while content-based approaches TagSphere and IGroup worked with similarity based query-by-example as did Tabulator which supports queries based on selected graph patterns.

Faceted navigation was standard in most approaches although, VisGets only supported three main facets – time, location and topic, while the result clusters in IGroup could also be considered a single (flat) facet. Neither of the approaches however supported interactive graph-based result exploration nor the exploration of related search results.

Textual and table-based information presentation was prevalent, but was often supplemented with previews or images in multimedia domains. More sophisticated visualizations included interactive facets in VisGets (maps for locations, tag clouds for topics) and timelines for time indexed data. Content-based or graph-based visualization was not present with the notable exception of mSpace, which offered limited content-specific presentation via previews (e.g., a short preview of a song).

Lastly, support for features typically expected by today's users as major parts of the overall user experience was virtually non-existent. Only the mSpace faceted browser supported tagging of content by users, manual personalization (i.e., reordering of columns by users), favourites and social networking. The only other exception was Tabulator, which claimed to support editing but in a hardly practical way. This can be best explained by the fact that mSpace development still continues, while Tabulator is also getting some minor updates despite its authors' claims that "The Tabulator project has led to many more questions than it has answered." (Berners-Lee, et al., 2006).

Recent work includes Microsoft Pivot for Silverlight, effectively a client-side visualization widget for faceted browsing, which offers a generic standalone user interface control for third-party applications. Its primary use lies in visualization of static faceted data collections although with a suitable server-side infrastructure, it could be used to browse fully dynamic collections generated at runtime. A compatible server-side search platform might be, for example, Apache Solr<sup>18</sup> – an open source enterprise search platform combining full-text search, faceted search and dynamic clustering of non-semantic data.

---

<sup>18</sup> Apache Solr project: <http://lucene.apache.org/solr/>

Similarly, OpenLink Virtuoso<sup>19</sup> might be used as a database and querying back-end for both textual and semantic data.

Based on these findings, the area of exploratory search and faceted browsers with specific focus on

- *dynamic content* (e.g., link, facet generation and ranking, view adaptation),
- *multi-modal search, advanced visualization and result exploration integration,*
- *personalization, recommendation, history tracking and revisitation support*

has so far not been sufficiently explored and offers great opportunities in combining and extending the aforementioned approaches into a seamless search and browsing solution for both legacy and semantic web content.

---

<sup>19</sup> OpenLink Virtuoso project: <http://virtuoso.openlinksw.com/>

## 4 Framework for Exploratory Search

---

In the previous chapters, we have defined our high-level goal to maintain and improve the usefulness of the Web as a global information space. Next, we investigated the current challenges in both Legacy Web and Semantic Web search and navigation in chapter 2, and provided an overview of the current state of the art in exploratory search approaches in chapter 3. Specifically, we confirmed that effective exploratory search is still an open issue and that based on the current lack of integrated and personalized exploration approaches, these pose a good direction for further research.

Thus to achieve our high-level goal, we opted to address several open research questions that we identified based on our review of related work:

- *Improved exploratory search* by integrating keyword-based, view-based and content-based search with advanced visualizations of search results and their relations thus facilitating visual query construction and interactive resource exploration.
- *Improvement of end-user browsing experience* via dynamic personalization and navigation, orientation and revisitation support focused on the specific needs of individual users and overall usability.
- *User interface generation* based on semantic metadata describing the schema of the presented information space aimed to achieve a smooth user experience also accounting for dynamic changes in the information space

In line with these questions, we believe that Semantic Web principles would address many of the aforementioned issues. Thus our aim is to solve the chicken-and-egg problem of the Semantic Web (no applications without data, no data without applications) by facilitating Semantic Web adoption by *providing end-user grade exploratory search experience in the Semantic Web* with focus on specific challenges as identified in section 2.2.

### 4.1 Example user scenario

First we describe our browsing approach in terms of user experience, i.e. how a user—Alice—would employ its capabilities for an exploratory search task, and next elaborate on selected aspects of its design.

Alice needs to find papers relevant to her research so she starts her session using the general keyword-based search of our browser. Somewhat expectedly, most of the top results appear to be her own papers. Although normally Alice would try to *guess* better keywords which others might use to describe relevant results, she instead takes advantage of the search by example capability of the browser to see similar/related results and rank them via a positive example selecting one of her better papers. The browser returns a (large) mixed set of her papers, other papers and also various somewhat related results as

returned by a back-end search engine. In order to filter out her own papers, she places a negative faceted restriction saying *not my papers*.

Looking at the results, Alice sees papers she had already read, digital library pages of newspapers, bookstore sites, conference programs and some broken links. To reduce the number of irrelevant results, she employs negative search by example saying *no shops and programs* (i.e., ranking those results low) while also restricting the results set to *not older than 4 years*. The browser returns an interesting looking paper on a digital library page warning Alice that she does not have an account to access the full paper. Since the paper is effectively unavailable, Alice rather explores another paper (described only by a bibliographic reference), which was recommended by the browser based on her social network data – Alice knows the authors personally.

This gave Alice an idea, which she decides to explore – how could she select all papers by all authors she knows to work in her field and the papers they reference? She starts by using nested facets to restrict the results to papers, then to papers that are authored by people she knows. Alice also adds papers authored by people whose papers are referenced by the people she knows. Lastly, since the browser tracks her profile, it recommends her to hide all the resources she had already seen in the past leaving her with a good set of results from relevant authors.

## 4.2 Design objectives and main principles

To achieve the aforementioned functionality, we integrate and extend various approaches from different research areas, which resulted in a *strongly multidisciplinary work* including Semantic Web, Adaptive Web, Social Web, information visualization, exploratory search, information retrieval, human-computer interaction. We devised a *comprehensive faceted exploration approach for the Semantic Web* taking advantage of exploratory search principles, personalization and social aspects to achieve our primary design objectives:

- *Visual query construction* by combining keyword search with faceted search and query-by-example thus also supporting query refinement.
- *Information overload prevention* by recommending relevant content while hiding less relevant content (e.g., facets, restrictions, result attributes).
- *Guidance support* via navigational shortcuts, which streamline navigation in deep/complex faceted hierarchies (e.g., restriction recommendation).
- *Orientation support* by showing additional information/cues simplifying user decisions about further navigation (e.g., tooltips showing future facet contents).
- *Improved response times* due to selective processing of facets and restrictions, since advanced (semantic) approaches proved to be “time consuming”.
- *Universality and flexibility* – suitability to different/changing application domains facilitated by (semi)automatic user interface generation.

To achieve these objectives, we take advantage of these main principles:

- *Semantic information space representation*, e.g. an ontological repository, where both metadata describing the structure of the information space and data are represented by ontologies (e.g., in RDFS or OWL as defined by W3C). Thus we assume an existing description of classes, individuals, relations and attributes describing a particular information domain – a domain ontology. We also employ a user ontology which stores the user models describing individual users' characteristics, and an event ontology which is used to preserve the semantics of user actions during logging.
- *Multi-paradigm exploration*, which integrates view-based faceted search with content-based (query-by-example) search and traditional keyword-based search to provide users with the most suitable means to create queries or navigate the information space. It also includes various visualization and navigation options for the browsing of search results such as result lists, result attribute tables and attribute/thumbnail matrices, incremental graph visualization and history visualization for revisitation and orientation support.
- *Adaptive view generation*, which facilitates the generation of user interfaces necessary for exploration, accommodates for the dynamics of the information space and preferences of individual users.
- *Personalized recommendation* to address information overload, provide guidance during complex information seeking sessions and during revisitation tasks.

### 4.3 Semantic information space representation

We work with semantically enriched information spaces, e.g. an ontological repository, where both metadata about the structure of the information space and data are represented by ontologies (e.g., in RDFS or OWL). Thus our approach assumes a description of classes, individuals, relations and attributes describing a particular domain. For example, in the digital image domain, *di:Author* and *di:Photo* are classes; *di:Author\_1* and *di:Photo\_1* are individuals, while *di:createdBy* is a relation between *di:Photo\_1* and *di:Author\_1*. Similarly, *di:viewedCount* equaling *10* is an attribute of *di:Photo\_1*, also defining the domain of the attribute as the class *di:Photo* and its range as an *xsd:int* (see Figure 22).

As shown in the above example, a domain ontology as defined by W3C contains a detailed standardized description of classes, properties (relations and attributes) and the used data types, effectively defining a data model. Ontologies can also be populated with individuals, which conform to the specified domain model and materialize it in instances of classes and properties. Note that the ontology is in fact an oriented graph where nodes represent individual resources.

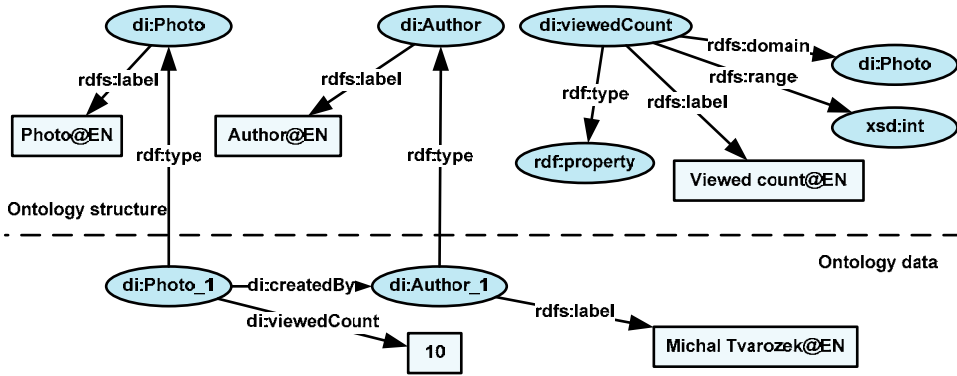


Figure 22. Example of a simple domain ontology for the digital image domain. Metadata describing the domain model are shown at the top; individuals representing data are shown below. Round nodes denote complex resources with URIs, rectangular nodes denote literals.

We employ ontological data representation taking advantage of OWL ontologies to define:

- The *domain ontology*, which describes domain concepts, the relations between them and their attributes. It contains metadata that describe the structure of the domain model (i.e., classes and properties) as well as actual domain data (i.e., instances). For example, in the scientific publications domain, it would describe authors and publications.
- The *user ontology*, which describes the estimated characteristics and preferences of individual users used for personalization.
- The *event ontology*, which facilitates user modelling by describing the events that occur in the faceted browser during user interaction.

We often refer to *information artefacts* or *resources*, which should be understood in the Semantic Web context, where a resource is basically anything with a URI and can denote for example a person, event or a photo. As shown in the example above, resources can link to other resources (information artefacts) while also linking to information in the legacy Web, such as web pages. Consequently, the fact that we use resources in a more general notion is more of an advantage than a limitation. The need for metadata is not technically much stronger than in existing content management systems, but rather focused on a common shared format (e.g., RDF or OWL) instead of the proprietary formats of existing content management systems.

## 4.4 Multi-paradigm exploration

We extend the opening-midgame-endgame approach to faceted browsing originally proposed in Flamenco (Yee, Swearingen, Li, & Hearst, 2003) into a comprehensive multi-



paradigm exploration approach. We add user support for the individual stages of the information seeking process and populate them with additional complementary approaches to facilitate end-user grade exploration experience (see Figure 23).

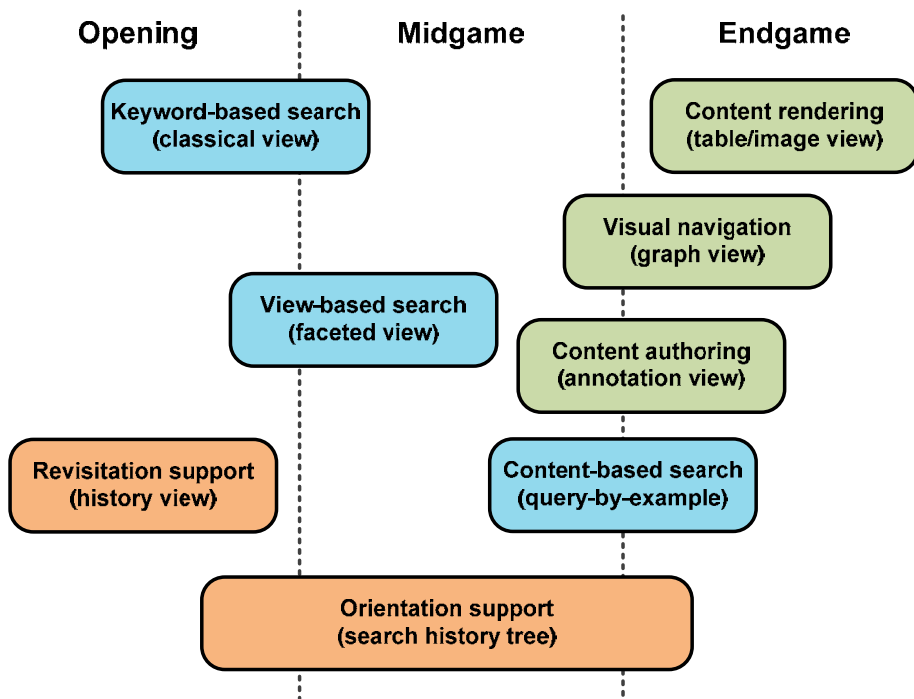


Figure 23. Overview of our multi-paradigm exploration approach showing the scope and applicability of individual sub-approaches to specific stages of the exploration process. Search approaches span primarily the opening and midgame (blue), content viewing, annotation and browsing approaches focus on the endgame (green), while support approaches span all stages (orange).

## Opening

We populate the opening stage with three views that can be used to initiate and exploratory search session:

- Our *classical view* augments the traditional keyword-based search window with a tag-based overview of the information space content. The tag cloud normally corresponds to the different information artefact types that are present in the information space (e.g., photos, people, places, events). Additional tag clouds, corresponding to e.g. the recently added items, popular items or important information artefacts, can also be shown to improve user orientation and quick access to information.

- The *faceted view* corresponds to our faceted browser interface without any selections. It is mostly used when we already have some information about user preferences and are thus able to provide a personalized set of initial facets for exploration. The faceted view also provides a search box for keyword queries and starts by showing a set of random search results to give users a glimpse of the information space contents.
- The *history view* supports information revisitation via the Semantic history map, which semantically organizes a user's search and browsing history. Here the users can see an overview of explored topics during past exploration sessions and quickly find and rediscover previously visited resources.

## Midgame

After users make their initial query they proceed to the midgame stage where they continuously refine their query and explore the search results. We populate the midgame stage primarily with our enhanced faceted browser by extending the *faceted view* from the opening with:

- Two *result views* (*list view* and *matrix view*) that provide result browsing.
- The *search history tree*, which provides orientation and history support.
- The *graph view*, which enables users to interactively explore the properties and relations between individual information artefacts, and lies somewhere between the midgame and the endgame as it can equally be used to find information related to specific resource or to freely explore the collection.

The faceted browser also includes support for keyword-based search in information artefacts and also in individual facets and restrictions. During the midgame, users can also use *query-by-example* either by searching for *similar information artefacts* (similarity evaluated via external tools) or by positively / negatively rating individual items and supplying the ratings to an external evaluation agent which in turn can build a user preference profile and supply a list of suitable results.

## Endgame

While during typical fact retrieval, user sessions end with the endgame, in an exploratory context, the opening-midgame-endgame process is of a more iterative nature and allows users to return to a previous stage. We thus populate the endgame with tools that enable the user to get a better understanding of the information space, to shape it or to simply view its contents in a natural way:

- Our nested *table view* displays the properties of individual information artefacts, which again can point to other information artefacts (thus a nested table). In order to present information resources naturally, we propose to employ content type specific views.

- Our specialized *image view* enables users to view the photos similarly to popular web-based photo galleries (e.g., supports image manipulation features such as zoom, rotate or slideshows).
- The *graph view* enables users to interactively view the information artefacts as a graph showing their relations and attributes. Again, the users can navigate the graph in various ways including node expansion, manual node relocation, node hiding, zooming or panning.
- The *annotation view* (after logging in) allows users to see a list of existing or optional properties of individual information artefacts and edit them, e.g. by selecting some of the predefined values or by creating entirely new ones.

## 4.5 Faceted browser extensions

Since the faceted browsing paradigm is principal to our approach, we extend the request handling of faceted browsers with additional stages that perform specific tasks. We extend search results processing with *result recommendation* which includes support for result annotation and adaptation (Figure 24, centre right). We employ external tools that evaluate the relevance of individual search results, e.g., by means of concept comparison with the user model. Subsequently, we reorder search results or annotate them with additional information. For example, in the domain of scientific publications, we can display the suitability of an article, based on its estimated relevance to the user's research, as background colour or via emoticons.

To facilitate automatic user modelling, we log events that occurred as results of user interaction with the browser and the current logical display state of the browser via a user modelling server (Figure 24, bottom right). The logging of user actions is closely tied to updates in the user model and subsequent updates of the relevance model (Figure 24, top left), which is crucial for our faceted personalization engine.

Facet processing is extended with *facet recommendation*, which includes the adaptation, annotation and recommendation of facets and restrictions (Figure 24, bottom left), which improve orientation and guidance support, reduce information overload and alleviate some disadvantages of faceted classification. If the set of available facets is insufficient, we use dynamic facet generation to add new facets at run-time on a per user basis (Figure 24, centre left) thus allowing the user to refine the search query and improving support for open information spaces.

Facet generation examines the metadata describing the schema of the presented information space and identifies specific (predefined) patterns that correspond to different facet types and generates the corresponding widgets and mappings for their use in the graphical user interface of the browser. We also take advantage of this metadata to generate the result overviews (i.e., list view, matrix view), which display relevant properties of information artefacts.

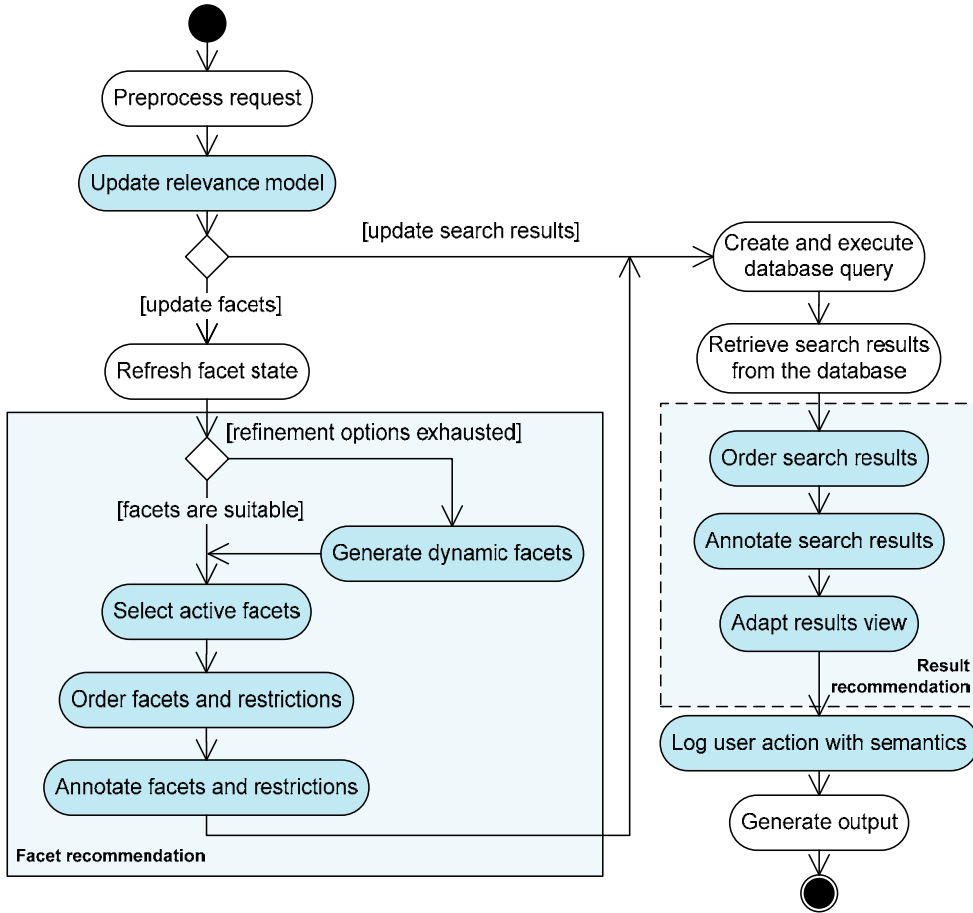


Figure 24. Request handling of our faceted semantic browser, extensions are shown in blue.

Lastly, we integrate the faceted view with specific views that augment its functionality as outlined in section 7. Users can start graph exploration sessions by exploring the properties of specific search results (graph centred on the result) or view resources discovered in the graph view via the faceted browser. The faceted browser can also be used for batch editing of information artefacts via the annotation view, where users first find and select resources via the faceted browser and next (batch) edit their properties via annotation view.

## 4.6 Validation overview

Although some of the presented research was performed individually, a large part of it was also performed as part of several research projects conducted at the Institute of Informatics and Software Engineering, Slovak University of Technology (see Appendix A).

While many information retrieval methods can be evaluated statistically, e.g., by computing precision and recall statistics, in many cases such exact evaluations cannot be performed for approaches dealing with user interaction and user interfaces for adaptive systems. The evaluation of such approaches can be done via experiments described as user studies, such as (Yee, Swearingen, Li, & Hearst, 2003) or (Wang, Jing, He, Du, & Zhang, 2007). Ideally, evaluation methodologies are well-known, defined beforehand and performed on several variants of a test system against a baseline system which might either be an existing system suitable for benchmarking or a system made using the best or most common features of comparable systems (Yee, Swearingen, Li, & Hearst, 2003). Furthermore, layered evaluation principles should be employed to effectively separate evaluation of individual stages of the information processing process – data collection, data interpretation, user modelling, adaptation selection and adaptation application (Paramythis & Weibelzahl, 2005).

However, the novelty of the exploratory search field and the general immaturity and unavailability of methodologies for task design and browser evaluation makes exact analytical validation of user-centred exploratory search approaches difficult if at all possible (Kules, Capra, Banta, & Sierra, 2009).

Consequently, we aim to evaluate our approach via a mixed set of exact experiments, practical user studies and proof of concept validation of individual approaches. We developed and performed experiments with two prototypes of our faceted semantic browser *Factic* in three different application domains (see Appendix B for a detailed description of the evaluation environment and Appendix C for a description of the used domain and user models):

- online job offers (project NAZOU),
- scientific publications (project MAPEKUS) and
- image collections (project PeWePro).

### First *Factic* prototype in projects NAZOU and MAPEKUS

Our first *Factic* prototype was developed as part of the evaluation framework for projects NAZOU and MAPEKUS. The purpose of *Factic* was to *evaluate the personalization aspects of our approach and to serve as a major integration platform for other tools* (realizing different approaches) within the evaluation framework. Consequently, it was strongly tied to other parts of the evaluation framework, mainly the user modelling back-end and also other tools that worked as optional plug-ins improving its functionality (see Figure 25).

Our *Factic presentation tool* was the main user interaction tool in the personalized presentation layer of the common portal framework where it provided query construction, execution and result exploration functionality. *Factic* forwarded its internal user events to the user modelling back-end represented by our *SemanticLog* tool, which gathered evidence of user actions which was processed by the *LogAnalyzer* tool realized by Michal Barla (Barla, Tvarožek, & Bielíková, Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems, 2009).

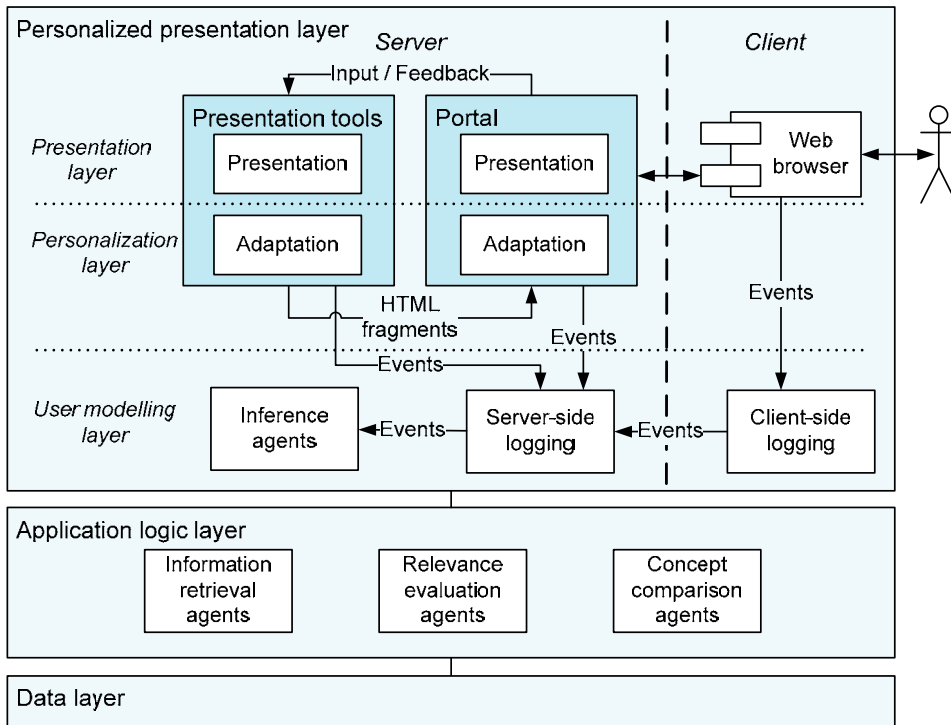


Figure 25. Architecture of the evaluation framework in projects NAZOU and MAPEKUS. The main portal incorporates the underlying presentation tools, such as *Factic* (top left), which in turn communicate with the user modelling layer comprised of logging and inference agents (centre). Individual plug-ins that enhance the functionality of presentation tools are within the application logic layer and include information retrieval agents, and relevance and similarity evaluation agents (bottom).

To provide better user experience, *Factic* was integrated with these additional tools:

- *CriteriaSearch*, which provided users with advanced search capability and also served for similarity evaluation;
- *TopK-UpreA-IGAP* tools chain, which evaluated explicit user feedback into abstract user characteristics and provided content-based similarity search by returning the K best results based on the inferred characteristics;
- *ConCom* and *JDBSearch*, which also served for the evaluation of aggregate similarity and subsequent search for the best (i.e., most similar) results.

The implementation of the first *Factic* prototype was strongly constrained by the limitations of the evaluation framework, which was restricted to open-source technologies. As a result, Java had to be selected as the implementation language, Apache Cocoon as the

corresponding web application framework and Sesame and MySQL as the database back-end. In practice, the use of Java and open-source solutions turned out to be a bad choice (although we had little choice) due to low performance of the XML pipeline based Apache Cocoon framework, Java memory limitations and the overall situation with open-source solutions (i.e., lacking or inaccurate documentation, missing functionality, to-be-done features).

Specifically the Sesame ontological repository had poor scalability and due to Java memory limitations had to be switched to MySQL backed storage instead of in-memory storage what further decreased performance for larger datasets. Consequently, we were practically limited to experiments with smaller manually created data sets of job offers or subsets of the entire publications dataset.

Since our goal was to evaluate the personalization aspects of our solution, we devised the domain models and the corresponding user models with personalization requirements in mind. Thus the domain models describe the respective application domains in detail and also provide several classifications of information that describe search results and can consequently be used as user characteristics (see Appendix C). The job offer datasets contained hundreds of job offers populated mostly by manually by users or augmented with specialized tools. Larger datasets were impractical due to two reasons:

- Scalability of the Sesame v1.2.x database back-end was poor, larger datasets had high memory requirements necessitating the shift towards MySQL backed storage what further degraded practical performance.
- Population of larger datasets manually was not possible due to the amount of effort required while at the same time fully automated approaches could not populate enough information for evaluation to make sense (i.e., they could only populate the title and location of a job offer, but could not classify it using the available hierarchies neither provide details such as salaries).

The description of the specific experiments performed to evaluate the usefulness of our facet personalization approach in the job offers domain via a user study and to gathered feedback on its design is given in chapter 5 (Tvarožek & Bieliková, *Personalized Faceted Navigation in the Semantic Web*, 2007).

## Second Factic prototype in project PeWePro

We developed the second *Factic* prototype to address the shortcomings of the first prototype and to evaluate additional aspects of our approach including interface generation and multi-paradigm exploration (Tvarožek & Bieliková, *Reinventing the Web Browser for the Semantic Web*, 2009). We specifically aimed to improve on the implementation aspects with respect to memory consumption, performance and overall support for the used environment. We implemented the second *Factic* prototype in Microsoft Silverlight 3 with C#/NET as the implementation language. We also changed

the static, heavily server-side architecture into a client-side lightweight browser in Silverlight and a set of server-side services (see Figure 26).

We developed these server-side services:

- The *Factic* faceted search engine service, which performs faceted queries over the ontological repository and also generates the corresponding facets.
- The *Steltecia* repository access service, which provides generic high-level read/write access to the underlying ontological database.
- The *SemanticLog* logging services, which performs logging of user actions.
- *Support services*, as required by plug-ins to provide additional functionality such as web page screenshots.

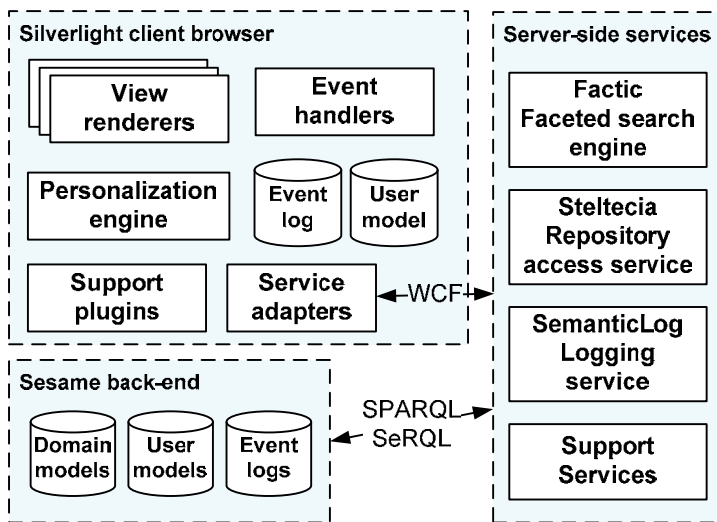


Figure 26. Architecture of the second *Factic* prototype in project *PeWePro*. The server includes web (WCF) services for faceted search (*Factic*), ontological repository access (*Steltecia*), and event logging (*SemanticLog*) for global statistics tracking (right). All services store their data in a common ontological repository in *Sesame*.

In addition to the faceted browsing functionality, we also integrated these additional plug-ins with the client-faceted browser:

- *Photo browser*, which allows users to view the photo collection in an intuitive way, similar to popular web-based photo galleries, such as Flickr or Picasa.
- *Graph view*, which allows users to explore the relations between resources via interactive graph-based visualization.
- *Annotation pane*, which allows authorized users to (batch) edit the properties of selected resources via a generated form widget using the *Steltecia* service.



- *Search history tree*, which allows users to view the session history (e.g., queries and query modifications, result exploration) and quickly return to a previous state.

The new architecture and evaluation environment enabled us to realize a large part of the functionality in the client browser which resulted in a decrease in the number of required server-side calls and thus also decreased network latency and application response time. This was further improved via asynchronous request processing and caching on both sides.

To improve database performance, we switched the database back-end from Sesame v1.2 to the next version of Sesame 2 which already supported the W3C standard query language SPARQL in addition to the proprietary SeRQL language. This enabled us to work with standard repositories which support the SPARQL endpoint interface (i.e., lifted the restriction on using Sesame), although due to the read-only nature of SPARQL, we still had to use SeRQL for database updates. As a side effect of Sesame 2, SPARQL queries and query aggregation, the overall query performance was much improved.

We used the second *Factic* prototype in the digital image domain to validate additional aspects of our solution – facet and result overview generation and the novel graph-view exploration approach. The description of individual experiments is given in chapters 6 and 7 after elaborating on individual approaches.



## 5 Personalized Recommendation

---

We employ personalization to improve user orientation in the information space, provide user guidance and reduce information overload. Our goal is mainly to provide users with additional information that would empower them to make their own decisions more effectively instead of relying only on automatic adaptation which might not work as expected. Nevertheless, we also provide direct adaptation, for example, the hiding of less relevant facets or the selection of only the most relevant attributes for visualization (others available on user demand).

Our personalized recommendation approach is mostly used during interface generation to evaluate what things should be generated and next how they should be generated and personalized to suit the need of the current user and his context (e.g., task at hand, position in the information space). We thus primarily:

- *Recommend facets* by reordering them and hiding less relevant ones thus reducing the number of facets to an acceptable level (otherwise it is possible to generate and thus display too many facets).
- *Recommend facet restrictions* and annotate them with additional information providing user guidance and navigation support while reducing the number of clicks necessary to reach specific search results.
- *Select and order result attributes* shown in result overviews based on their estimated importance to the user. We also can select the most suitable view to present the results in or show/hide specific attributes of results in our graph exploration view.

Our adaptation approach relies on an implicitly created user model acquired by continuous tracking of user actions within the browser and their successive evaluation into a user model, which in turn is used to derive a relevance model for resources in the processed information space. The key aspect of our user modelling approach is the *semantic logging* of user actions as they occur in the faceted browser preserving their semantics (e.g., what exactly happened, what resources were affected) as opposed to traditional web server logs, which only store implicit information in request URLs and lack the detailed information required for quick in-session user characteristics estimation. Each logged event uses our event ontology to specify the semantics of the respective user action and also references the domain and user ontologies as required.

As opposed to most existing approaches, we perform *personalization primarily on the client side* (i.e., in the client browser), which has two benefits:

- Personally sensitive data is kept entirely on the client side thus preserving privacy. Optional server-side user modelling and statistics tracking can be enabled to further improve user models and provide social information to authorized users.
- Server-side services (e.g., search engines or repositories) do not necessarily need to have support for adaptation as this is performed by our browser on the client side,

thus providing personalization for all information resources without additional cost as no server modifications are necessary (providing that standardized semantic metadata are available).

## 5.1 User characteristics model

Typically adaptation is performed based on a user context model, which describes in detail the characteristics and preferences of users as well as the time, location and properties of the device and network they use. This is especially important for the usability of mobile applications, where screen space, network bandwidth or input capabilities are limited.

The information used by an adaptive system to perform user adaptation can be acquired by various means and from different sources. The user can either explicitly enter the information (e.g., into a form) or user characteristics might be acquired implicitly by observing user behaviour (e.g., automatic user action logging with successive data mining). Either way, the kinds of data acquired belong to one of these categories:

- *Personal characteristics* describe the user's knowledge about specific concepts in case of educational systems, the user's background, preferences or individual traits. These can be subdivided into (Brusilovsky, 1996):
  - *Knowledge* of the respective subject, often used via an overlay model.
  - *Goals* or tasks related to the user's reason for using the system.
  - *Background and experience* describing the background of the user such as his profession, education, work experience, and his/her experience with the structure and usage of the adaptive system.
  - *Preferences* describing the general preferences (interests, likes or dislikes) of the user, which cannot be directly inferred.
- *Environment characteristics* describe the place where the user is or the current time.
- *Device characteristics* describe the technological aspects of the device, which the user uses to access the system (e.g., screen resolution or network bandwidth).
- *Social characteristics* describe the relationships with other users.

Device characteristics are especially important for mobile applications where the available resources in terms of computing power, screen space, control options (keys/joystick, keyboard/touch screen) and power are very limited. For example, the typical screen size of a mobile phone is about 100 times smaller than that of a typical desktop computer, which results in completely different usage requirements (Smyth & Cotter, 2004).

Since an updated user (context) model is required for successful adaptation, a continuous user modelling process must be used to constantly update the model based on user interaction. We do this automatically by employing implicit feedback based on the observation of user behaviour (Barla, Tvarožek, & Bieliková, Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems,

2009) using an integrated framework for user evidence acquisition and preference inference (Andrejko, Barla, Bielíková, & Tvarožek, *Softvérové nástroje pre získavanie charakteristík používateľa*, 2006). We focused primarily on the acquisition of user actions performed during exploratory sessions, while the actual evaluation of log records via specialized inference agents was done by Michal Barla.

We also forward explicit user feedback (e.g. result ratings) to external user modelling and search tools that in turn return results tuned to the estimated user characteristics (Gurský, Horváth, Novotný, Vaneková, & Vojtáš, 2006).

## 5.2 Model for relevance evaluation

The user modelling back-end provides us with several sources of adaptation, which we employ with different weights depending on how closely related they are to the current user task (Figure 27):

- *In-session user behaviour* – user navigation, facet and restriction selection during the current user session (i.e., user clicks). Frequent use of specific items indicates higher relevance to the current task or user interest in the corresponding domain concepts. For example, if *ConferencePaper* is selected as the publication type, showing user interest, additional facets associated with the domain concept *Conference* are likely to be generated in order to allow users to refine queries.
- *Short/long term user model* – user characteristics acquired during multiple sessions described by their *relevance* to the user and the *confidence* in their estimation in the range  $\langle 0,1 \rangle$ . High attribute (restriction) relevance in the user model denotes good choices for facet generation and restriction recommendation.
- *Similar/related user models* are assumed to belong to users with similar needs and are thus used for relevance evaluation if user specific data is unavailable or has low confidence. Social user context can be exploited by assigning custom weights to specific relations between users resulting in social recommendation. Moreover, if usage data about other users are “publicly” available, users might directly browse the trails of their peers (e.g., see what images their friends viewed or what papers their colleagues downloaded).
- *Global usage statistics* computed from the overall relevance and usage of individual domain concepts (e.g., facets, restrictions, target objects – be it images, publications or job offers) from all user models. The overall “popularity” of facets and restrictions increases the likelihood of their recommendation for a specific user, especially if his or her specific preferences are unknown or have low confidence.

Let  $L_U(X) = \text{relevance}_U(X)$  be the local relevance of facet  $X$  for user  $U$ . We define  $C_U(X)$  as the cross relevance of  $X$  determined as the average local relevance for all users  $V$

weighted by their similarity to user  $U$  (1), and  $G(X)$  as the global relevance of  $X$  defined as its mean local relevance for all users (2).

$$C_U(X) = \frac{\sum_{V \in \text{users}} (\text{dist}(U, V) * L_V(X))}{|\text{users}|}, U \neq V \quad (1)$$

$$G(X) = \frac{\sum_{V \in \text{users}} L_V(X)}{|\text{users}|} \quad (2)$$

To evaluate the user similarity weight  $\text{dist}(U, V)$  we employ external concept comparison tools. Alternatively, similarity can be evaluated as the weight between users in a social network or the sum of differences between relevance of specific concepts between users (3).

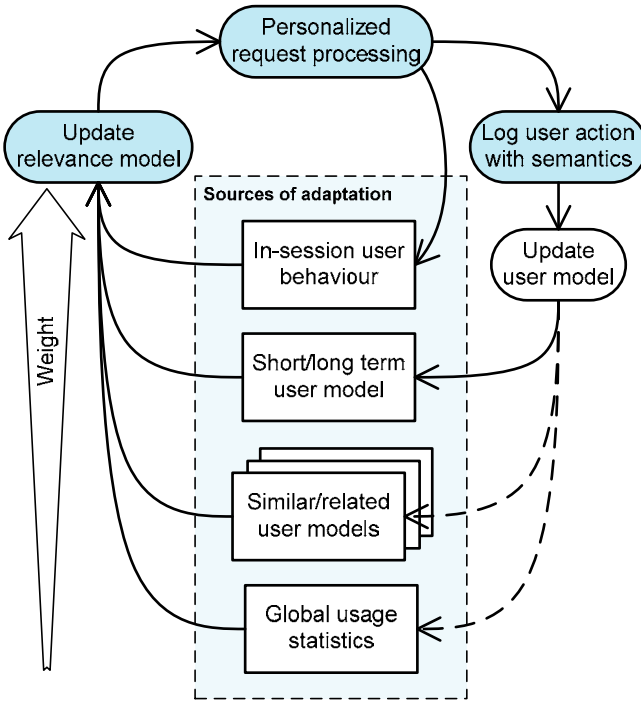


Figure 27. Overview of our user modelling-personalization loop (in blue) and the used sources of adaptation in descending order of weights (in-session behaviour, short- and long-term user preferences, and global usage statistics).

$$\text{dist}(U, V) = \sum_{X \in \text{facets}} (L_U(X) - L_V(X))^2 \quad (3)$$

We define  $T_U(X)$  as the temporary in-session relevance of facet  $X$  determined as the percentage of user clicks on facet  $X$  from the total number of clicks (4). Static relevance  $S_U(X)$  defines the relevance of facet  $X$  based on the user model and the respective

*confidence* in the relevance estimation (5). Dynamic relevance  $D_U(X)$  defines the total relevance of facet  $X$  based on the user model and current in-session user behaviour (6).

$$T_U(X) = \frac{\text{Clicks}(X)}{\text{TotalClicks}} \quad (4)$$

$$S_U(X) = L_U(X) * \text{conf}_U(X) + \left( \frac{C_U(X) + G_U(X)}{2} \right) * (1 - \text{conf}_U(X)) \quad (5)$$

$$D_U(X) = S_U(X) + T_U(X) \quad (6)$$

Figure 28 show an example of relevance model evaluation for Alice based on a total of three users. In the example, we employ the social weight  $W$  for cross relevance computation instead of user similarity based on local relevance. We see that in the final relevance evaluation  $S(\text{Alice})$ , the high local relevance for  $i:\text{hasTopic}$  and  $\text{rdf:type}=i:\text{Image}$  for Alice has been further reinforced via cross-relevance. Similarly, although Alice has no record in the user model for  $i:\text{hasWeather}$ , the high social weight towards Mary had increased its relevance above  $i:\text{hasAuthor}$ , which is in itself irrelevant to Alice.

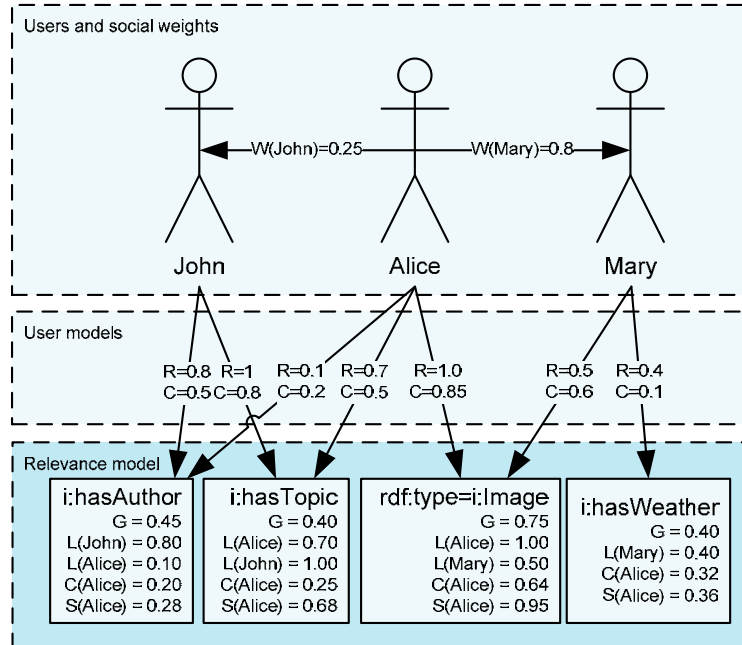


Figure 28. Example of relevance model evaluation for Alice based on social network weights  $W$  assigned to cross relevance between Alice, John and Mary (top). In user models  $R$  denotes relevance and  $C$  denotes confidence (centre); in the relevance model,  $G$  denotes global relevance based on local user,  $L$  denotes local relevance, while  $C$  denotes cross relevance based on social weights  $W$ . The final static relevance based on the user models  $S$  considers local, global and cross relevance (bottom).

### 5.3 Personalization method overview

Our facet personalization approach is based on the request processing scheme shown in Figure 24, p. 54 and works by executing steps in Algorithm 1:

#### Algorithm 1: Facet personalization

Input: *event*, *facets*, *relevanceModel*, *userModel*

Output: *facets*, *activeFacets*, *results*, *resultViews*, *eventLog*

1. **update** *query* **with** *event*
2. **foreach** *characteristic* **in** *userModel* **do**
3.     **update** *relevanceModel* **with** *characteristic*
4. **end foreach**
5. **if** *usableFacets* < *K* **then**
6.     **add** *generateDynamicFacets()* **to** *facets*
7. **end if**
8. **foreach** *facet* **in** *facets* **do**
9.     *updateFacetState(facet, query)*
10.    *updateFacetRelevance(facet, relevanceModel)*
11. **end foreach**
12. **sort** *facets* **by** *relevance*
13. *activeFacets* ← *facets*[1..K]
14. **foreach** *facet* **in** *activeFacets* **do**
15.     **sort** *facet.restrictions* **by** *label*
16.     *facet.annotation* ← *annotateFacet(facet, relevanceModel, userModel)*
17.     **foreach** *restriction* **in** *facet.restrictions* **do**
18.         *restriction.annotation* ← *annotateRestriction(restriction, relevanceModel, userModel)*
19.     **end foreach**
20. **end foreach**
21. *results* ← *retrieveResults(query)*
22. **foreach** *result* **in** *results* **do**
23.     *updateResultRelevance(result, relevanceModel)*
24.     *result.annotation* ← *annotateResult(result, relevanceModel, userModel)*
25.     **sort** *result.attributes* **by** *relevance*
26.     **add** *createView(result, result.attributes[1..L])* **to** *resultViews*



27. **end foreach**

28. **sort results by relevance**

29. **store event in eventLog**

Facet personalization is part of the request processing in the faceted browser and takes place once a user performs an event within the browser. First, the current query is updated based on the supplied user. Next the relevance model is updated based on the current user model, which may have been changed by user model inference agents due to previous user actions.

If the set of available facets is too small (i.e., smaller than a given  $K$ ), we try to generate new facets based on the domain ontology schema (see chapter 6). The current set of facets is then updated with respect to the current query and the relevance of each facet is recomputed based on the updated relevance model. We select the  $K$  most relevant facets as active facets by first ordering all facets based on their relevance and then selecting the top- $K$  facets. Note that the relevance of the last used facet is automatically boosted to keep it at the top of the list.

For each facet we sort facet restrictions based on their labels and add annotations either based on relevance (i.e., traffic light colours), history from the user model (i.e., background colour) or external annotation approaches such as the Panda tool in project NAZOU (Návrát & et.al, 2007).

The results of the current query are retrieved and ordered based on their relevance computed as an aggregate value from the relevance of their attributes. Similarly to facets, annotations are generated based on estimated relevance or via external tools. The most relevant attributes of search results are added to the final result view, which is then rendered to the user in the result overview.

Lastly, the event is stored in the event log for later processing by external user model acquisition agents (Barla, Tvarožek, & Bieliková, Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems, 2009).

## 5.4 Facet and restriction recommendation

Based on the computed relevance, facet adaptation processes facets and adapts them at run-time to the specific needs of individual users in these steps (see Figure 29):

- *Facet ordering* – all facets are ordered in descending order based on their relevance with the last used facet always being at the top.
- *Active facet selection* – the number of active facets is reduced to  $K$  (2 or 3) most relevant facets since many facets are potentially available. Inactive facets are used for queries but their contents are not updated, disabled facets are unused. Both inactive and disabled facets are still available on demand.

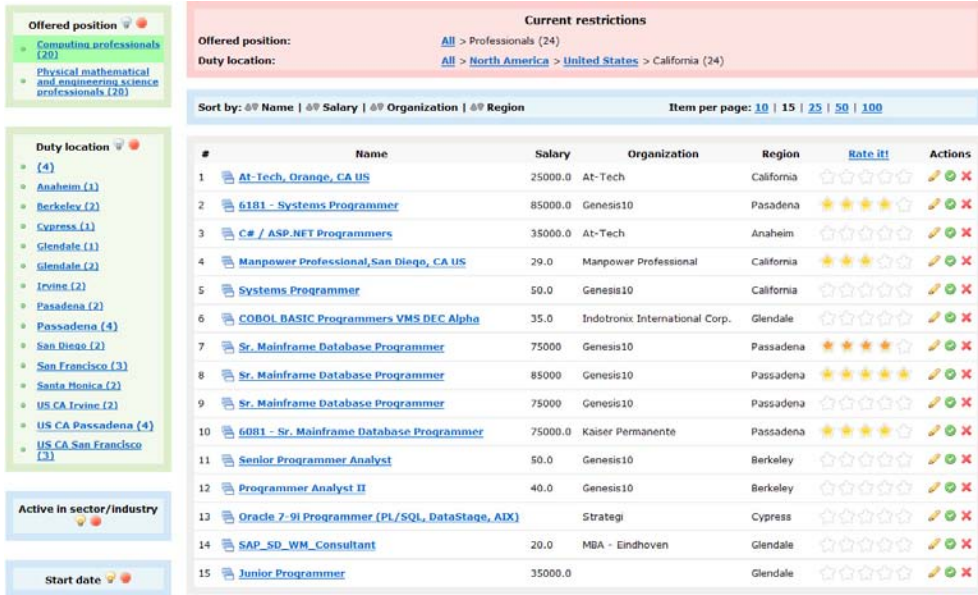


Figure 29. Example from our first prototype showing facet adaptation, annotation and restriction recommendation with active, and inactive facets (left), also showing a list view of search results with attributes and additional operations (right).

- *Facet and restriction annotation* – active facets are annotated with tooltips describing the facet, numbers of instances satisfying each restriction and the relative number of instances satisfying each restriction via font size/type.
- *Facet restriction recommendation* – the most relevant restrictions in a facet are marked as recommended (e.g., with background colour or “traffic lights”) effectively providing shortcuts to deeply nested restrictions.

Furthermore, the adaptation of search results adapts the displayed attributes of instances based on their relevance (i.e., what attributes are presented and their ordering). Using external evaluation tools, we also optionally add annotations to specific instances based on their “suitability” and reorder instances based on their personalized ranking specific for each user (e.g., acquired from explicit user feedback given by rating instances).

## 5.5 Search result recommendation

Based on the computed relevance and the results of external tools, we perform these recommendation steps (see Figure 29):

- 1) *Search result ordering* – we support simple results ordering – unordered results or ordered based on a single attribute (e.g., date). Additionally, we employ external ordering (relevance evaluation) tools, which either evaluate relevance based on

common global preferences, or on personalized ratings constructed from explicit user feedback (i.e., rating of instances) (Gurský, Horváth, Novotný, Vaneková, & Vojtáš, 2006). Furthermore, we employ external similarity evaluation tools, which enable users to search for instances similar to a given search result (Návrat & et.al, 2007).

- 2) *Search result annotation* – individual search result attributes are annotated similarly to facets and restrictions. Tooltips show their meanings (*rdfs:comment*) or their properties from the domain ontology. Alternatively, external annotation tools are used to provide custom (personalized) annotations generated from the domain and user ontologies (Návrat & et.al, 2007). For example, in the movie domain, we can display the suitability of a movie, based on its estimated relevance to the user's preferences, as background colour or via emoticons.
- 3) *View adaptation* – we support several adaptive views – simple overview, extended overview, thumbnail matrix or detailed view, which display increasingly more detailed information about individual search results (for details see chapter 7).

## 5.6 Discussion and evaluation

We used our first prototype to evaluate the usefulness of our facet personalization approach in the job offers domain (Project NAZOU) and partially in the scientific publications domain (Project MAPEKUS). Individual experiments were performed with the common evaluation framework that in addition to our faceted browser *Factic* also contained additional tools for user modelling (Barla, Tvarožek, & Bieliková, Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems, 2009) and relevance and similarity evaluation (Gurský, Horváth, Novotný, Vaneková, & Vojtáš, 2006).

Our aim was to perform a user study and to gather feedback on the design of our faceted browser prototype (Tvarožek & Bieliková, Personalized Faceted Navigation in the Semantic Web, 2007). The goal was to validate three aspects of our approach:

- *User action acquisition with semantics*; this includes the capability to capture and evaluate user actions, and the ability to derive meaningful user characteristics for successive personalization.
- *Efficiency and scalability with respect to large information spaces*; this includes the response time of the browser, the time required to find results of faceted queries and the time required to refresh facets for different complexity of queries.
- *Personalization based on the acquired user characteristics*; this includes the total task time to complete a given user scenario and the number of mouse clicks required to formulate the faceted queries and browse the search results.

## Data

We employed the ontological datasets from projects NAZOU and MAPEKUS for evaluation, whose detailed description can be found in Appendix C. The primary job offer dataset was used in different sizes with 101, 410 and 717 job offer instances and a total of about 700 classes. Of these, the first 101 instances were manually populated with the rest being automatically acquired via acquisition tools Ontea and Wrapper in project NAZOU (Návrát & et.al, 2007). While a larger dataset could have been used, the quality (i.e., the accuracy, correctness and completeness) of the automatically acquired instances was unsatisfactory for evaluation.

Our secondary dataset of publications from project MAPEKUS was automatically acquired from popular digital libraries (ACM DL, SpringerLink, DBLP). Since the entire dataset was too large for practical evaluation (hundreds of thousands of publications), we used a subset of the acquired ontology with roughly 10,000 selected publication instances from the IT field.

## Methodology

To evaluate user action acquisition, we employed the *SemanticLog* logging service, which supplied the recorded events to the user modelling back-end provided by the *LogAnalyzer* inference agent developed by Michal Barla (Barla, Tvarožek, & Bielíková, Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems, 2009). In this experiment, we manually predefined several different user profiles (e.g., senior programmer in San Francisco with high salary preferences) and asked users to behave based on these user profiles. We directly observed the estimated user model and compared it to the predefined user profiles and discovered a very high degree of consistency. The user modelling back-end was able to almost perfectly match the predefined user profiles thanks to the underlying semantics and mining rules. With increasing time spent in the system, the relevance and confidence in the inferred user model was also sufficiently high for personalization. We also indirectly evaluated user characteristics acquisition by evaluating the personalization engine assuming that if personalization was successful, user characteristics acquisition was also successful.

In order to evaluate the scalability, we experimented with the faceted browser using differently sized data sets in the job offers domain (101, 410 and 717 instances respectively) and publications (9996 and 770,000 instances). In each case, users were asked to execute a given user scenario and find a suitable set of job offer instances. Their goal was to find *job offers suitable for programmers in California with a start date in October 2005* (there was a total of 9 job offers satisfying these conditions). The publications domain was used primarily for scalability evaluation as it was automatically acquired and the overall completeness of the ontology was very limited for practical evaluation (i.e., it contained too few facets with actual data).

We conducted the experiment with 5 IT proficient users and measured the time required to perform individual actions in the browser (e.g., initialization with a data set,

facet selections, facet refresh times, result refresh times), and the total task times and number of mouse clicks required to find the desired set of search results. To evaluate the effects of personalization, we run experiments in three adaptation modes:

- *Without adaptation*; the baseline approach, where no enhancements were used and all 11 facets were active all the time. To emulate the effects limited facet evaluation, we also used this mode with fewer facets by initially selecting the first K facets and then the last used facet thereafter.
- *With adaptation*; the intermediate approach, where we adaptively selected K active facets that were visible, ordered facets based on their estimated relevance and hid less relevant facets.
- *With recommendation*; the fully personalized approach, where in addition to active facet selection, ordering and hiding we also selected and recommended the most relevant restrictions in facets.

To work around the cold start problem, we bootstrapped the user model with the corresponding user profiles in the first (non-adaptive) session, when the user modelling took place via *SemanticLog* and *LogAnalyzer* and then used this model in the second and third sessions with personalization. Since each user had the original user profile at hand during the experiment and the user modelling back-end had a high success rate, we only present result for one scenario (results for other scenarios were consistent). Although the order of individual experiments was fixed – non-adaptive, adaptive without recommendation, adaptive with recommendation, due to user modelling constraints, the variation between users was small. Since the primary limiting factor has been the response time (i.e., the time required to refresh the user interface), we surmise that the actual order of the experiments had little effect on the outcome.

## Results and lessons learned

Based on our observation of semantic logging and user characteristics evaluation, the *user modelling back-end worked as expected and discovered relevant user characteristics matching the initial user profiles* based on which users behaved. Still, we noticed a significant delay when logging was enabled via the *SemanticLog* web service. Depending on the current view state, event logging took longer than the actual processing of the faceted browser and output generation due to web service communication overhead and network latencies. This forced us to modify our logging service to a direct logging approach of the view states into the database effectively omitting the web service for the most part. This design modification was implemented before the actual personalization experiments took place.

Our scalability evaluation discovered serious issues with the scalability of the Sesame ontological repository which in turn resulted in poor performance of our browser. Figure 30 shows the achieved response for different user actions and dataset sizes (shown in brackets – job offers 101, 410, 717 instances, 9996 publications) without adaptation. Result refresh times were usually in the 10-100 ms range for simple or no queries and in

the 100-1000 ms range for faceted queries with multiple restrictions. Facet refresh times were much longer due to the increasing number of restrictions that had to be computed in the 1-100 second range which made the browser effectively unusable.

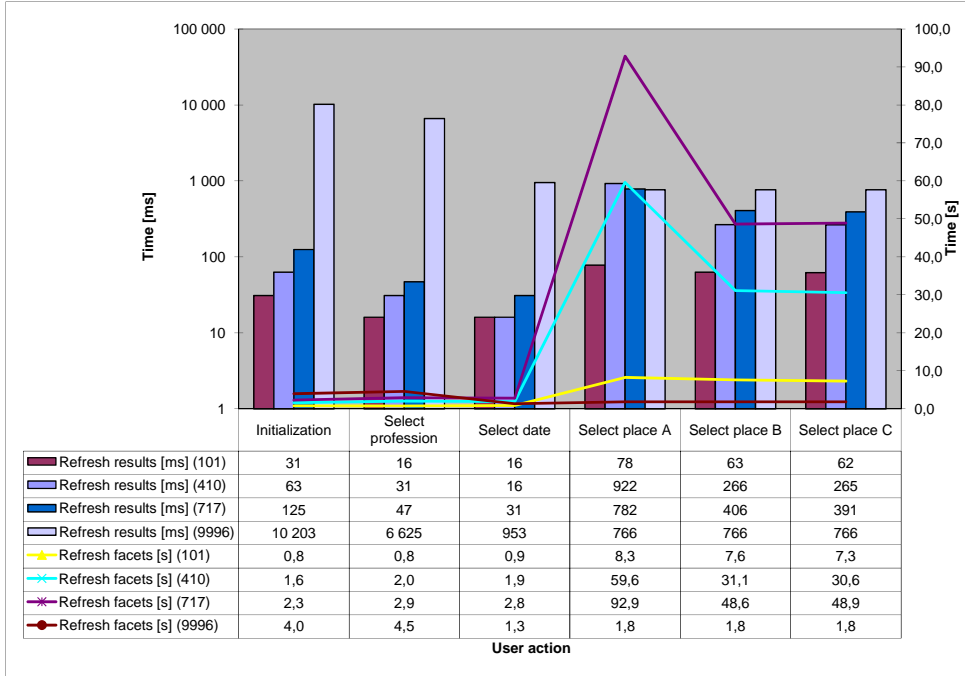


Figure 30. Experimental results showing the dependency of refresh times on performed actions (in columns) and dataset size (shown in brackets). Facet refresh times correspond to the recalculation of facet contents while result refresh times correspond to the time required to calculate the set of result URIs.

This was mostly due to the high complexity of hierarchical faceted queries (e.g., location) which result in a high branching factor and thus high time complexity in addition to the growing size of the data set. Already with a 717 instance dataset, facet refresh times reached 92.9 seconds, which only dropped with additional facet refinements that reduced the branching factor although 49 seconds is still too much for being practical. Consequently, our observations confirmed the theoretical linear dependency between restriction time and facet refresh times.

We next explored the facet refresh times with respect to the number of active facets. We defined 11 facets based on the domain ontology of job offers although the ontology would allow us to define many more with an even higher branching factor. Since it is not possible to pre-compute all possible restriction combinations, they must be computed at runtime. For  $N$  facets and an average facet branching factor  $K$ , the theoretical complexity

for restriction updates in facets is  $O(N * K)$ , with the total number of possible combinations being  $K^N$ .

Our evaluation showed that adaptive selection of active facets (i.e., fully rendered) can significantly reduce information overload (i.e. the number of facets a user must examine) and thus total processing time which depends roughly linearly on the number of displayed facets (see Figure 31). While without adaptation 9 clicks and about 300 seconds were required, with adaptation the number of clicks increased to 10-11 since the right facets were not always active and thus had to be manually enabled. Still, this resulted in shorter refresh times and thus shorter total task times around 63-296 seconds.

Recommendation of suitable restrictions based on the user model further improved total task time to 36-61 seconds and also decreased the number of necessary clicks to 5-6 due to the effective creation of navigational shortcuts that allowed users to skip several clicks by directly selecting suitable restrictions within a restriction hierarchy. As before, the number of clicks increased as the number of active facets decreased as more facets had to be manually activated.

Based on our experiments we discovered that  $K$ , the ideal number of active facets seems to be between 1-3 with adaptation or recommendation. Without recommendation, the ideal number of facets seems to be 1 so that the user can select the facet manually and save on refresh times, which however defeats the purpose of having facets in the first place. Still, the ideal number of active facets would also depend on the domain.

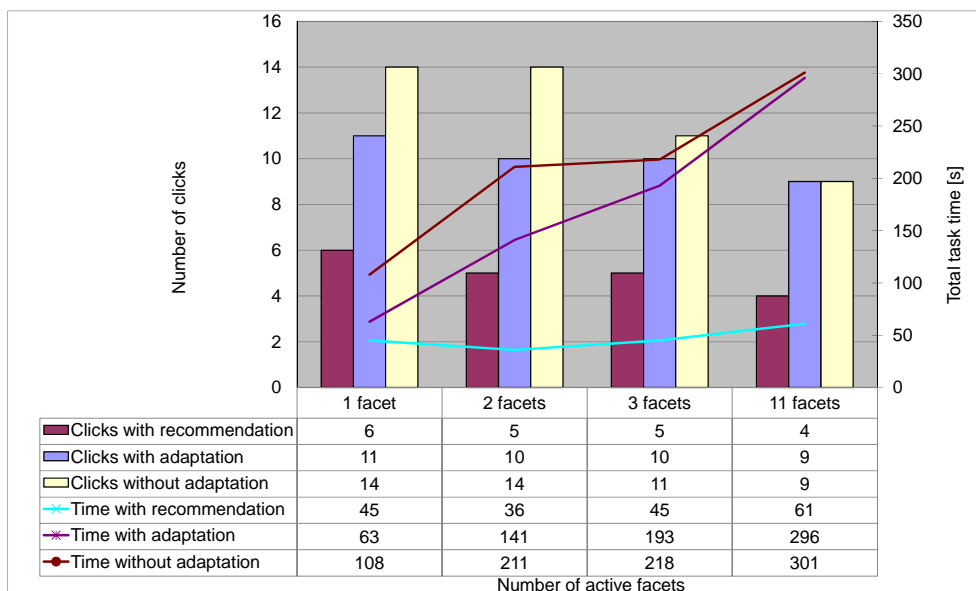


Figure 31. Experimental results of personalization for different numbers of simultaneously active facets in different adaptation modes (non-adaptive, with adaptation, with recommendation).

Despite the very positive feedback and highly promising results, we encountered issues in the form of several performance and scalability bottlenecks with remote repositories due to repository querying limitations and network delays, which we addressed in our second prototype:

- The cost of ontological queries is high and consequently, the processing of ontological queries is slow. We were unable to resolve this problem although we improved overall performance by caching data in *Factic*. Furthermore, the ontological repository Sesame is rather immature – it is slow, unoptimized and contains several bugs, which prevent correct evaluation of queries.
- SeRQL – the recommended query language for *Sesame* and thus *Sesame* lack several important features such as COUNT() or ORDER BY. These must thus be emulated by our application which further reduces performance.

While our approach proved to be particularly suited for the job offers domain – a very complex information spaces with several deep hierarchical classifications (e.g., regions or positions) and intricate concept relations, it was also useful for less complex domains – publications and images. Still, specifically for large data sets in the publication domain in the 700k instance range, the browser became unusable due to long response times in the minutes range. We tried to address this by caching some results, but the performance of Sesame was unsatisfactory and also bottlenecked in its inability to use enough memory for a memory-based repository where a MySQL backed repository had to be used instead.



## 6 Adaptive View Generation

---

Our focus lies with the generation of exploratory search interfaces for the Semantic Web environment, although this can be somewhat generalized towards the Deep Web and even legacy Web environment. In order to support exploratory search and achieve these goals, we need to support three parts of user experience:

- *Query construction*, which includes the initial construction of an exploratory search query, its modification and execution; to support multi-paradigm search and exploration based on our previous work, we need to support keyword-based, view-based (faceted) and content-based (query-by-example) query construction.
- *Result browsing*, which includes the rendering of suitable result overviews, selection of result ordering and the displayed result attributes, and support for effective selection of individual results for further exploration.
- *Resource exploration*, which includes the detailed presentation of individual resources, their attributes and relationships with other existing resources.

We address these issues by generating a set of user interfaces each supporting the individual stages of the exploratory search process. We generate:

- *Faceted browser interfaces* for advanced query construction and modification.
- *Result overviews* for effective presentation of selected result attributes.
- *Graph-based exploration views* for incremental horizontal exploration of semantic resources and their relations with other resources.

We focus on querying interface generation and thus primarily describe faceted browser interface generation. The use and generation of interfaces for result browsing and exploration – result overviews, editing views and graph views is covered in chapter 7.

### 6.1 Facet Generation

During facet generation, we examine metadata describing the information space, identify patterns corresponding to facets, construct facet restrictions based on the identified metadata and map the resulting facet onto the graphical user interface and the semantic back-end, which provides querying services. As such, facet generation must define these facet properties:

- A *facet template*, which corresponds to a pattern found in domain metadata and specifies the overall type and behaviour of the facet.
- A *restriction template*, which defines how the individual restrictions in the facet are constructed and mapped onto the domain ontology.

- A *query template*, which defines how the back-end query engine creates database queries and maps them onto facet restrictions.
- A *visualization and interaction template* (i.e., the corresponding widget type), which binds the facet to the graphical user interface and handles user input.

The purpose of the facet generation process is to identify specific predefined patterns in the metadata and map them onto a set of predefined templates in three successive steps: facet identification, construction and mapping as described below.

### 6.1.1 Facet identification

During the facet identification stage, we identify the facet template, restriction template and query template as described in Algorithm 2:

#### Algorithm 2: Facet identification

Input: *domainOntology*, *facetPatterns*, *query*

Output: *facetCandidates*

```

1.  candidateProperties empty

2.  foreach patterns in facetPatterns do
3.    add findCandidates(domainOntology, pattern) to candidateProperties
4.  end foreach

5.  foreach property in candidateProperties do
6.    facetCandidate empty

7.    if property is literal then
8.      facetCandidate.facetTemplate literalFacet
9.    else
10.     facetCandidate.facetTemplate objectFacet
11.    end if

12.   if property is hasHierarchicalValues(property) then
13.     facetCandidate.restrictionTemplate hierarchicalFacet
14.   else
15.     facetCandidate.restrictionTemplate enumerationFacet
16.   end if

17.   facetCandidate.queryTemplate findQueryTemplate(property, query)

```

```

18.   add facetCandidate to facetCandidates
19.   end foreach

```

We first search for eligible candidate properties by examining properties of individual instance types and their transitively associated properties by trying to match predefined facet patterns onto the ontology schema. Next, we identify specific facet types based on low-level metadata facet templates.

We distinguish:

- *object facet templates* that correspond to properties having complex object values (e.g., a class such as *di:Author*), and
- *literal facet templates* that correspond to properties having simple values (e.g., numbers, dates or strings).

In the example in Figure 32, *di:createdBy* is a suitable candidate property matching the *class-property-class* pattern between *di:Photo* and *di:Author* resulting in a direct object facet template; *di:viewedCount* is a suitable candidate property matching the *class-property-literal* pattern between *di:Photo* and *xsd:int* resulting in a direct literal facet.

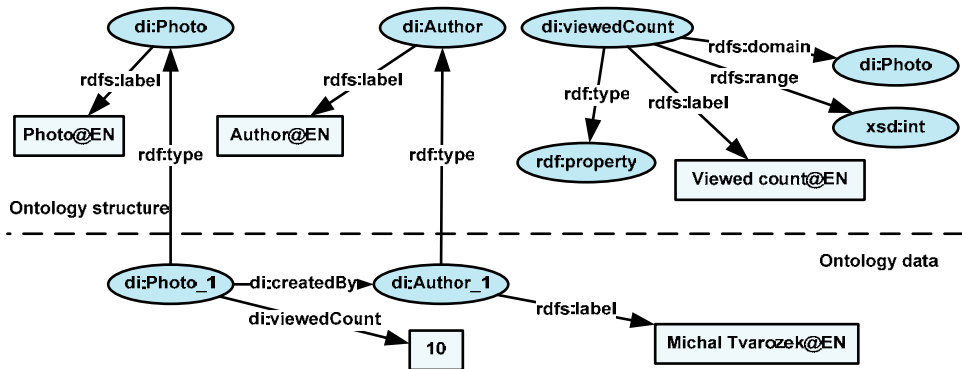


Figure 32. Example of a domain ontology of photos showing suitable candidate properties, e.g. *di:createdBy* or *di:viewedCount* for the *class-property-class* and *class-property-literal* patterns respectively.

Similarly, we define two *restriction templates*:

- *enumeration*, which corresponds to a flat list (e.g., days of the week), and
- *hierarchical taxonomy*, which corresponds to a hierarchical tree of values connected via a transitive property in the domain model (e.g., a hierarchy of geographical locations such as country-state-city-street).

We distinguish these *query templates* based on the instance-property relation:

- *Direct query template*, which corresponds to the direct property pattern:  $\{instance\} \text{ property } \{value\}$
- *Indirect query template*, which corresponds to the indirect property pattern:  $\{instance\} \text{ property}_1 \{ \} \dots \{ \} \text{ property}_N \{value\}$

The identification of query templates depends on the instance types for which a facet is generated, i.e. for one type a facet might be a direct facet while for another it might be an indirect facet. For example, the facet for the author name (corresponding to the *rdfs:label* property of the *Author* class) is a direct facet for the type *di:Author* whereas it is an indirect facet for the type *di:Photo*. Thus our facet identification algorithm tries to match these predefined templates onto the domain ontology metadata with respect to specific instance types (e.g., specified in the current query), evaluates possible matches and forwards matches to the facet construction stage.

### 6.1.2 Facet construction

Since in practice it is not desirable to generate all possible facets due to their large number, efficient attribute selection is crucial in order to select the most suitable attributes based on their relevance for specific users. Consequently, the purpose of the facet construction stage is to select suitable facets for use and construct their descriptions for use within our faceted browser as shown in Algorithm 3:

#### Algorithm 3: Facet construction

Input: *facetCandidates*, *domainOntology*, *query*, *relevanceModel*

Output: *facets*

1. *facets* **empty**
2. **foreach** *facet* **in** *facetCandidates* **do**
3.   **if** *computeUsefulness* (*facet*, *domainOntology*, *query*) < *S* **then continue**
4.   **if** *computeRelevance* (*facet*, *relevanceModel*) < *T* **then continue**
5.   *facet* *initializeFacet* (*facet*, *domainOntology*)
6.   *facet.type* *selectType* (*facet*, *domainOntology*)
7.   *facet.restrictions* *initializeRestrictions* (*facet*, *domainOntology*)
8.   **add** *facet* **to** *facets*
9. **end foreach**

We first determine the usefulness of a facet candidate and discard useless facets (e.g., facets that would have no restrictions, facets without corresponding results). Similarly, we discard facets whose estimated relevance to the user would be too low.

As dynamically generated facets are created from either direct or indirect attributes of instances, we propose different facet types (Figure 33):

- *Simple facets* – top-level facets based on direct or indirect attributes of target instances, e.g. directly for images – the object, keywords or location, or indirectly – the resolution of the camera used to take the photo.
- *Nested facets* – facets that in addition to (or instead of) a set of individual restrictions contain a set of *child* facets, e.g. a facet that contains facets for the impact, topics and location of a conference associated with a paper.

Direct attributes of target instances are always presented by means of direct facets. If only one indirect attribute of an associated instance type is presented an indirect facet is used. If multiple indirect attributes of the same type are presented, a nested facet can be used so that each nesting level corresponds to one level of attribute indirection.

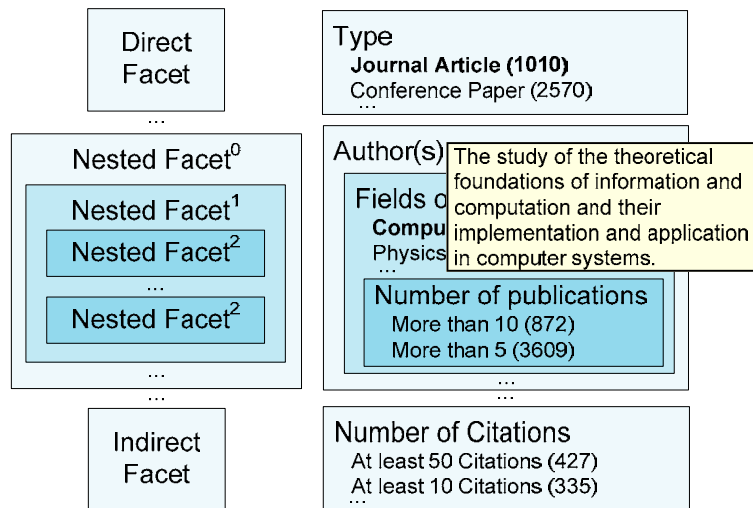


Figure 33. Facet types (left) and adaptation examples (right). Bold text is used for recommendation, tooltips and instance counts for annotation.

Once a facet candidate is deemed suitable, its internal representation must be constructed before it can be used in the browser. In the facet construction stage, we apply the templates identified in the previous stage, construct facet restrictions based on the restriction template, and persistently store facet metadata for future use.

The crucial step of facet construction is the initialization of facet restrictions using the restriction template and the definition of the interaction mode. For enumeration facets, the restriction list is constructed from the instances of the corresponding ontology class;

for a hierarchical facet, the restriction tree is constructed based on the transitive property connecting the values of the restrictions in the domain ontology.

Normally a facet can work as a list of restrictions from which users can select one or more values, or as a search box where users can search for and select a specific restriction. We determine the interaction mode based on the overall number of potential restrictions; *list mode* is used for a small number of predefined values (e.g., days of the week), *search mode* is used for large numbers of values (e.g., all cities on Earth). If an ordering of values is defined in the ontology, we can also create restriction intervals to cover continuous values (e.g., real numbers or dates).

### 6.1.3 Facet mapping

The last facet mapping stage selects a suitable user interface widget to render the generated facet in the faceted browser, and maps the constructed facet and restriction values onto the widget. The widget provides facet visualization and handles user interaction forwarding events and facet metadata to the back-end search services, which use the *query template* and the user selection in the facet to construct SPARQL queries in order to retrieve results corresponding to the generated facet.

Although a broad range of potential interface widgets could be developed, such as lists, histograms, maps, timelines, they were beyond our focus as automated discovery of what specific visualization/interaction to use would likely prove difficult. Thus we only employ list widgets at this time and leave the use of more advanced widget types as one possible direction of future work.

## 6.2 Discussion and evaluation

We used our second prototype in the digital image domain with an image dataset containing about 8,000 manually and semi-automatically annotated images to perform a proof of concept experiment with our facet generation approach, and to perform a user study with our graph exploration approach. The second prototype was realized as a client-side Silverlight application working inside a web browser, which allowed us to move user specific functionality onto the client and also provide interactive features not supported by HTML (e.g., the interactive graph view). We performed several experiments to validate individual parts of our approach in the digital image domain.

Our goal was to validate our facet and result overview generation approach via a proof of concept experiment with facet/overview generation approach. The goal is to verify that our approach generates meaningful and usable facets for our personalized faceted browser. Note that the goal is not to generate the best possible set of facets, but rather a good enough set to use for personalization.

## Data

Our domain ontology of images is based on the popular Kanzaki EXIF ontology (<http://www.kanzaki.com/ns/exif>) and contains about 8,000 manually and semi-automatically annotated images. The entire ontology consists of 35 classes, 50 properties (including relations and attributes), more than 32,000 individuals and in excess of 150 000 facts. For individual photos, the ontology describes EXIF metadata as supplied by the camera, information about formats in which the photos are available (e.g., resolution, aspect ratios), and optional additional annotations such as the author, the object and background of the photo, the place, overall theme and expression, lighting conditions, weather and the event to which the photo belongs.

## Methodology

In the proof of concept experiment, we generated facets from the available data and examined how the original browser behaved in practice and whether the interface was still usable for its intended purpose in terms of usability and performance. We performed several experiments with and without personalization, and also after some changes in the information space have been made.

## Results and lessons learned

The experiments with facet generation proved the approach was viable for interface generation with minimal performance impact. We successfully managed to distinguish facet and restriction templates, direct query templates, and construct and map facets to interface widgets and use them in our exploration interface without any significant negative impact over manually created facets due to facet generation. Note that it is not possible to quantitatively evaluate the “quality” of the generated set of facets, because there is no “best” set of facets. Based on our experiments, we point out these lessons learned:

- Identification of direct query templates resulted in many facets being generated, which we expected to handle at the personalization stage later in the browser. However, this had negative impact on performance and we had to employ selection metrics (e.g., based on significance) already during the facet identification stage, similarly to (Oren, Delbru, & Decker, 2006).
- The identification of indirect query templates was limited due to the complexity of selecting viable options. Consequently, either the identification algorithm must be further refined or a workaround via indirect nested facets (i.e., facets in facets) needs to be used complicating facet generation and mapping.
- During facet generation and result overview generation, blank nodes and helper objects in the domain ontology caused problems as, e.g., empty, meaningless or unnamed interface items were generated and had to be accounted for.

- Some generated facets such as location eventually had too many restrictions (e.g., hundreds) making them unusable and significantly decreasing performance. This required the change of the interaction mode from *list mode* to *search mode*, where users could type in their desired restriction instead of selecting from a list of hundreds of items. This problem could also be alleviated by prior hierarchical structuring of the information space before facet generation.
- The users preferred *alphabetical restriction ordering* in facets; other orderings such as relevance based or potency based had negative impact on user experience as users were unable to seek in the restrictions which were in an unexpected order.
- The preferred ordering of facets was based on their *relevance towards the current task* and secondarily, once the primary facets (principal to the task) were already exhausted, based on their respective specificity (i.e., capability to further restrict the information space to a smaller set of results).
- Using type/information specific facet widgets instead of list widgets would likely improve usability in specific cases, such as date selection via calendars, location selection via maps or timeline selection via histograms as was done in (Dörk, Carpendale, Collins, & Williamson, 2008), but effectively generating mappings for advanced widgets would be more complex.



## 7 Multi-Paradigm Exploration

Multi-paradigm exploration constitutes the cornerstone of our approach by integrating a set of search, navigation and visualization approaches into a comprehensive exploratory search solution. We improve the opening-midgame-endgame scheme, originally proposed in Flamenco, by adding user support for the individual stages and populating them with additional complementary approaches to facilitate end-user grade exploration experience (see Figure 34, which follows Figure 23 with added transitions). Our main focus in this part of our approach is the *integration of several specialized approaches* into a single coherent solution.

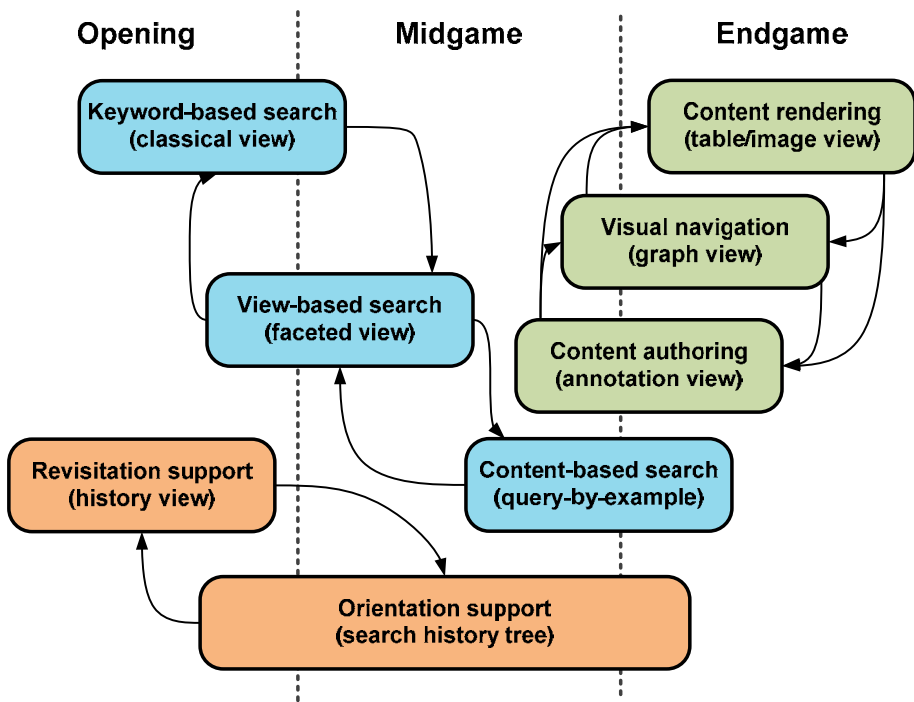


Figure 34. Overview of our multi-paradigm exploration approach showing the scope and applicability of individual sub-approaches to specific stages of the exploration process. Search approaches span primarily the opening and midgame (blue), content viewing, annotation and browsing approaches focus on the endgame (green), while orientation support approaches span all stages (orange).

Figure 34 also shows the overlap between individual approaches and the three stages of the exploration process, which indicates the usefulness and applicability of a given approach with respect to views employed in a specific exploration stage. Transitions between all approaches within a group are normally possible, e.g. between the history view and search

history tree, or table/photo view, graph view and annotation view. Transitions between view groups are normally performed via the faceted view, i.e. are initiated from the faceted view or return the user back to the faceted view (although technically, other transitions would be possible too).

## 7.1 Searching and browsing

### 7.1.1 Classical view

Figure 35 shows the *classical view* which is based on the initial screens of existing web search engines such as Google or Bing. In addition to the search box, we employ a tag cloud-based multi-purpose view that can correspond to:

- *Information artefact types* thus giving users an overview of what kinds of information can be explored (e.g., photos, events, regions).
- *Tags (topics) of recently added or modified information artefacts* thus giving users an overview of what new information is available
- *Popular information artefacts* effectively providing social (either global or community based) recommendation and providing users with an overview of current trends.



Figure 35. Example of the class view from our second browser prototype, which is based on existing initial search engine screens with a logo, search box and the additional tag cloud corresponding to types of information artefacts (bottom).

We normally only display one kind of a tag cloud although several tag clouds corresponding to different information (e.g., recently visited resources, popular resources) could be shown at once on larger screens also combined with the *history view* effectively providing users with a homepage like experience.

### 7.1.2 Faceted view

Since prior work by Kules et al. has shown that users mostly use the facets and the result overview when working with faceted browsers, we focused mainly on their improvement (Kules, Capra, Banta, & Sierra, 2009). Our faceted view integrates these approaches (see Figure 36):

- *Faceted browsing* based on the traditional layout with facets on the left, query at the top and results in the centre.
- *Adaptive search result overviews* (list view, matrix view) providing users with quick and easy understanding of the current result set.
- *Search history tree* based on interactive graph visualization for orientation and history support.
- *Query-by-example* via search result rating or similarity search (performed via external tools) which improves querying capability of users and supports the exploration of similar information artefacts.
- *Optional keyword-based full text search* as a complementary approach to faceted- and content-based search. Note that in the semantic web environment, there normally is no full text to search so keyword based search is limited to the labels and comments of resources.

### Facet visualization

Our approach works with the notion of *facet widgets* (i.e., user interface controls), which correspond to back-end (faceted) querying services. We distinguish two primary widget categories:

- *Object facet widgets* correspond to associations of search results with specific information artefacts (e.g., a photo associated with an author).
- *Literal facet widgets* correspond to associations of search results with literal values (e.g., number or dates).

Consequently, objects widgets usually correspond to text-based lists of existing resource labels, while literal widgets correspond to more abstract intervals or sets of possible values. Both of these can be hierarchically organized either in a flat enumeration (e.g., days of the week) or a deep taxonomy (e.g., ACM subject headings).



Figure 36. Example of our tree-based history visualization showing an initial keyword query (top left) and the successive faceted query refinements (left). The rest of the interface shows the list of available facets (centre) and the list of search results (right).

Object facets typically correspond to different domain properties which can have either few or many values. We thus devised two primary views for object facets:

- *Hierarchical enumeration facet*, which shows an exhaustive list of available values at a given level with optional recommendations to specific restrictions.
- *Hierarchical search facet*, which shows the most relevant examples of restrictions and provides a search (auto-complete) box where users can enter a specific value if they know what they are looking for.

While there are several possible orderings of restrictions within facets (alphabetical, count-based, relevance-based), the users typically preferred alphabetical ordering of items which allowed them to search for known values. Consequently, we primarily employed alphabetical ordering of items in enumeration facets and relevance-/count- based ordering in search facets.

Although type based visualization of object facets could be useful, e.g. via a map for locations such as in VisGets (Dörk, Carpendale, Collins, & Williamson, 2008), we focused on the generic use of text-based visualization of facets.

Literal facets, such as dates or numbers can be seen as hierarchical intervals of values (although not all literals necessarily have an ordering). Thus we employ *literal enumeration facets* for discrete or nominal values and *literal interval facets* for ordinal values.

Additionally, we employ personalization to the ordering, hiding and selection of facets and restrictions based on estimated user relevance (see chapter 5 for details of personalization).

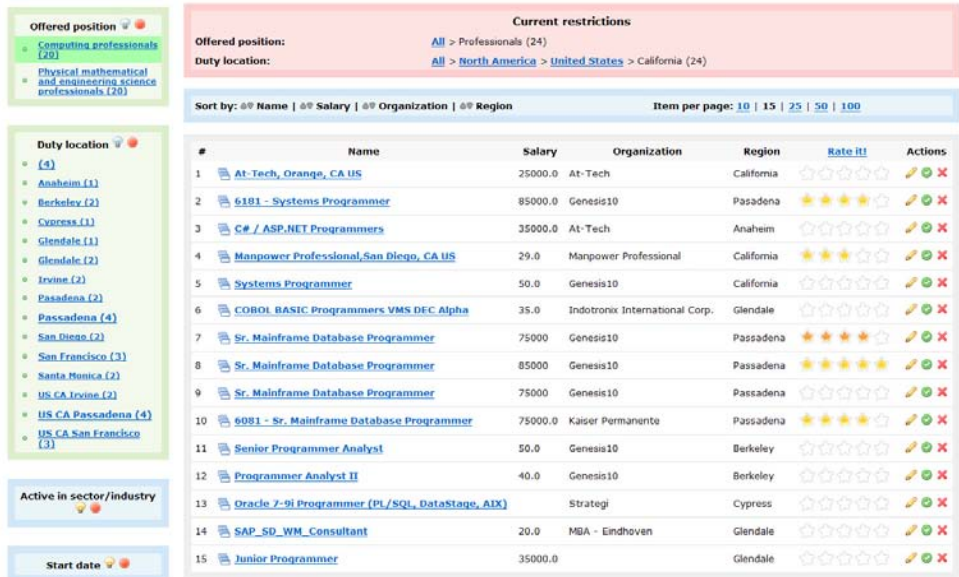


Figure 37. Example of our first browser prototype showing facet personalization (left), list view search results (centre) with query-by-example via rating and similarity search (right). The current query is shown at the top along with result sorting options.

## Query-by-example

Our query-by-example approach takes advantage of both implicit and explicit user feedback. In the implicit feedback scenario, we evaluate similar information artefacts to a user specified positive example via external concept comparison approaches (Návrát & et.al, 2007) and present the results via our faceted browser, which enables users to perform additional faceted exploration of the result set (see Figure 37).

In the explicit feedback scenario, we allow users to rate individual search results on a five level Likert scale thus allowing users to select both positive and negative examples. After a sufficient number of user ratings has been acquired, users can toggle the evaluation of the ratings into a user preference model and the corresponding search for the best matching search results, which is performed via external tools working with fuzzy logic and multi-criteria top-k search (Gurský, Horváth, Novotný, Vaneková, & Vojtáš, 2006).

## Adaptive result overviews

The results of user queries are first shown in result overviews in the faceted browser before individual results can be explored in the endgame. We support several result overview types which display increasingly more detailed information about search results. The attributes of the displayed instances are adaptively chosen and ordered based on their estimated relevance derived from the user model (see chapter 5.2). Moreover, since the faceted browser can show instances of different types, users can seamlessly switch from

browsing/searching for e.g., images to videos, then to actors and back to images. We devised result overviews based on two levels of abstraction:

- The *list view* provides detailed information about search results by showing either all or a personalized subset of search result attributes (see Figure 38).
- The *matrix view* provides a general overview of many results showing only their labels and associated thumbnails (if applicable) with additional information being provided in tooltips (see Figure 36, p. 86); additionally, it allows users to access the annotation view for (batch) editing of information.



Figure 38. Example of the faceted view from our second browser prototype showing the facets (left), current query (top) and the list view results overview showing all result properties (centre). The Author facet corresponds to a direct object facet, while the Aspect ratio and Camera facets are indirect object facets associated via an EXIF helper object with the original photo.

The list view is generated dynamically for each information artefact type and displays its existing attributes, also accounting for multiple property values and both object-type (shows labels) and data-type properties (shows values). In Figure 38, *ListView* shows properties of a specific result directly derived from the domain ontology visualized as *label-value* pairs. While normally showing all result properties to maximize information, once enough information about user preferences is present, the list view can be personalized to select only the most relevant properties. This is done based on the estimated relevance of individual result attributes, where the most relevance attributes are shown either up to a given threshold, when scrollbars are used, or up to the given screen size (see section 5 for details of personalization and relevance evaluation):

- Customize the order of the presented attributes based on their relevance towards the estimated user task and/or goal.
- Hide irrelevant or scarcely used attributes based on their relevance and global usage statistics.

The matrix view provides a quick, high-level overview of the current search result set. It also provides additional editing functions for the presented content (after logging in) via the annotation view, which can be triggered after selecting one or more search results for (batch) editing of their properties.

Both views also support the transition to content exploration views – the table view, photo view and graph view, which are used during the endgame to explore individual search result properties.

## 7.2 Result exploration

The endgame of the exploration process consists of individual result exploration, where users need see and understand the properties (i.e., actual content) of the information artefacts. We provide three types of exploration views:

- Textual attribute exploration via the nested *table view* for visualization of information properties and the *annotation view* for their modification.
- Relation exploration via the *graph view* for interactive exploration of relations between information artefacts.
- Content viewing via the *image view*, which renders associated content (i.e., photos) in a native user friendly way.

### 7.2.1 Textual result exploration

The *table view* recursively renders the properties of an individual search result in a nested table thus providing users with an exhaustive visualization of the details associated with a given resource (see Figure 39). Compared to some other existing approaches, which only provide direct properties, the nested visualization improves user orientation by maintaining the context of resources, as normally one would have to click a link to view properties of other resources thus losing the original context.

One downside to this approach would be information overload, which again can be addressed by personalization (see further chapter 5) and by allowing the user to selectively expand the table entries (or graph nodes in section 7.2.2).

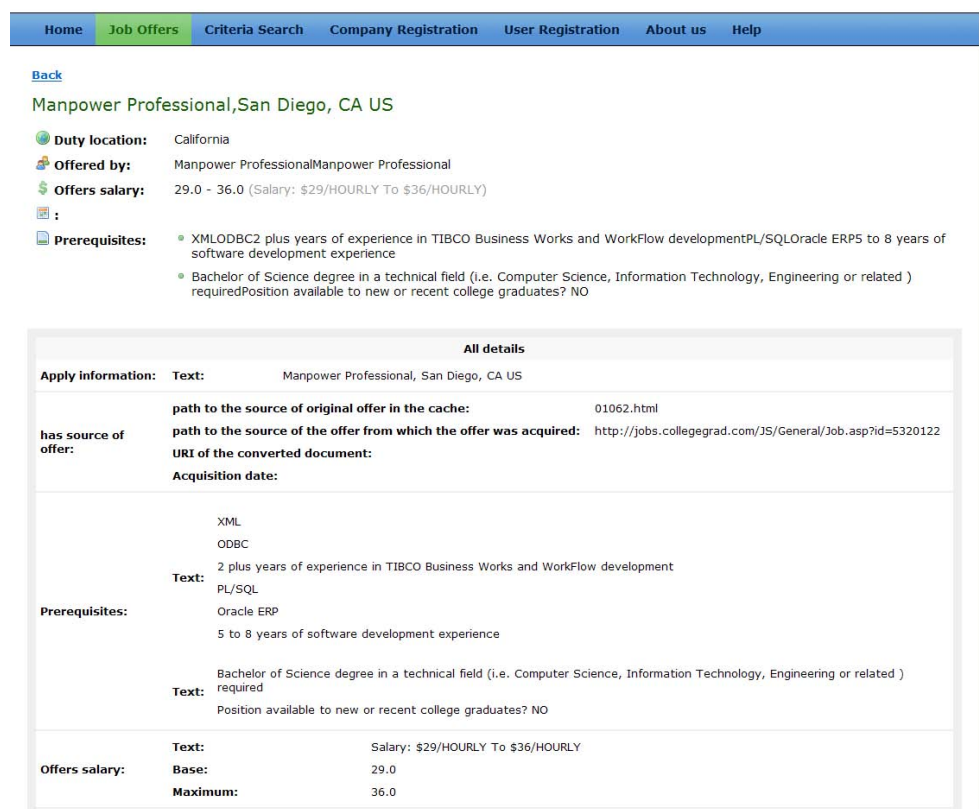


Figure 39. Example of our first browser prototype showing the table view of a selected job offer. The details in the bottom part show the nested table approach recursively rendering the properties of associated resources.

The *annotation view* supports collaborative content creation by allowing authorized users to create new information artefacts, modify or optionally delete existing ones via a generated form-based interface (see Figure 40, left). Users can either enter entirely new values or select pre-existing values from drop-down menus. Moreover, users can remove property values or entire resources, create new resources and even alter the schema of the repository (in ontologies schema and data are treated equally).

Similarly to result overviews, we generate the annotation view (accessible from result overviews) separately for each specific resource type. We identify all applicable properties from the domain ontology metadata, construct editing widgets based on property types (e.g., text boxes with language selection or auto-complete combo boxes, with single/multi-value support). Properties with existing values are shown first, while properties without values are shown at the bottom (see Figure 40, left).



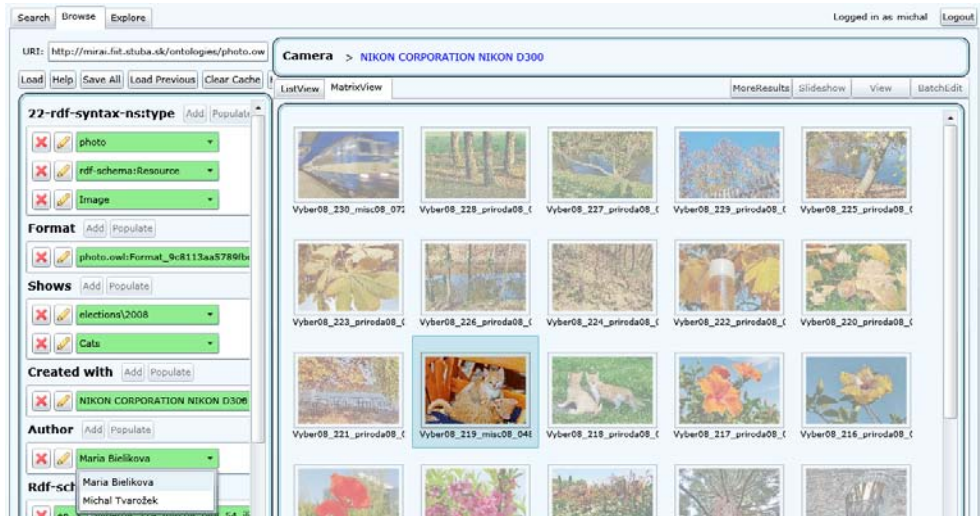


Figure 40. Example of a generated matrix result overview showing image thumbnails (right), and the correspondingly generated annotation pane for collaborative content creation (left).

## 7.2.2 Content-sensitive result exploration

We provide domain specific visualizations based on resource type to support “natural” access to information artefacts. As we also worked with an image collection, we devised a specialized *image view*, which enables users to view the photos similarly to popular web-based photo galleries (see Figure 41). Image view supports image manipulation features such as zoom, rotate or slideshows, it shows image thumbnails and can also display basic image attributes.

## 7.2.3 Visual relation exploration

We provide visual result exploration support via a graph-based visualization of resource properties. The graph exploration view consists of the graph visualization window, predicate filtering windows and an options toolbar (see Figure 42). Users can access the view either directly by typing in the URI of the node they wish to explore, or by exploring a result found in faceted view.

The graph view is generated directly from a domain ontology showing individual resources and their relations, also taking advantage of relevance evaluation from the personalization engine. Relations are intentionally visualized as separate nodes connecting resources to reduce information overload when one relation can have multiple values and to improve graph layout. In Figure 42, the relation *weather* shown in the right part of the graph would otherwise have to be displayed on all edges making the graph less readable.



Figure 41. The photo view shows images via an interactive interface similar to popular web-based or desktop photo galleries with support for slideshows, zooming, panning, image rotation, thumbnails strip and additional customizations.

An exploration session starts when a user selects the first dark node (information artefact), e.g. via facets or its URI. This shows selected resource (central node) and its properties (i.e., relations to other resources), which corresponds to a window or a view of the graph. Our graph view supports incremental horizontal exploration of resources, as users can move the view's focus to different nodes or further expand nodes to show their properties. I.e., users can next move the visible window by selecting another central node, or incrementally expand the view by expanding one or more of the visible nodes. The view in Figure 42 was initiated by showing the node *Trees* (left) and expanding the node *Sunny*.

We visualize both resources and properties as nodes to reduce information overload and to improve graph layout as a single property can connect multiple resources at the same time. Dark nodes correspond to individual resources, white nodes correspond to relations between them; arrows denote relation directions, node attributes (i.e., values of literal properties) are normally hidden and only shown as tooltips after hovering over a node. To make the graph understandable, we employ a force-based layout algorithm, but also allow the user to fix and manually reposition nodes in the resulting graph.

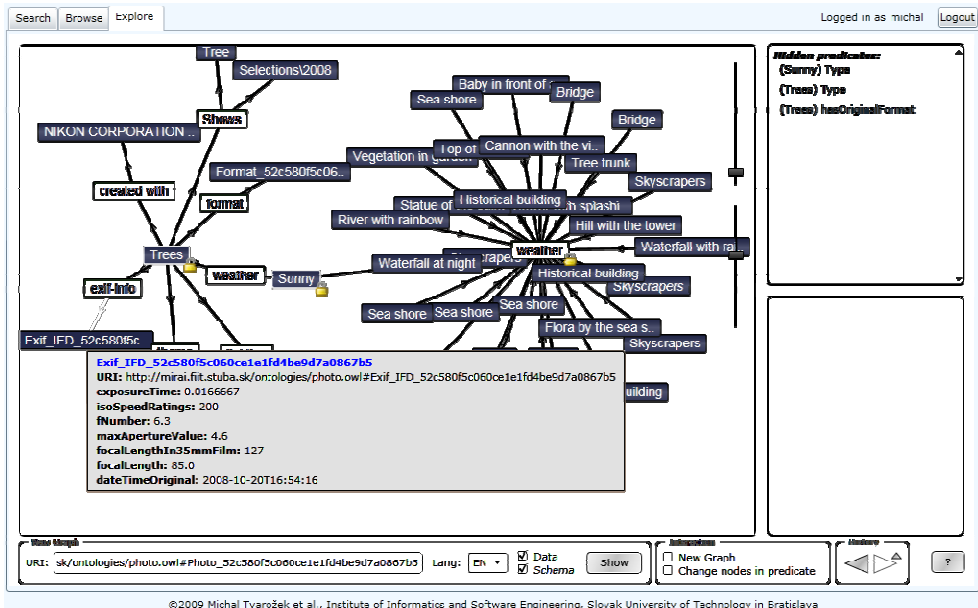


Figure 42. Example of our generated graph-view exploration interface. Dark nodes represent individual resources, white nodes correspond to relations (top). Hovering over nodes shows the attributes of a node (centre); additional tools include zooming, node hiding and history (right), with additional filtering options for languages and data/schema only visualization (bottom). Note that the colour scheme has been modified for printing purposes.

Apart from traditional view panning, users can use two zoom options – regular zoom enlarges or shrinks the view, advanced zoom spatially expands dense node clusters to make them less crowded. Lastly to further improve user orientation, we use personalization to adapt the displayed properties and/or attributes, while also allowing users to manually customize the visible properties of resources thus reducing information overload.

### 7.3 Revisitation and orientation support

The *history view* provides a tree-based visualization of search and browsing history that improves user orientation within complex navigation sessions and provides revisitation support for previously discovered (distributed) information during exploratory search sessions. We continually record user actions performed within our browser (e.g., facet selections, result exploration) and construct a tree of query modifications and result visits (see Figure 36). The tree is shown to users while they are browsing and also stored for future reference and processing.

We devised two interconnected approaches to revisitation support:

- *Search History Tree* – an in-session tree-based history visualization,

- *Semantic History Map* – an interactive, semantically organized, graph-based visualization of longer-term browsing history that shows the original context of individual history entries.

Our method records user sessions (i.e., queries and visited web resources), identifies and separates individual user goals (i.e., coherent user sessions with similar terms), preserves their context by persistently storing history trees corresponding to relations between queries and visited web resources, and ultimately synthesizes navigable graphs from extracted terms, visited resources and user goals.

We identify user agendas (i.e., goals users aimed to achieve) defined as a set of weighted terms related to individual sessions. We extract terms from queries and from visited results using term extraction approaches, and modify weights of extracted terms by the factor of user interest in the result, computed based on time spent on a result or after explicit bookmarking. We employ cosine similarity, with vectors consisting of weighted terms, multiplied by the factor of time elapsed between the last two actions to measure the distance between the actual agenda and a new query in order to distinguish different user agendas.

Search history tree also provides guidance for complex search sessions via full-text search and exploits implicitly or explicitly discovered item relevance (e.g., via user bookmarks or click-stream analysis).

Lastly, we combine individual history trees into a single history map by merging common history tree nodes (e.g., result visits, queries). The history map covers a user's entire browsing history, with support for keyword search and personalized presentation (e.g., hiding less visited subgraphs).

### 7.3.1 Search history tree

Continuing our original user scenario, we describe Search history tree by showing how Alice, a new resident of London, can find a restaurant serving Chinese crispy duck and preferably also fried ice cream for dessert (see Figure 43). Alice starts with the query “Chinese food” and immediately visits two websites about Chinese cuisine creating two *web document* nodes with thumbnails. As this was not what she was looking for, she adds “London” to her query creating a new *query node*, which results in sites referring to restaurants. She now adds “crispy duck” and later simplifies the query as her husband does not like “crispy duck”. Next, she searches for fried ice cream by substituting “fried ice cream” for “duck” creating a new *query node* connected to the common ancestor. As the results are irrelevant, Alice examines the SHT, finds the query that returned the best results – “Chinese food London” – and clicks that node in the search history tree to bring up those results again for closer examination.

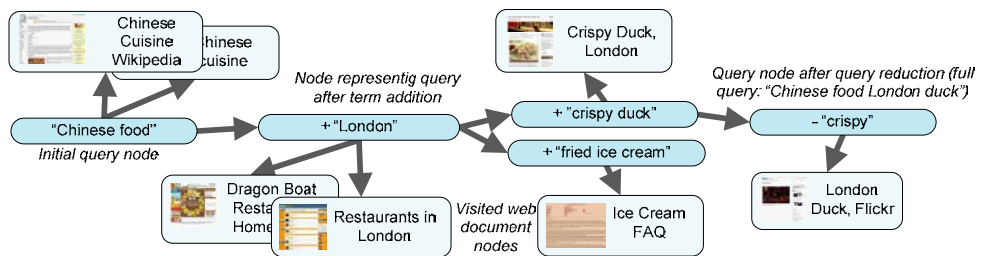


Figure 43. A search session as shown with Search history tree (normally shown vertically). Query nodes (shown in the middle layer) display information about query modification and form the core of the session. Web document nodes with thumbnails are attached to queries from which they were accessed.

Search history tree continuously records user activity in a browser (e.g., queries, back button use, result visits) and constructs a tree-based representation of query modifications. Queries are defined as either full text query changes or faceted restriction changes, when a semantically rich corpus is explored (as shown in Figure 36, p. 86). During sessions (i.e., at creation time) the purpose is to provide orientation support within recent queries and results, and streamline revisitation of results or queries. We also store history trees for future reference and processing.

We define sessions based on goals that users want to achieve rather than instances of web search applications. A goal is defined as a set of weighted terms related to individual sessions. Crucial is the correct recognition of different goals i. e. the correct grouping of individual web search log entries. This is a non-trivial task as users seldom work on single task in a single browser instance consequently requiring the analysis of the semantics of the performed user actions.

Prior to session identification, we determine:

- *What search logs to cluster – queries bundled with subsequent search results.* We recognize two types of web search logs – query entries and the corresponding visited results. However, from a user agenda point of view, a single query with subsequent result visits serves the same goal so we consider it to be a single element for clustering, represented by the aggregated vector and time span. We preserve the inner structure of the element for later stages, but that is transparent to session identification process.
- *How to compute their term and URI vectors.* Search history tree parses queries into words and using WordNet.net uses lemmatization to create weighted term vectors (excluding stopwords). Using external term extraction services (TagTheNet, OpenCalais), it retrieves term vectors of the visited results. Existing metadata and other related resources are added to vectors as URIs. The combined vector of the group is afterwards computed as the sum of the query vector and normalized sum

of all visit vectors. The normalization is required to suppress “overrun” by general terms in the aggregated vector produced by term extraction services.

- *How to compare similarity of query-result groups.* When users create a new query-result group (by entering a new query), the group's aggregated vector is compared with recently identified sessions. Each session is characterized by an aggregated term vector of its members (i.e., the normalized sum of the group's aggregated vectors). When resolving similarity, two criteria are commonly considered: term vector cosine similarity and time distance (Zhang & Nasraoui, 2006), (Huang & Efthimiadis, 2009). We adopted this approach and combine criteria using the  $N \times N \rightarrow N$  fuzzy function. The output of the fuzzy function is the final decision whether to continue in an existing session or start a new session: *certain continuation*, *weak continuation*, *uncertain*, *weak split*, *certain split*. If there are multiple candidates for a session continuation (more than one session is similar to the actual query), the query is attached to the one with the best score.

### 7.3.2 Semantic history map

Individual Search history trees are synthesized into a Semantic history map – graph of terms and web resources.

Let us consider Alice's Semantic history map comprising two sessions, one dealing with Chinese food, another performed to find cheap local lunch facilities (see Figure 44). Both sessions deal with similar topics and are bound closely together by merging identical results (restaurant portals) and by word proximity (food – lunch). Alice can navigate the map in order to revisit or reconstruct information distributed among several documents or sessions in the past.

In order to provide full-text search capability, we create two term indexes. The *item index* reflects characteristics of individual history entries, the *goal index* lists whole session trees and their overall term properties. Therefore, the results of history search are twofold:

- *Past goal summary* representing a whole session from the user goal index, ideal as a starting point for revisitation of distributed information.
- *Past query or web search result* corresponding to an individual history entry from the item index. Tooltips show the original context of the item, i.e. the neighbouring elements in its original Search history tree. The context serves as a cue for users, in addition to the document's text snippet or related terms, to recall whether it was the desired target document or not.

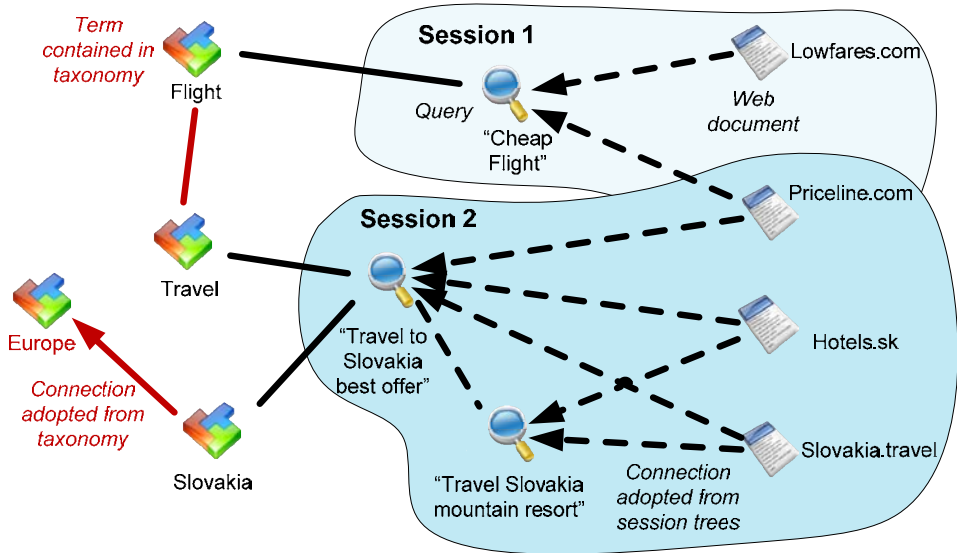


Figure 44. Example of a Semantic history map. Bubbles denote original session trees (right). Folksonomy terms (left) are linked to query terms (centre) and web resources (right) based on data derived from browsing history.

The Semantic history map is constructed via merging stored Search history trees into a single graph via matching identical terms and resources from different history trees. Since this alone may produce too few connections, we also connect terms by exploiting the existing folksonomy of Delicious<sup>20</sup>. We address dense (sub)graphs or too many irrelevant connections via term filters, item relevance ratings and successive filtering.

The creation of Semantic history maps follows these steps:

- 1) Copy each Search history tree into the Semantic history map and transform query nodes from history trees into term nodes of the Semantic history map.
- 2) Merge multiple identical web search results or queries into single nodes.
- 3) Preserve original multi-term queries as term nodes. For each particular term create a new term and attach it to the original query as a predecessor.
- 4) If any of the Semantic history maps' terms is also present in the external folksonomy, add all its folksonomy neighbours to the Semantic history map (this will load directly related parts of the folksonomy into the map).
- 5) Connect multi-term queries with their subqueries. If the term set of query *A* is a subset of the term set of query *B* then *A* is a subquery of *B*.

<sup>20</sup> Delicious, <http://del.icio.us>

## 7.4 Discussion and evaluation

Due to the nature of our multi-paradigm exploration approach and the current state of exploratory search evaluation methodologies, it is not practically feasible to perform an analytical evaluation of the whole approach. Furthermore, several of its global aspects can only be reasonable evaluated via proof of concept validation or qualitatively as has been done in the past with other approaches (i.e., it works and makes sense, or the users liked this more than something else). We thus focus on evaluation of individual approaches based on layered evaluation principles:

- A user study with our graph exploration approach, the goal being to gather user feedback on the generated GUI and its usefulness for Semantic Web exploration.
- A user study of our history-based orientation and revisitation support approaches.

### Data

We employ the same domain ontology of images as in the previous chapter, with about 8,000 manually and semi-automatically annotated images. To restate, the ontology consists of 35 classes, 50 properties (including relations and attributes), more than 32,000 individuals and in excess of 150 000 facts. For individual photos, the ontology describes EXIF metadata as supplied by the camera, information about formats in which the photos are available (e.g., resolution, aspect ratios), and optional additional annotations such as the author, the object and background of the photo, the place, overall theme and expression, lighting conditions, weather and the event to which the photo belongs.

### Methodology

In the user study with our graph exploration interface, we made our browser available to a target group of 10 end-users aged between 20 and 25 years with an IT background. As none of the users had previous knowledge of Semantic Web principles nor had used similar graph-based tools before, each user was given a brief introduction about the functionality of the browser. Next, the users were asked to complete a set of 5 tasks using the browser which also counted the time and number of clicks made (e.g., finding a specific image, discovering image properties or getting a better understanding of the domain). Lastly, each user was asked to fill out a questionnaire with the results of the tasks and his experience with the browser.

We evaluate Search history tree and Semantic history map over generic web documents and also in conjunction with our personalized faceted semantic browser *Factic*, which facilitates exploratory search over a collection of semantically annotated photographs or scientific publications respectively.

In our controlled experiment we evaluated the quantitative benefit of our history approach with a given set of exploratory and query answering tasks given a time limit against a set of baseline approaches. As a controlled experiment's time span was not long



enough to cover the evaluation of long-term evolution and use of a user's personal Semantic history map, its primary goal was to evaluate orientation support during complex search sessions. We implicitly measure task success rate and the time spent on individual tasks, and gather explicit user feedback via post-experiment questionnaires.

In our uncontrolled user study, volunteers with experience with existing baseline history tools (standard browsers with mature history extensions) will be offered to use our approach (application) as their primary search tool. The focus of this longer-term study is to gather real-world usage data and also (qualitative) feedback from users via questionnaires with the primary goal being Semantic history map validation. The key idea is to confront the overall number of revisitations with the number of those where Semantic history map was used. Within this scope, we distinguish cases when users use Semantic history map directly or tried to search with the regular search function first.

Our secondary goal is to measure user affinity for new alternative widgets (graphs) instead of classic approaches (graph navigation vs. back button usage), or gather user feedback on the proposed approach/application in general.

## Results and lessons learned

The user study with the graph exploration interface showed that 9 out of 10 users managed to find the specified image, although the time required varied widely – 141 seconds and 8 clicks were required on average, although the fastest user needed less than 50 seconds while the slowest one required almost 5 minutes. Overall, the users managed to answer 75% of the questions correctly leaving 25% false answers (this also includes answers that were close to the correct ones, but not exactly right).

Based on these results, we conclude that graph-based exploration is viable for Semantic Web browsing as most users were able to accomplish the given tasks despite having no prior experience with a similar interface. Still, improvements to layouting and node selection are necessary to improve understandability and task times, which was also confirmed by user feedback which indicates that non-expanded graphs are easy to understand (rating 4.5 on a 5 level Likert scale), while expanded graphs are less readable (rating 3.4).

Further feedback indicates that although response times were generally acceptable, some operations took too long to complete (e.g., loading the new graph after expanding a node took sometimes too long).

Initial experiments with our prototype Search history tree integrated with the faceted browser indicate promising results in terms of improved user orientation in the already explored part of the information space. We plan to work on its evaluation next.



## 8 Conclusions

---

### 8.1 Multi-paradigm exploration summary

Today, effective access to information has already become crucial to many aspects of daily life; in the corporate environment it is often paramount to operation efficiency and market success, in a personal environment it is often a matter of convenience and user satisfaction.

We described our novel approach to multi-paradigm faceted exploration of Semantic Web content with specific focus on personalization, user interface generation including facet generation, result overview generation and graph view generation.

Our exploratory search approach offers a combined interface for both searching and browsing, and is suited for effective navigation in large open information spaces represented by OWL ontologies. It can also be used for semantic information retrieval where the search query is visually created via navigation – the selection of restrictions in the set of available facets. Consequently, our approach provides these benefits to end users:

- Multiple adaptive views
- Information overload prevention
- Orientation and guidance support
- Social navigation and recommendation

#### Multiple adaptive views

Users can choose from several visualization options by selecting one of the many available views, which display increasingly more detailed information about individual search results (ontology instances) or visualize them in different ways (e.g., text, graphs, images). The attributes of the displayed instances are adaptively chosen based on their estimated relevance derived from the user model.

Moreover, the faceted browser shows instances of different types so that users can seamlessly switch from browsing/searching for e.g. publications to conferences, then to authors and back to publications.

#### Information overload prevention

Based on facet and restriction relevance we reduce the total number of accessible facet categories in order to allow users to find relevant facets and restrictions more efficiently without having to constantly scroll several screens down, e.g. due to many facets. The selection of appropriate facet types and displayed restrictions is performed automatically based on their relevance in the user model and based on the current in-session user behaviour so that it matches both long-term user interests and short-term user goals.

This reduces the overall information overload during exploratory search sessions and thus the cognitive load on users which stems from a more complex user interface compared to traditional web search engines. Similarly, view adaptation tailors the presented information to the estimated needs of users by selecting the most relevant attributes of results for visualization further improving user experience by reducing information.

### **Orientation and guidance support**

User orientation is improved via facet and restriction annotation, which include the number of instances that satisfy a restriction and a textual description of their meaning. Individual restrictions can be further annotated with background colour (e.g., indicating their relation to users' field of work), while individual search results are annotated based on their relation to a given set of instances (e.g., already read or the author's own publications) by means of an external concept comparison tool.

Facets are reordered based on their estimated user relevance thus recommending the most relevant facets, while the most relevant restrictions are recommended to provide navigation shortcuts. Moreover, we recommend the most relevant search results by ordering them using external ordering tools.

### **Social navigation and recommendation**

We take advantage of other users' preferences in the evaluation of concept relevance. *Global relevance* describes the overall “popularity” of concepts while *cross relevance* also considers the relations between users. Thus we can recommend a publication if it is relevant for many researchers in the field of Adaptive Hypermedia and the user is also interested in Adaptive Hypermedia or a generic publication that seems to be relevant for many users.

## **8.2 Contribution**

Our results in the job offers and digital image domains have shown the viability of the proposed approaches (personalization, faceted interface generation, graph exploration) for their intended purposes in terms of their practicality (i.e., it can be done) and improved user experience (i.e., improved task times, better understanding of the information space, efficient resource revisitation).

Although the described method was primarily intended for Semantic Web repositories and possibly Linked data exploration, most of the described principles could be extended to Deep Web relational databases or existing content management systems, provided that metadata describing the structure of the information space were available.

We have also discovered some limitations of current technology, which lie mainly with the immaturity of some existing semantic technologies (e.g., databases) and the

overall scope of the work, where proper cloud based solutions would be necessary to insure web scale scalability and satisfactory performance. The main problems included database scalability issues due to query complexity and latency of processing over remote repositories due to network delays.

Our main contributions lie in the development of *novel methods for navigation and presentation*, and in the combination of approaches from the Semantic Web, Social Web and Adaptive Web initiatives. We devised *a comprehensive faceted exploration approach for the Semantic Web* and claim specific contribution to:

- **Multi-paradigm exploration** – *integrating* view-based, content-based and keyword-based search with advanced adaptive visualizations and incremental graph exploration of both content and browsing history.
- **Personalized recommendation** – *devising a dynamic method* of facet and restriction adaptation based on semantic logging of user action and continuous evaluation of the devised ontological user and relevance models.
- **Exploratory interface generation** – *devising a method* for facet identification in ontological metadata, its transformation into interface widgets and their mapping onto the ontological querying backend (e.g., semantic search engines).

We thus improve upon the state of the art *information exploration possibilities by providing end-users with effective means for browsing, presentation and understanding* by incorporating semantics, adaptation, personalization and collaboration for seamless access to web resources, ultimately *enabling end-user grade exploration of the Adaptive Social Semantic Web* and achieving our original goals:

- *Empowering end-users with access to semantic information spaces* by providing an end-user grade exploratory browser for the Semantic Web with interfaces for effective query formulation, result overview browsing and individual result exploration.
- *Facilitating the adoption of the Semantic Web* by enabling Adaptive Social Semantic Web exploration for end-users via our exploratory search browser.

An important part of the presented results has been achieved in the course of several research projects conducted at FIIT SUT and published at international venues endorsed by ACM, IEEE CS and IFIP (see Appendix A for the full list of outcomes).

### 8.3 Discussion

Our work has several important implications. First, it has the potential to *significantly improve overall user experience* in many exploratory search tasks, which already employ any of the combined approaches (e.g., faceted navigation, query by example). This is supported by the fact that *many contemporary applications already employ exploratory aspects*,

*but lack the advanced layer of personalized support provided by our approach*, as has been shown in the review of the current state of the art in chapter 3.

Second, the capability to take advantage of, process and present semantic information spaces in an end-user friendly way *allows even inexperienced users to create sophisticated semantic queries without proficiency in any semantic query language (e.g., SPARQL) and prior knowledge of the information domain*. Semantic queries have in turn much potential to improve information retrieval and consequently offer more incentive for content providers to author content with semantic metadata.

Third, due to the *increased efficiency of information access and sharing*, our approach has the potential to *improve enterprise information access thus reducing costs and improving response times towards customers* (e.g. the search in and exploration of enterprise knowledge bases of technical support data or customer data).

At present, our approach also has some limitations that prevent its straightforward application in practice. It *requires a semantic description of the information space* (e.g., in RDF/OWL), which is not always readily available. In an enterprise environment, the *conversion of existing data might be semi-automatic as much of the existing data is often already stored in (semi)structured form* (i.e., it has attached semantics, just in non-standard form). Although this conversion may be driven by business needs it would still present an entry cost that would have to be made. We explore the possibilities of legacy web content integration and processing as future work in chapter 9.

The computational complexity of faceted exploration and the corresponding adaptation and personalization present a scalability issue specifically with respect to semantic repositories. In practical web scale applications *this would necessitate in cloud-based systems* which we did not explore in our work. *We partly addressed the issue of scalability by offloading server-side personalization computations onto the client-side browser*, which tracks user behaviour, evaluates it and only forwards the necessary summary data to the server back-end thus reducing load and increasing end-user privacy.

## 9 Looking Ahead

---

We see several possible directions of future work with respect to the extension of our approach, some of which we have already partially explored. We worked on and devised two possible extensions of our approach that we did not fully explore:

- *Legacy web content integration*, i.e., for specific pages (e.g., personal browsing history) or for whole web sites (e.g., generating a faceted browsing interface for a typical corporate web site). This could be accomplished by taking advantage of contextual/navigational links between pages and entity extraction approaches *ultimately providing a seamless search and browsing experience for both legacy web and semantic web content*.
- *Interactive content exploration*, e.g. for digital images the selection of further exploration by hovering the mouse over annotated regions of images with dynamic selection of the next images based on these annotations. This would require fine-grained annotation of images and thus ideally a semi-automatic user friendly annotation approach.

### 9.1 Next generation exploratory (Web) browser

While the main focus of our work lies in advanced exploration of Semantic Web content, we also outlined means of integrating our approach with legacy Web browsing in what we call a *next generation exploratory (Web) browser*. Although we presented our work primarily in the Web context it can be also applied to any other application working with a large information space described by ontological metadata (e.g., enterprise knowledge bases).

In the Web context, our browser acts as an integrated tool for search when it acts like client-side a semantic search engine front-end, and for navigation when it supports navigation across a collection of “pure” information artefacts accessed via a semantic endpoint. In the Semantic Web these correspond to individual resources (or sets of resources) while in the legacy Web, they would correspond to HTML pages stripped of non-essential parts such as hard-coded site navigation menus, banners, language selectors or external links. Since these non-content parts of web pages often correspond to (navigational) metadata created by site authors to aid users in navigation they can be effectively used to create semantic annotations describing the corresponding information artefacts. For example a hierarchical site navigation menu is considered to be a facet, while the links in non-content parts are used as annotations for the content present on the pages they link to.

Our proposal is to make our exploratory browsing approach the principal means of web browsing by turning it into a fully-featured web browser ultimately replacing existing web browsers. Thus our next generation exploratory web browser would employ facets as first-class navigation tools also on regular web content, which would not be view just as

“plain web pages” but as HTML representation of information resources. Consequently, the generated views of ontological resources and the HTML visualization of legacy web content would be equally handled and presented alongside without end-users having to worry about which Web they are browsing.

To this end, we devised a lightweight semantics extraction approach for legacy Web content that crawls and pre-processes web sites on the page-level (i.e., we do not try to extract and link individual objects within a page). In order to turn legacy Web content into semantically enriched content, we gather:

- *content-related metadata*, which are derived from actual page content using term extraction algorithms,
- *usage-related metadata*, based on how users browse the specific site.

To acquire content metadata, we crawl web sites, identify page content stripped of banners, navigational menus and other “irrelevant” items. Next we index the pages, and apply several metadata extraction approaches:

- *Metadata extraction* from page content using an external term extraction library, which also queries public bookmarking systems to identify existing tags.
- *Hierarchical classification extraction* from local navigation menus interlinking web pages within a site.
- *Annotation extraction* from incoming contextual links in page content.

We acquire usage data from an external proxy server, which improves web search via social-context driven query expansion based on user action tracking and evaluation (Kramár, Barla, & Bielíková, 2010).

Consequently, each page would be indexed for full-text search, and would have additional metadata describing its size, document type, recency, links to other pages, associated topics (also classified using external resources, e.g., Delicious folksonomy), association to the local site hierarchy extracted from menus, annotations from incoming contextual links, and usage data (e.g., how many users visited the site, (anonymous) social relations to other users), which could be used for exploration via our faceted semantic browser.

## 9.2 Interactive content exploration

In order to support exploratory experience also during image viewing, we devised an *approach to interactive navigation* in image collections (i.e., selection of next images to show) where the user selects the direction of viewing by simply hovering the mouse cursor over specific (semantically annotated) areas of the image. The next images to be shown will then be selected based on the users choice, e.g. if the user hovers the cursor over cats in an



image, the next images would be that of cats in the current collection, or if he clicks that of cats from the entire information space.

In large image collections it becomes vital to devise means for effective selection of images to present to users. This might either be performed manually (e.g., by the creator of the collection who selects the best images) but soon becomes impractical with growing collection sizes. Furthermore, even if someone could preselect the images to present, this still does not solve the issues of proper image ordering and special needs or interests of particular users (i.e., results in the ‘one size fits all’ problem).

We believe this approach to be specifically useful when viewing (large) image slideshows where users are mostly passive but sometimes become interested in a particular aspect of an image. By allowing them to implicitly select the direction of the slideshow we would address the problem of image ordering and selection while also catering to the needs of specific users instead of an average solution which does not really suit anyone.



## References

---

- Adkisson, H. P. (2005). *Use of Faceted Classification*. Retrieved October 23, 2007, from Web design practices: <http://www.webdesignpractices.com/navigation/facets.html>
- Andrejko, A., Barla, M., Bieliková, M., & Tvarožek, M. (2006). Softvérové nástroje pre získavanie charakteristík používateľa. In P. Vojtáš, & T. Skopal (Ed.), *Proceedings of DATAKON '06*, (pp. 139-148). Brno, Czech Republic.
- Andrejko, A., Barla, M., Bieliková, M., & Tvarožek, M. (2007). User Characteristics Acquisition from Logs with Semantics. In A. Kelemenová, D. Kolář, A. Meduna, & J. Zendulka (Ed.), *ISIM 2007: Proceedings of the 10th International Conference on Information System Implementation and Modeling* (pp. 103-110). Hradec nad Moravicí, Czech Republic: Slezská universita v Opavě, Czech Republic.
- Auer, S., Dietzold, S., & Riechert, T. (2006). OntoWiki – A Tool for Social, Semantic Collaboration. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 736-749. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Aurnhammer, M., Hanappe, P., & Steels, L. (2006). Augmenting Navigation for Collaborative Tagging with Emergent Semantics. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 58-71. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Australian Government Information Management Office. (2004, May). *Better Practice Checklist – Practical guides for effective use of new technologies in Government: Website Navigation*. Retrieved October 23, 2007, from Better Practice Center @ AGIMO: [http://www.agimo.gov.au/\\_\\_data/assets/file/0005/33908/BPC2.pdf](http://www.agimo.gov.au/__data/assets/file/0005/33908/BPC2.pdf)
- Baeza-Yates, R., Castillo, C., & Saint-Jaen, F. (2004). Web Dynamics, Structure, and Page Quality. In M. Levene, & A. Poullovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 93-109). Springer-Verlag, Berlin Heidelberg.
- Barla, M., Bartalos, P., Bieliková, M., Filkorn, R., & Tvarožek, M. (2007). Adaptive Portal Framework for Semantic Web Applications. In M. Brambilla, & E. Mendes (Ed.), *AEWSE'07: Proceedings of the Second International Workshop on*

- Adaptation and Evolution in Web Systems Engineering* (pp. 87-93). Como, Italy: Politecnico di Milano, Milano, Italy.
- Barla, M., Tvarožek, M., & Bieliková, M. (2009). Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems. *Computing and Informatics*, 28 (4), 399-427.
- Barla, M., Tvarožek, M., & Bieliková, M. (2009). Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems. *Computing and Informatics*, 28 (4), 399-427, ISSN 1335-9150.
- Bergman, M. K. (2007, August 23). *Information is the Basis for Economic Growth*. Retrieved October 20, 2007, from AI3: <http://www.mkbergman.com/?p=395>
- Bergman, M. K. (2000). *The Deep Web: Surfacing Hidden Value*. White paper, Bright Planet.
- Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., et al. (2006). Tabulator: Exploring and analyzing linked data on the semantic web. *Proceedings of the 3rd International Semantic Web User Interaction Workshop*.
- Bieliková, M. (2003). Presentation of Adaptive Hypermedia on the Web. In L. Popelínský (Ed.), *Proceedings of DATAKON 2003*, (pp. 1-19). Brno, Czech Republic.
- Bordogna, G. C., Ronchi, S., & Psaila, G. (2009). Query Disambiguation Based on Novelty and Similarity User's Feedback. *WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (pp. 125-128). Milano, Italy: IEEE CS.
- Braak, P. t., Abdullah, N., & Xu, Y. (2009). Improving the Performance of Collaborative Filtering Recommender Systems through User Profile Clustering. *WI-IAT'09: Proceedings of Int. Joint Conference on Web Intelligence and Intelligent Agent Technology* (pp. 147-150). Milano, Italy: IEEE CS.
- Broder, A. (2002). A taxonomy of Web search. *ACM SIGIR forum*, 36 (2), 3-10.
- Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Strata, R., et al. (2000). Graph structure in the Web. *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications networking* (pp. 309-320). Amsterdam, The Netherlands: North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands.

- Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11 (1-2), 87-110.
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), 87-129.
- Cailliau, R. (1995). *A Little History of the World Wide Web*. Retrieved October 22, 2007, from W3C: <http://www.w3.org/History.html>
- Cannataro, M., & Pugliese, A. (2004). A Survey of Architectures for Adaptive Hypermedia. In M. Levene, & A. Poulovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 357-386). Springer-Verlag, Berlin Heidelberg.
- Celino, I., & Valle, E. D. (2005). Multiple Vehicles for a Semantic Navigation Across Hyper-environments. In A. Gómez-Pérez, & J. Euzenat (Ed.), *ESWC 2005: Proceedings of the 2nd European Semantic Web Conference on Research and Applications. LNCS 3532*, pp. 423-438. Heraklion, Crete, Greece: Springer-Verlag, Berlin Heidelberg.
- Dakka, W., Ipeirotis, P. G., & Wood, K. R. (2005). Automatic Construction of Multifaceted Browsing Interfaces. *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 768-775). Bremen, Germany: ACM Press, New York, NY, USA.
- De Bra, P., Aroyo, L., & Cristea, A. (2004). Adaptive Web-Based Educational Hypermedia. In M. Levene, & A. Poulovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 387-410). Springer-Verlag, Berlin Heidelberg.
- Diederich, J., & Balke, W.-T. (2007). The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems. In L. Kovács, N. Furh, & C. Meghini (Ed.), *ECDL 2007: Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries. LNCS 4675*, pp. 1-13. Budapest, Hungary: Springer-Verlag, Berlin Heidelberg.
- Ding, H., & Sølvsberg, I. T. (2005). Semantic Search in Peer-to-Peer-Based Digital Libraries. In M. Marlina, T. Sumner, & F. Shipman (Ed.), *Proceedings of JCDL 2005*. Denver, CO, USA: ACM Press, New York, NY, USA.
- Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., & Kolari, P. (2005). Finding and Ranking Knowledge on the Semantic Web. In Y. Gil, E. Motta, V. R. Benjamins,

- & M. A. Musen (Ed.), *ISWC 2005: Proceedings of the 4th International Semantic Web Conference. LNCS 3729*, pp. 156-170. Galway, Ireland: Springer Verlag, Berlin Heidelberg.
- Dobra, A., & Fienberg, S. E. (2004). How Large Is the World Wide Web. In M. Levene, & A. Poullovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 23-43). Springer-Verlag, Berlin Heidelberg.
- Dörk, M., Carpendale, S., Collins, C., & Williamson, C. (2008). Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14 (6), 1205-1212.
- Flake, G. W., Tsioutsoulouklis, K., & Zhukov, L. (2004). Methods of Mining Web Communitites: Bibliometric, Spectral, and Flow. In M. Levene, & A. Poullovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 45-68). Springer-Verlag, Berlin Heidelberg.
- Fox, E. A., & Flanagan, J. W. (2003, November 21). ETANA-DL: Managing complex information applications - An archeological digital library. 414-414. ACM Press, New York, NY, USA.
- Fox, S., Manduca, C., & Iverson, E. (2005). Building Educational Portals atop Digital Libraries. *D-Lib Magazine*, 11 (1).
- Geman, D. (2006). Interactive image retrieval by mental matching. *Proceedings of the 8th ACM international workshop on Multimedia information* (pp. 1-2). Santa Barbara, California, USA: ACM Press, New York, NY, USA.
- Guha, R., McCool, R., & Miller, E. (2003). Semantic Search. *Proc. of the 12th international conference on World Wide Web* (pp. 700-709). ACM Press, New York, NY, USA.
- Gulli, A., & Signorini, A. (2005). The indexable web is more than 11.5 billion pages. *Special interest tracks and posters of the 14th international conference on World Wide Web* (pp. 902-903). ACM Press, New York, NY, USA.
- Gurský, P., Horváth, T., Novotný, R., Vaneková, V., & Vojtáš, P. (2006). Upre: User preference based search system. *WI 2006: Proceedings of the International Conference on Web Intelligence* (pp. 841-844). Hong Kong, China: IEEE CS.
- Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T., & Weitzner, D. (2008). Web science: An Interdisciplinary Approach to Understanding the Web. *Communications of the ACM*, 51 (7), 60-69.

- Hildebrand, M., van Ossenbruggen, J., & Hardman, L. (2006). /facet: A Browser for Heterogeneous Semantic Web Repositories. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 272-285. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Huang, J., & Efthimiadis, E. N. (2009). Analyzing and evaluating query reformulation strategies in web search logs. *Proceeding of the 18th ACM conference on Information and knowledge management* (pp. 77-86). Hong Kong, China: ACM, New York, NY, USA.
- Huang, L., Ulrich, T., Hemmje, M., & Neuhold, E. J. (2001). Adaptively constructing the query interface for meta-search engines. *Proceedings of the 6th international conference on Intelligent user interfaces* (pp. 97-100). Santa Fe, New Mexico, USA: ACM Press, New York, NY, USA.
- Ishikawa, Y., & Hasegawa, M. (2007). T-Scroll: Visualizing Trends in a Time-Series of Documents for Interactive User Exploration. In L. Kovács, N. Furh, & C. Meghini (Ed.), *ECDL 2007: LNCS 4675*, pp. 235-246. Budapest, Hungary: Springer Verlag, Berlin Heidelberg.
- Jansen, B. J., Spink, A., & Pedersen, J. (2003). An Analysis of Multimedia Searching on AltaVista. *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval* (pp. 186-192). Berkeley, California, USA: ACM Press, New York, NY, USA.
- Jansen, B. J., Spink, A., Bateman, J., & Saracevic, T. (1998). Real life information retrieval: a study of user queries on the Web. *ACM SIGIR Forum*, 32 (1), 5-17.
- Kampa, S., Miles-Board, T., Carr, L., & Hall, W. (2001). *Linking with Meaning: Ontological Hypertext for Scholars*. Technical Report, University of Southampton, School of Electronics and Computer Science, Southampton, United Kingdom.
- Koren, J., Zhang, Y., & Liu, X. (2008). Personalized Interactive Faceted Search. *WWW 2008: Proceeding of the 17th International Conference on World Wide Web* (pp. 477-486). Beijing, China: ACM Press, New York, NY, USA.
- Kramár, T., Barla, M., & Bieliková, M. (2010). Disambiguating search by leveraging a social context based on the stream of users activity. *UMAP 2010: Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization. LNCS 6075*, pp. 387-392. Hawaii, USA: Springer-Verlag, Berlin Heidelberg.

- Kules, B., Capra, R., Banta, M., & Sierra, T. (2009). What do exploratory searchers look at in a faceted search interface? *JCDL '09: Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 313-322). Austin, Texas, USA: ACM, New York, NY, USA.
- Levene, M., & Wheeldon, R. (2004). Navigating the World Wide Web. In M. Levene, & A. Poullovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 117-151). Springer-Verlag, Berlin Heidelberg.
- Li, C., Yan, N., Roy, S. B., Lisham, L., & Das, G. (2010). Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia. *Proceedings of the 19th International Conference on World Wide Web* (pp. 651-660). Raleigh, North Carolina, USA: ACM, New York, NY, USA.
- Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49 (4), 41-46.
- Mayer, M. (2009). Web History Tools and Revisitation Support: A Survey of Existing Approaches and Directions. *Foundations and Trends in Human-Computer Interaction*, 2 (3), 173-278.
- Mäkelä, E., Hyvönen, E., & Saarela, S. (2006). Ontogator - a semantic view-based search engine service for web applications. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 847-860. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Mäkelä, E., Hyvönen, E., Saarela, S., & Viljanen, K. (2004). OntoViews - A Tool for Creating Semantic Web Portals. In S. A. McIlraith, D. Plexousakis, & F. van Harmelen (Ed.), *ISWC 2004: Proceedings of the 3rd International Semantic Web Conference. LNCS 3298*, pp. 797-811. Hiroshima, Japan: Springer-Verlag, Berlin Heidelberg.
- Mendes, J. F. (2004). Theory of Random Networks and Their Role in Communications Networks. In M. Levene, & A. Poullovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 69-92). Springer-Verlag, Berlin Heidelberg.
- Miniwatts Marketing Group. (2010, September). *Internet Usage Statistics - The Big Picture*. Retrieved May 9, 2010, from Internet World Stats – Usage and Population Statistics: <http://www.internetworldstats.com/stats.htm>



- Nadeem, T., & Killam, B. (2001). A Study of Three Browser History Mechanisms for Web Navigation. *IV 2001: Fifth International Conference on Information Visualisation* (pp. 13-21). Los Alamitos, CA, USA: IEEE Computer Society.
- Návrát, P., & et.al. (2007). *Proceedings of the Research Project Workshop: Tools for Acquisition, Organization and Presenting of Information and Knowledge*. Bratislava: STU Press.
- Nielsen, J. (2007). *Writing for the Web*. Retrieved July 1, 2007, from useit.com: Jakob Nielsen's Website: <http://www.useit.com/papers/webwriting/>
- O'Reilly, T. (2005, September 30). *What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software*. Retrieved October 23, 2007, from O'Reilly Media: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- Oren, E., Delbru, R., & Decker, S. (2006). Extending Faceted Navigation for RDF Data. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 559-572. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Pandit, S., & Olston, C. (2007). Navigation-aided Retrieval. *WWW 2007: Proceedings of the 16th international conference on World Wide Web* (pp. 391-400). Banff, Alberta, Canada: ACM, New York, NY, USA.
- Pant, G., Srinivasan, P., & Menczer, F. (2004). Crawling the Web. In M. Levene, & A. Poulovassilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 153-177). Springer-Verlag, Berlin Heidelberg.
- Paramythis, A., & Weibelzahl, S. (2005). A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In L. Ardissono, P. Brna, & T. Mitrovic (Ed.), *UM 2005: Proceedings of the 10th International Conference on User Modeling. LNCS 3538*, pp. 438-442. Edinburgh, Scotland, UK: Springer Verlag, Berlin Heidelberg.
- Reif, G., & Gall, H. C. (2006). An Architecture for a Semantic Portal. *DISWeb 2006: Proceedings of the International Workshop on Data Integration and Semantic Web*. Springer-Verlag, Berlin Heidelberg.
- Rocketface® Graphics. (2007). *Website Navigation*. Retrieved December 10, 2007, from How to Design a Website - Webmasters Tutorial: [http://www.rocketface.com/organize\\_website/website\\_navigation.html](http://www.rocketface.com/organize_website/website_navigation.html)

- Roy, S. B., Wang, H., Nambiar, U., Das, G., & Mohania, M. (2009). DynaCet: Building Dynamic Faceted Search Systems over Databases. *Proceedings of the International Conference on Data Engineering* (pp. 1463-1466). IEEE CS.
- Sacco, G. M., & Tzitzikas, Y. (Eds.). (2009). *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience*. Springer.
- Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems*, 21 (3), 96-101.
- schraefel, m. c., Smith, D. a., Owens, A., Russell, A., Harris, C., & Wilson, M. L. (2005). The evolving mSpace platform: leveraging the Semantic Web on the Trail of the Memex. *Hypertext* (pp. 174-183). Salzburg, Austria: ACM Press.
- Schulz, H.-J., & Schumann, H. (2006). Visualizing Graphs - A Generalized View. *IV 2006: Tenth International Conference on Information Visualisation* (pp. 166-173). Los Alamitos, CA, USA: IEEE Computer Society.
- Siberski, W., Pan, J. Z., & Thaden, U. (2006). Querying the Semantic Web with Preferences. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 612-624. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Smyth, B., & Cotter, P. (2004). MP3 - Mobile Portals, Profiles and Personalization. In M. Levene, & A. Poulouvasilis (Eds.), *Web Dynamics – Adapting to Change in Content, Size, Topology and Use* (pp. 411-433). Springer-Verlag, Berlin Heidelberg.
- Staab, S., Domingos, P., Mika, P., Golbeck, J., Ding, L., Finin, T., et al. (2005). Social Networks Applied. *IEEE Intelligent Systems*, 20 (1), 80-93.
- Stewart, R., Scott, G., & Zelevinsky, V. (2008). Idea navigation: structured browsing for unstructured text. *Proceeding of the 26th annual SIGCHI Conference on Human Factors in Computing Systems* (pp. 1789-1792). Florence, Italy: ACM, New York, NY, USA.
- Studer, R., Benjamins, R., & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, 25 (1-2), 161-198.
- Technical Advisory Service for Images. (2006, May). *A Review of Image Search Engines*. Retrieved June 27, 2007, from TASI: Technical Advisory Service for Images: <http://www.tasi.ac.uk/resources/searchengines.html>

- Brusilovsky, P., Kobsa, A., & Nejdl, W. (Eds.). (2007). *The Adaptive Web: Methods and Strategies of Web Personalization. LNCS. 4321*. Springer Berlin Heidelberg.
- The Knowledge Management Connection. (2006). *Faceted Classification of Information*. Retrieved October 23, 2007, from The Knowledge Management Connection: <http://www.kmconnection.com/DOC100100.htm>
- Tvarožek, M. (2006). Personalized Navigation in the Semantic Web. In V. P. Wade, H. Ashman, & B. Smyth (Ed.), *Lecture Notes in Computer Science, AH 2006: Adaptive Hypermedia and Adaptive Web-Based Systems. Vol. 4018*, pp. 467-472. Springer-Verlag, ISSN 0302-9763.
- Tvarožek, M., & Bieliková, M. (2008). Collaborative Multi-Paradigm Exploratory Search. *WSW 2008: Proceedings of the Hypertext 2008 Workshop on Collaboration and Collective Intelligence* (pp. 29-33). Pittsburgh, USA: ACM Press, New York, NY, USA.
- Tvarožek, M., & Bieliková, M. (2010). Generating Exploratory Search Interfaces for the Semantic Web. In P. Forbrig, F. Paternò, & A. M. Pejtersen (Ed.), *IFIP - World Computer Congress 2010: Human-Computer Interaction Symposium* (pp. 175-186). Brisbane, Australia: Springer.
- Tvarožek, M., & Bieliková, M. (2007). Personalized Faceted Navigation for Multimedia Collections. *SMAP 2007: Proceedings of the 2nd International Workshop on Semantic Media Adaptation and Personalization*. London, United Kingdom: IEEE CS.
- Tvarožek, M., & Bieliková, M. (2007). Personalized Faceted Navigation in the Semantic Web. In L. Baresi, P. Fraternali, & G.-J. Houben (Ed.), *ICWE 2007: Proceedings of the International Conference on Web Engineering. LNCS 4607*, pp. 511-515. Como, Italy: Springer-Verlag, Berlin Heidelberg.
- Tvarožek, M., & Bieliková, M. (2009). Reinventing the Web Browser for the Semantic Web. *WIRSS 2009: Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, Workshop on Web Information Retrieval Support Systems* (pp. 113-116). Milan, Italy: IEEE CS.
- Tvarožek, M., & Bieliková, M. (2008). Visualization of Personalized Faceted Browsing. In P. Forbrig, F. Paternò, & A. M. Pejtersen (Ed.), *IFIP - International Federation for Information Processing: Human-Computer Interaction Symposium. IFIP 272*, pp. 213-218. Milan, Italy: Springer.

- Tvarožek, M., & Bielíková, M. (2008). Visualization of Personalized Faceted Browsing. In P. Forbrig, F. Paternò, & A. M. Pejtersen (Ed.), *IFIP - World Computer Congress 2008: Human-Computer Interaction Symposium. IFIP 272*, pp. 213-218. Milan, Italy: Springer.
- Tvarožek, M., Adam, M., Barla, M., Sivák, P., & Bielíková, M. (2006). Spot-it: Going Beyond the Vision Loss Boundaries. *ISIE2006: Proceedings of International Symposium on Intelligent Environments - Improving the quality of life in a changing world* (pp. 67-76). Cambridge, United Kingdom: Microsoft Research Ltd, Cambridge.
- Völkel, M., Krötzsch, M., Vrandečić, D., Haller, H., & Studer, R. (2006). Semantic Wikipedia. In C. Goble, & M. Dahlin (Ed.), *WWW 2006: Proceedings of the 15th international conference on World Wide Web* (pp. 585-594). Edinburgh, Scotland, UK: ACM Press, New York, NY, USA.
- Wang, S., Jing, F., He, J., Du, Q., & Zhang, L. (2007). IGroup: presenting web image search results in semantic clusters. *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 587-596). San Jose, California, USA: ACM Press, New York, NY, USA.
- Wang, T. D., & Parsia, B. (2006). CropCircles: Topology Sensitive Visualization Class Hierarchies. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, et al. (Ed.), *ISWC 2006: Proceedings of the 5th International Semantic Web Conference. LNCS 4273*, pp. 695-708. Athens, GA, USA: Springer-Verlag, Berlin Heidelberg.
- Weinrich, H., Obendorf, H., Herder, E., & Mayer, M. (2006). Off the beaten tracks: exploring three aspects of web navigation. In C. Goble, & M. Dahlin (Ed.), *WWW 2006: Proceedings of the 15th international conference on World Wide Web* (pp. 133-142). Edinburgh, Scotland, UK: ACM Press, New York, NY, USA.
- Wilson, M. L., & schraefel, m. (2008). A Longitudinal Study of Exploratory and Keyword Search. *JCDL 2008: Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 52-56). Pittsburgh, USA: ACM Press, New York, NY, USA.
- Wilson, M. L., schraefel, m. c., & White, R. W. (2009). Evaluating Advanced Search Interfaces using Established Information-Seeking Models. *Journal of the American Society for Information Science and Technology*, 60 (7), 1407-1422.

- Wynar, B. S., & Taylor, A. G. (1992). *Introduction to cataloging and classification* (8th ed.). Libraries Unlimited.
- Yee, K.-P., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted metadata for image search and browsing. *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 401-408). Ft. Lauderdale, Florida, USA: ACM Press, New York, NY, USA.
- Yuanguai, L., Motta, E., & Domingue, J. (2004). OntoWeaver-S: Supporting the Design of Knowledge Portals. In E. Motta, N. Shadbolt, A. Stutt, & N. Gibbins (Ed.), *EKAU 2004: Proceedings 14th International Conference on Knowledge Engineering and Knowledge Management. LNCS 3257*, pp. 216-230. Whittlebury Hall, Northamptonshire, UK: Springer-Verlag, Berlin Heidelberg.
- Zhang, J., & Marchionini, G. (2005). Evaluation and evolution of a browse and search interface: relation browser. *Proceedings of the 2005 national conference on Digital government research* (pp. 179-188). Atlanta, Georgia, USA: Digital Government Research Center.
- Zhang, Z., & Nasraoui, O. (2006). Mining search engine query logs for query recommendation. *Proceedings of the 15th international conference on World Wide Web* (pp. 1039-1040). Edinburgh, Scotland, UK: ACM, New York, NY, USA.
- Zwol, R. v., & Sigurbjornsson, B. (2010). Faceted exploration of image search results. *Proceedings of the 19th International Conference on World Wide Web* (pp. 961-970). Raleigh, North Carolina, USA: ACM, New York, NY, USA.



# Appendix A      Dissertation Outcomes

---

## A.1      List of projects participated in

Several of the presented results have been achieved in the course of the following research projects conducted at FIIT SUT, which I took part in as a researcher:

- Project NAZOU: Tools for Acquisition, Organisation, and Maintenance of Knowledge in an Environment of Heterogeneous Information Resources (1025/2004)  
Project leader: Pavol Návrat for STU  
Supported by: State programme of research and development “Establishing of Information Society”  
Duration: September 2004 – May 2008
- Project MAPEKUS: Modelling and Acquisition, Processing and Employing Knowledge about User Activities (APVT-20-007104)  
Project leader: Mária Bieliková  
Supported by: Slovak Research and Development Agency  
Duration: January 2005 – April 2008
- Project PeWePro: Adaptive web-based portal for learning programming (KEGA 3/5187/07)  
Project leader: Mária Bieliková  
Supported by: Cultural and Educational Grand Agency of the Ministry of Education of Slovak Republic  
Duration: January 2005 – December 2009
- Adaptive Social Web and its services for information access (VG 1/0508/09)  
Project leader: Pavol Návrat  
Supported by: Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Sciences  
Duration: January 2009 – December 2011
- Models of software systems in the semantic web environment (VG 1/3102/06)  
Project leader: Pavol Návrat  
Supported by: Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Sciences  
Duration: January 2006 – December 2008
- E-learning support via social relations and intelligent recommendation (KG 028-025STU-4/2010)

Project leader: prof. Ing. Mária Bieliková, PhD.

Supported by: Cultural and Educational Grand Agency of the Ministry of Education of Slovak Republic

Duration: 01/2010 – 12/2011

- Project of European structural funds ‘Centre of Excellence’ SMART
- Project of European structural funds for support of young researchers

## **A.2 List of awards received**

- Winner of the 2<sup>nd</sup> place in the ACM.SRC Grand Finals 2010
- Winner of the 2010 doctoral studentship grant “We support individualities” offered by the grant agency Intenda for the finishing of doctoral studies
- Winner of the title “Student figure of the year 2008/2009” in the field of mathematics, physics and informatics awarded by Junior Chamber International and endorsed by the President of the Slovak republic
- Winner of Werner von Siemens Excellence Award 2007 for my diploma thesis
- 1<sup>st</sup> place in the Czecho-Slovak ACM.SRC 2005 organized by the Czech chamber of ACM for the paper “Spot-it Going Beyond the Vision Loss Boundaries”, co-authored as a part of a four member team
- 2<sup>nd</sup> place in the Czecho-Slovak ACM.SRC 2005 organized by the Czech chamber of ACM for the paper “Personalized navigation in the Semantic Web”
- “Best paper” awards for papers at the IIT.SRC 2005, 2006, 2007

## **A.3 List of publications**

### **A.3.1 Publications with outstanding international recognition (A)**

Barla, M., Tvarožek, M., & Bieliková, M. (2009). Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-Based Systems. *Computing and Informatics*, 28 (4), 399-427, ISSN 1335-9150.

Tvarožek, M. (2006). Personalized Navigation in the Semantic Web. In V. P. Wade, H. Ashman, & B. Smyth (Ed.), *Lecture Notes in Computer Science, AH 2006: Adaptive Hypermedia and Adaptive Web-Based Systems*. Vol. 4018, pp. 467-472. Springer-Verlag, ISSN 0302-9763.

Tvarožek, M., & Bieliková, M. (2010). Generating Exploratory Search Interfaces for the Semantic Web. In P. Forbrig, F. Paternò, & A. M. Pejtersen (Ed.), *IFIP - World*



Computer Congress 2010: Human-Computer Interaction Symposium (pp. 175-186). Brisbane, Australia: Springer.

Tvarožek, M., & Bieliková, M. (2008). Visualization of Personalized Faceted Browsing. In P. Forbrig, F. Paternò, & A. M. Pejtersen (Ed.), IFIP - World Computer Congress 2008: Human-Computer Interaction Symposium. IFIP 272, pp. 213-218. Milan, Italy: Springer.

### **A.3.2 Publications with international recognition (B)**

Bartalos, P., Barla, M., Frivolt, G., Tvarožek, M., Andrejko, A., Bieliková, M., et al. (2007). Building an Ontological Base for Experimental Evaluation of Semantic Web Applications. In J. Van Leeuwen, G. F. Italiano, W. van der Hoek, H. Sack, C. Meinel, & F. Plášil (Ed.), SOFSEM 2007: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science. LNCS 4362, pp. 682-692. Harrachov, Czech Republic: Springer-Verlag, Berlin Heidelberg.

Filkorn, R., Barla, M., Bartalos, P., Sivák, P., Szobi, K., & Tvarožek, M. (2007). Ontology as an Information Base for a Family of Domain Oriented Portal Solutions. In G. Knapp, G. Wojtkowski, J. Zupancic, & S. Wrycza (Ed.), Advances in Information Systems Development: New Methods and Practice for the Networked Society - Proc. of 15th Int. Conf. on Information Systems Development, ISD'06. I, pp. 423-433. Budapest, Hungary: Springer-Verlag, Berlin Heidelberg.

Šimko, J., Tvarožek, M., & Bieliková, M. (2010). Semantic History Map: Graphs Aiding Web Revisitation Support. DEXA/WebS 2010: 9th International Workshop on Web Semantics (pp. 206-210). Bilbao, Spain: IEEE Computer Society Press.

Tvarožek, M., & Bieliková, M. (2007). Adaptive faceted browser for navigation in open information spaces. Proceedings of the 16th international conference on World Wide Web (pp. 1311-1312). Banff, Alberta, Canada: ACM Press, New York, NY, USA.

Tvarožek, M., & Bieliková, M. (2010). Bridging Semantic and Legacy Web Exploration: Orientation, Revisitation and Result Exploration Support. WIRSS 2010: Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, Workshop on Web Information Retrieval Support Systems (pp. 326-329). Toronto, Canada: IEEE CS.

Tvarožek, M., & Bieliková, M. (2008). Collaborative Multi-Paradigm Exploratory Search. WSW 2008: Proceedings of the Hypertext 2008 Workshop on Collaboration and Collective Intelligence (pp. 29-33). Pittsburgh, USA: ACM Press, New York, NY, USA.

- Tvarožek, M., & Bielíková, M. (2010). Factic: Personalized Exploratory Search in the Semantic Web. In B. Benatallah (Ed.), *Lecture Notes in Computer Science, ICWE 2010: Web Engineering*. Vol. 6189, pp. 528-531. Springer-Verlag, ISSN 0302-9763.
- Tvarožek, M., & Bielíková, M. (2007). Personalized Faceted Browsing for Digital Libraries. In N. Furf, L. Kovacs, & C. Meghini (Ed.), *ECDL 2007: Proceeding of the European Conference on Research and Advanced Technology for Digital Libraries*. LNCS 4675, pp. 485-488. Budapest, Hungary: Springer-Verlag, Berlin Heidelberg.
- Tvarožek, M., & Bielíková, M. (2007). Personalized Faceted Navigation for Multimedia Collections. *SMAP 2007: Proceedings of the 2nd International Workshop on Semantic Media Adaptation and Personalization* (pp. 104-109). London, United Kingdom: IEEE CS.
- Tvarožek, M., & Bielíková, M. (2009). Personalized Faceted Navigation in Semantically Enriched Information Spaces. In M. C. Angelides, P. Mylonas, & M. Wallace (Eds.), *Advances in Semantic Media Adaptation and Personalization* (Vol. 2, pp. 181-201). CRC Press.
- Tvarožek, M., & Bielíková, M. (2007). Personalized Faceted Navigation in the Semantic Web. In L. Baresi, P. Fraternali, & G.-J. Houben (Ed.), *ICWE 2007: Proceedings of the International Conference on Web Engineering*. LNCS 4607, pp. 511-515. Como, Italy: Springer-Verlag, Berlin Heidelberg.
- Tvarožek, M., & Bielíková, M. (2008). Personalized View-Based Search and Visualization as a Means for Deep/Semantic Web Data Access. *WWW 2008: Proceedings of the 17th International Conference on World Wide Web* (pp. 1023-1024). Beijing, China: ACM Press, New York, NY, USA.
- Tvarožek, M., & Bielíková, M. (2009). Reinventing the Web Browser for the Semantic Web. *WIRSS 2009: Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, Workshop on Web Information Retrieval Support Systems* (pp. 113-116). Milan, Italy: IEEE CS.
- Tvarožek, M., Barla, M., & Bielíková, M. (2007). Personalized Presentation in Web-Based Information Systems. In J. Van Leeuwen, G. F. Italiano, W. van der Hoek, H. Sack, C. Meinel, & F. Plášil (Ed.), *SOFSEM 2007: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science*. LNCS 4362, pp. 796-807. Harrachov, Czech Republic: Springer-Verlag, Berlin Heidelberg.
- Tvarožek, M., Barla, M., Frivolt, G., Tomša, M., & Bielíková, M. (2008). Improving Semantic Search Via Integrated Personalized Faceted and Visual Graph Navigation. In V. Geffert, J. Karhumäki, A. Bertoni, B. Preneel, P. Návrat, & M. Bielíková (Ed.), *SOFSEM 2008: Proceedings of the 34rd Conference on Current Trends in Theory and Practice of Computer Science*. LNCS 4910, pp. 778-789. Nový Smokovec, Slovakia: Springer-Verlag, Berlin Heidelberg.

### **A.3.3 Publications with national recognition (C)**

- Andrejko, A., Barla, M., Bielíková, M., & Tvarožek, M. (2007). User Characteristics Acquisition from Logs with Semantics. In A. Kelemenová, D. Kolář, A. Meduna, & J. Zendulka (Ed.), *ISIM 2007: Proceedings of the 10th International Conference on Information System Implementation and Modeling* (pp. 103-110). Hradec nad Moravicí, Czech Republic: Slezská universita v Opavě, Czech Republic.
- Barla, M., Bartalos, P., Bielíková, M., Filkorn, R., & Tvarožek, M. (2007). Adaptive Portal Framework for Semantic Web Applications. In M. Brambilla, & E. Mendes (Ed.), *AEWSE'07: Proceedings of the Second International Workshop on Adaptation and Evolution in Web Systems Engineering* (pp. 87-93). Como, Italy: Politecnico di Milano, Milano, Italy.
- Tvarožek, M., Adam, M., Barla, M., Sivák, P., & Bielíková, M. (2006). Spot-it: Going Beyond the Vision Loss Boundaries. *ISIE2006: Proceedings of International Symposium on Intelligent Environments - Improving the quality of life in a changing world* (pp. 67-76). Cambridge, United Kingdom: Microsoft Research Ltd, Cambridge.

### **A.3.4 Project workshops**

- Andrejko, A., Barla, M., & Tvarožek, M. (2006). Comparing Ontological Concepts to Evaluate Similarity. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition, Organization and Maintenance of Knowledge* (pp. 71-78). Bystrá dolina, Nízke Tatry, Slovakia: SUT Press, Bratislava, Slovakia.
- Barla, M., & Tvarožek, M. (2006). Automatic Acquisition of Comprehensive Semantic User Activity Logs. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition, Organization and Maintenance of Knowledge* (pp. 169-174). Bystrá dolina, Nízke Tatry, Slovakia: SUT Press, Bratislava, Slovakia.
- Barla, M., Bartalos, P., Bielíková, M., Filkorn, R., & Tvarožek, M. (2007). Integration and Presentation Platform for Semantic Web Applications. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition, Organization and Maintenance of Knowledge*. 2, pp. 48-62. Poľana, Slovakia: SUT Press, Bratislava, Slovakia.
- Barla, M., Bartalos, P., Filkorn, R., Sivák, P., Szobi, K., & Tvarožek, M. (2006). An Ontological Representation Based Approach Towards Automated Building of Data Acquisition Portals. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition,*

Organization and Maintenance of Knowledge (pp. 223-230). Bystrá dolina, Nízke Tatry, Slovakia: SUT Press, Bratislava, Slovakia.

Tvarožek, M. (2006). Personalized Navigation in the Semantic Web. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition, Organization and Maintenance of Knowledge* (pp. 182-192). Bystrá dolina, Nízke Tatry, Slovakia: SUT Press, Bratislava, Slovakia.

Tvarožek, M., & Bielíková, M. (2007). Adaptive Faceted Browsing in Job Offers. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition, Organization and Maintenance of Knowledge. 2*, pp. 149-160. Poľana, Slovakia: SUT Press, Bratislava, Slovakia.

Tvarožek, M., Barla, M., Bielíková, M., Grlický, V., Andrejko, A., Bartalos, P., et al. (2006). Presentation and Personalization of Information in the Semantic Web. In P. Návrat, P. Bartoš, M. Bielíková, L. Hluchý, & P. Vojtáš (Ed.), *Proceedings in Informatics and Information Technologies: Tools for Acquisition, Organization and Maintenance of Knowledge* (pp. 201-207). Bystrá dolina, Nízke Tatry, Slovakia: SUT Press, Bratislava, Slovakia.

### **A.3.5 Student Research Conference Papers**

Adam, M., Barla, M., Sivák, P., & Tvarožek, M. (2005). Spot-it – Going Beyond the Vision Loss Boundaries. *Proceedings of the 3rd Student Research Competition: ACM.SRC 2005*. Prague, Czech Republic.

Adam, M., Barla, M., Sivák, P., & Tvarožek, M. (2005). Spot-it: Going Beyond the Vision Loss Boundaries. In M. Bielíková (Ed.), *Proceedings of IIT.SRC 2005: Student Research Conference* (pp. 195-200). Bratislava, Slovakia: SUT Press.

Barla, M., Bartalos, P., Porubský, J., Sivák, P., Szobi, K., & Tvarožek, M. (2006). Dynamic Ontology Based Form Generation for Portal Solutions. In M. Bielíková (Ed.), *Proceedings of IIT.SRC 2006: Student Research Conference* (pp. 113-120). Bratislava, Slovakia: SUT Press.

Tvarožek, M. (2007). Adaptive Faceted Browser for Navigation in Open Information Spaces. In M. Bielíková (Ed.), *Proceedings of IIT.SRC 2007: Student Research Conference in Informatics and Information Technologies* (pp. 164-171). Bratislava: SUT Press.

Tvarožek, M. (2008). Facilitating exploratory search by combining multiple search paradigms. In M. Bielíková (Ed.), *Proceedings of IIT.SRC 2008: Student Research*

Conference in Informatics an Information Technologies (pp. 177-184). Bratislava: SUT Press.

Tvarožek, M. (2010). Interface Generation for Semantic Web Exploration. In M. Bieliková (Ed.), *Proceedings of IIT.SRC 2010: Student Research Conference in Informatics an Information Technologies*. 1, pp. 146-153. Bratislava: SUT Press.

Tvarožek, M. (2006). Personalized Navigation in the Semantic Web. In B. Mannová, P. Šaloun, & M. Bieliková (Ed.), *Proceedings of 4th Student Research Competition: ACM.SRC 2006*, (pp. 73-80). Prague, Czech Republic.

Tvarožek, M. (2006). Personalized Navigation in the Semantic Web. In M. Bieliková (Ed.), *Proceedings of IIT.SRC 2006: Student Research Conference* (pp. 145-152). Bratislava: SUT Press.

Tvarožek, M. (2009). Toward Search and Browsing in the Social Semantic Web. In M. Bieliková (Ed.), *Proceedings of IIT.SRC 2009: Student Research Conference in Informatics an Information Technologies* (pp. 145-152). Bratislava: SUT Press.

### **A.3.6 Other**

Andrejko, A., Barla, M., Bieliková, M., & Tvarožek, M. (2006). Software tools for user characteristics acquisition. In P. Vojtáš, & T. Skopal (Ed.), *Proceedings of DATAKON '06*, (pp. 139-148). Brno, Czech Republic (in Slovak).

Šimko, J., Tvarožek, M., & Bieliková, M. (2010). Little Google Game: Term network creation via a search game. In P. Šaloun (Ed.), *DATAKON*. Mikulov, Czech Republic (in Slovak).

Tvarožek, M. (2007). Software quality attributes. In M. Bieliková, P. Návrat, M. Barla, P. Bartalos, M. Ciglan, J. Hamar, et al., *Selected studies on software and information systems*, (Vol. 3, pp. 14-24). Bratislava, Slovakia: SUT Press (in Slovak).

Tvarožek, M. (2010). Exploratory Search in the Adaptive Social Semantic Web. In M. Bieliková, & P. Návrat (Ed.), *Workshop on the Web - Science, Technologies and Engineering* (pp. 21-22). Smolenice Castle, Slovakia: SUT Press.

Tvarožek, M. (2007). Case study design. In M. Bieliková, P. Návrat, M. Barla, P. Bartalos, M. Ciglan, J. Hamar, et al., *Selected studies on software and information systems*, (Vol. 3, pp. 58-70). Bratislava, Slovakia SUT Press (in Slovak).

Tvarožek, M. (2006). Personalized navigation in an information space represented by an ontology. Master thesis, Bratislava (in Slovak).

- Tvarožek, M. (2009). Semantic Based Navigation in Open Spaces. In M. Bieliková, & P. Návrat (Eds.), *Selected studies on software and information systems*, (Vol. 4, pp. 273-302). Bratislava, Slovakia: SUT Press.
- Tvarožek, M., & Bieliková, M. (2008). Adaptive navigation, presentation and search in the large information space of the Semantic Web. In F. Babič, & J. Paralič (Ed.), *WIKT 2007: Proceedings of the 2nd Workshop on Intelligent and Knowledge oriented Technologies*, (s. 76-80). Košice, Slovakia.
- Tvarožek, M., & Bieliková, M. (2009). Integrating Web and Semantic Web browsing via a next generation browser (. In P. Návrat, & V. Vranič (Ed.), *WIKT 2008: Proceedings of the 3rd Workshop on Intelligent and Knowledge oriented Technologies* (s. 1-4). Smolenice, Slovakia: Nakladateľstvo STU.
- Tvarožek, M., & Bieliková, M. (2007). Personalized navigation for support of search and understanding of information in the Semantic Web. In M. Laclavík, I. Budinská, & L. Hluchý (Ed.), *WIKT 2006: Proceedings of the 1st Workshop on Intelligent and Knowledge oriented Technologies*, (pp. 37-40). Bratislava.

## A.4 List of citations

---

*Barla, Michal - Tvarožek, Michal - Bieliková, Mária: Rule-Based User Characteristics Acquisition from Logs with Semantics for Personalized Web-based Systems. In: Computing and Informatics. Vol. 28, No. 4 (2009), pp. 399-427*

cited in:

1. Laclavík, Michal - Šeleng, Martin - Ciglan, Marek - Hluchý, Ladislav: Ontea: Platform for Pattern Based Automated Semantic Annotation. In: *Computing and Informatics*. - ISSN 1335-9150. - Vol. 28, No. 4 (2009), pp. 555-579.
2. Vladimír Mihál, Mária Bieliková: An Approach to Annotation of Learning Texts on Programming within a Web-Based Educational System, *Semantic Media Adaptation and Personalization, International Workshop on*, pp. 99-104, 2009 *Fourth International Workshop on Semantic Media Adaptation and Personalization*, 2009.

---

*Tvarožek, Michal - Bieliková, Mária: Adaptive Faceted Browser for Navigation in Open Information Spaces. In: WWW2007, Proc. of the 16th int. Conf. on World Wide Web, Banff, Canada, 2007. New York, The Association for Computing Machinery, 2007. pp. 1311-1312*

cited in:

3. Harth, Andreas: Vol. 5690 *VisiNav: Visual Web Data Search and Navigation*. - , 2009. - ISBN 978-3-642-03572-2 In: *Lecture Notes in Computer Science*. - Berlin Heidelberg : Springer. - ISSN 0302-9743.
4. Kajaba, Michal - Návrat, Pavol - Chudá, Daniela: A Simple Personalization Layer Improving Relevancy of Web. In: *Computing and Information Systems Journal*. - ISSN 1352-9404. - Vol. 13, Issue 3 (2009), s. 29-35.

5. Vaneková, V.: Methods for user preference acquisition. In Znalosti 2008, Bratislava, ISBN 978-80-227-2827-0, pp. 387-390 (in Slovak).
6. Kajaba, Michal - Návrat, Pavol: Personalized Web Search Using Context Enhanced Query. In: Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing. CompSysTech '09, IIIA 18-1 -18-6.
7. Vallet, D. and Zaragoza, H. 2008. Inferring the most important types of a query: a semantic approach. In Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Singapore, Singapore, July 20 - 24, 2008). SIGIR '08. ACM, New York, NY, 857-858.

---

*Tvarožek, Michal - Bielíková, Mária: Personalized View-Based Search and Visualization as a Means for Deep/Semantic Web Data Access. In: WWW 2008, Proceedings of the 17th International World Wide Web Conference, Beijing, China, April 21-25. - New York : ACM Press, 2008. - ISBN 978-1-60558-085-2. - S. 1023-1024*

cited in:

8. Perugini, S.: Supporting multiple paths to objects in information hierarchies: Faceted classification, faceted search. and symbolic links. In: Information Processing and management. - Vol. 46, Issue 1 (2010), s. 22-43.

---

*Tvarožek, M. (2006). Personalized Navigation in the Semantic Web. In V. P. Wade, H. Ashman, & B. Smyth (Ed.), AH 2006: Proceedings of the 4th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems. LNCS 4018, pp. 467-472. Dublin, Ireland: Springer-Verlag, Berlin Heidelberg.*

cited in:

9. Návrat, P. (2008): Information in the semantically enriched World Wide Web. V Kelemen, Jozef (editor): Chapters on knowledge society. Iura Edition, 2008, 221-253 (in Slovak).
10. Matušíková, K., Bielíková, M. (2007): Social Navigation for Semantic Web Applications Using Space Maps. Computing and Informatics Vol. 26, No. 3, 281-299.
11. Bielíková, M., Návrat, P. (2007): Modelling, acquisition, processing and use of knowledge about user behaviour in the Internet hyperspace. Proceedings of Znalosti 2007, Ostrava, 21.-23.2.2007, 367-371 (in Slovak).

---

*Tvarožek, Michal - Bielíková, Mária: Personalized Faceted Navigation for Multimedia Collections. In: Semantic Media Adaptation and Personalization: Proc.. Second Int. Workshop SMAP 07. London, United Kingdom 2007. Los Alamitos, IEEE Computer Society, 2007. pp. 104-109*

cited in:

12. van der Sluijs, K., Houben, G.-J.: Metadata-based Access to Cultural Heritage Collections: the RHCe Use Case. In: AH 2008, 5th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Personalized Access to Cultural Heritage, Hannover, Germany, 29 July – 1 August 2008, pp. 15 - 24.

13. Clough, P. - Marlow, J. - Ireson, N.: Enabling semantic access to cultural heritage: A case study of Tate online. In: Proceedings of the ECDL 2008. Workshop on Information Access to Cultural Heritage, Aarhus, Denmark 2008, ISBN 978-90-813489-1-1.
14. Suchal, Ján - Vojtek, Peter: Navigation in the social network of the Slovak business registry. In: Datakon 2009, Proceedings of the Annual Database Conference. Srní, Czech Republic, October 10-13. - Prague: University of Economics, 2009. - ISBN 978-80-245-1568-7. - S. 145-151 (in Slovak).

---

*Tvarožek, Michal - Bielíková, Mária: Personalized Faceted Navigation in Semantically Enriched Information Spaces. In: Advances in Semantic Media Adaptation and Personalization, Vol. 2. Boca Raton : Taylor & Francis Group, 2009. pp. 181-201*

cited in:

15. Suchal, Ján - Vojtek, Peter: Navigation in the social network of the Slovak business registry. In: Datakon 2009, Proceedings of the Annual Database Conference. Srní, Czech Republic, October 10-13. - Prague: University of Economics, 2009. - ISBN 978-80-245-1568-7. - S. 145-151 (in Slovak).

---

*Andrejko, Anton – Barla, Michal – Bielíková, Mária – Tvarožek, Michal. Tools for User Characteristics Acquisition. In: Proc. of Datakon 2006, P. Vojtáš, T. Skopal (Eds.), Brno, Czech Republic, pp. 139-148.*

cited in:

16. Klempa, T.: Question Generation Based on Domain Ontology Aimed at User Model Maintenance. In: IIT.SRC 2007 – 3rd Student Research Conference, M. Bielíková (Ed.), April 2007, Bratislava, Slovakia, pp. 127 – 134.

---

*Michal Tvarožek and Mária Bielíková. Collaborative multi-paradigm exploratory search. In WebScience '08: Proceedings of the hypertext 2008 workshop on Collaboration and collective intelligence, pages 29–33, New York, NY, USA, 2008. ACM.*

cited in:

17. Perugini, S.: Supporting multiple paths to objects in information hierarchies: Faceted classification, faceted search. and symbolic links. In: Information Processing and management. - Vol. 46, Issue 1 (2010), s. 22-43.
18. Axel Rauschmayer: Connected Information Management, München, 2010-01-29, Dissertation.

---

*M. Tvarožek and M. Bielíková, Reinventing the web browser for the semantic web, in WIRSS'09: Proc. Of the WIRSS Workshop at the IEEE/WIC/ACM International Conferencie on Web Intelligence. IEEE CS, 2009, pp. 113–116.*

cited in:

19. Vladimír Mihál, Mária Bielíková: An Approach to Annotation of Learning Texts on Programming within a Web-Based Educational System, Semantic Media Adaptation



and Personalization, International Workshop on, pp. 99-104, 2009 Fourth International Workshop on Semantic Media Adaptation and Personalization, 2009.

---

*Tvarožek, Michal - Bielíková, Mária. Personalized Faceted Navigation in the semantic web. In Lecture Notes in Computer Science 4607, ICWE 2007, Baresi, L, Fraternali, P. Houben, G.J. (Eds.), Springer-Verlag, 2007, 511–515.*

cited in:

20. Kramár, T.: Mining Web Usage Patterns. In: IIT.SRC 2008, 4nd Student Research Conference, M. Bielíková (Ed.), April 2009, Bratislava, Slovakia. pp. 93 - 100.
21. Kramár, T. – Barla, M.: Dolovanie vzorov používania webového sídla. In: Znalosti 2009, P. Návrat, D. Chudá (Eds.). STU Bratislava FIIT, ISBN 978-80-227-3015-0, pp. 309-312.
22. Labaj, M. - Líška, P. - Lohnický, M. - Švoňava, D.: in FUNmation - Novel Approach to Information Presentation Employing a Game. In: IIT.SRC 2009. 5th Student Research Conference in Informatics and Information Technologies Bratislava, April 2009, Bratislava, pp. 259-266.
23. Stefaner, Moritz – Ferré, Sébastien – Perugini, Saverio – Koren, Jonathan – Zhang, Yi: Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience. In: The Information Retrieval Series, G.M. Sacco, Y. Tzitzikas (eds.), Vol. 25, Springer Berlin Heidelberg, ISSN 1387-5264, 2009, pp. 75–112.

---

*Barla, Michal – Bartalos, Peter – Bielíková, Mária – Filkorn, Roman – Tvarožek, Michal. Adaptive portal framework for Semantic Web applications. In ICWE 2007 Workshops. 7th Int. Conf. on Web Engineering, L. Bares, P. Fraternali, G.J. Houben (Eds.), Como, Italy, 2007, pp. 87-93.*

cited in:

24. Berhe, S. – Demurjian, S. – Ren, H. – Devineni, M. – Vegad, S. – Polineni, K.: Axon – An Adaptive Collaborative Web Portal. In: ICWE 2008 Workshops, 3rd Int. Workshop on Adaptation and Evolution in Web Systems Engineering – AEWSE 2008, pp. 81-87.
25. Balík, M., Jelínek, I. (2008): Experimental Adaptive Web Portal with Semantic Data Store. Semantic Media Adaptation and Personalization: Proceedings, Third International Workshop, Prague, 15-16 December 2008, IEEE CS 189-192.

---

*A. Andrejko, M. Barla, and M. Tvarožek. Comparing ontological concepts to evaluate similarity. In Tools for Acquisition, Organisation, and Presenting of Information and Knowledge: Research Project Workshop, September 2006.*

cited in:

26. Stefaner, Moritz – Ferré, Sébastien – Perugini, Saverio – Koren, Jonathan – Zhang, Yi: Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience. In: The Information Retrieval Series, G.M. Sacco, Y. Tzitzikas (eds.), Vol. 25, Springer Berlin Heidelberg, ISSN 1387-5264, 2009, pp. 75–112.

---

*Tvarožek, Michal – Barla, Michal – Bielíková, Mária. Personalized Presentation in Web-Based Information Systems. In Lecture Notes in Computer Science 4362, Sofsem 2007, Jan van Leeuwen et al. (Eds.), Springer-Verlag, 796–807.*

cited in:

27. Leal, J. P. - Silva, P.: An extensible architecture for web adaptability. In: WTI 2008, Web and Text Intelligence 08, Salvador, Bahia, Brazil, October 2008, pp. 19-26.
28. Jorge, A. - Leal, J. P. - Soares, C. - Borges, J.: Web site adaptation and automation: The site-o-matic project book. January 18, 2009.
29. Syed Toufeeq Ahmed - K. Selcuk Candan - Sangwoo Han - Yan Qi: Topic development pattern analysis-based adaptation of information spaces. In: The New Review of Hypermedia and Multimedia, Vol 15, Iss. 1 (April 2009), ISSN 1361-4568, pp. 73-96.
30. Alípio Jorge, José Paulo Leal, Carlos Soares and José Borges (Eds) Web Site Adaptation and Automation: The Site-O-Matic Project Book. 2009

---

*Bartalos, Peter – Barla, Michal – Frivolt, György – Tvarožek, Michal – Andrejko, Anton – Bielíková, Mária – Návrat, Pavol: Building an Ontological Base for Experimental Evaluation of Semantic Web Applications. In: Lecture Notes in Computer Science. Vol. 4362 SOFSEM 2007: Theory and Practice of Computer Science 2007, Springer, pp. 682-692.*

cited in:

31. Gurský, P. - Horváth, T. - Jirásek, J. - Novotný, R. - Pribolová, J. - Vaneková, V. - Vojtáš, P.: Knowledge Processing for Web Search - an Integrated Model and Experiments. In: Scalable Computing: Practice and Experience. - ISSN 1895-1767. - Vol. 9, Nr. 1, Special Issue: Distributed Intelligent Systems (2008), pp. 51-59.

## Appendix B Evaluation Environment

---

A major part of our research was performed as part of several research projects conducted at the Institute of Informatics and Software Engineering, Slovak University of Technology. In order to evaluate our approach, we devised, developed and used two distinct prototypes of our faceted browser *Factic* and the associated back-end services within the scope of these projects.

Our first prototype was developed as part of projects NAZOU<sup>21</sup> and MAPEKUS<sup>22</sup>, where it was a key integration part of the devised evaluation environment and thus also used services provided by other parts of the entire evaluation framework. The second *Factic* prototype was developed separately to evaluate methods devised as part of later projects, e.g., PeWePro<sup>23</sup>. The second prototype also had a major integrating role in conjunction with other exploratory approaches (mostly in collaboration with bachelor and master students), while also taking advantage of the experience gained from the first prototype by addressing some of its shortcomings.

This appendix provides a brief overview of the used evaluation environments and developed prototypes, partly adapted from individual project documentations.

### B.1 First Factic prototype (NAZOU, MAPEKUS)

The evaluation environment in projects NAZOU and MAPEKUS was built around a personalized presentation layer architecture, which as the primary means of user interaction, integrated presentation tools with user modelling and personalization tools, and also provided an interface to information organization tools in the application layer. As such the evaluation environment consisted of a web portal, implemented in the open-source Apache Cocoon web framework<sup>24</sup>, which integrated other presented tools as plug-ins in a generic way via XML/XSLT transformations of data and code invocation via Java reflection.

I participated mainly in the overall design of the architectural solution and the realization of the faceted browser *Factic* and the back-end logging service *SemanticLog*.

#### B.1.1 Personalized Presentation Layer Architecture

Many contemporary information systems employ a standard three-layer architecture consisting of a data layer, an application layer and a presentation layer. In this context, the

---

<sup>21</sup> Project NAZOU: <http://nazou.fiit.stuba.sk>

<sup>22</sup> Project MAPEKUS: <http://mapekus.fiit.stuba.sk>

<sup>23</sup> Project PeWePro: <http://pewepro.fiit.stuba.sk>

<sup>24</sup> Apache Cocoon Project: <http://cocoon.apache.org/>

data layer stores and retrieves data from a database, the application layer performs the core business logic of the system while the presentation layer takes care of the presentation and user interaction. Our personalized presentation layer extends the traditional presentation layer of the typical three-layer architecture with additional personalized features aimed at Semantic Web applications.

Consequently, we think of the presentation layer as a personalized presentation layer that performs three primary tasks:

- It provides a user interface that offers simple access to all of the system's functionality while effectively hiding all of its inner complexity from the user.
- It dynamically adapts the user interface to the needs, usage patterns and goals of individual users in order to increase their comfort, productivity and satisfaction by exploiting information stored in a user model.
- It creates a comprehensive log of user activity, evaluate it and extract and store meaningful user characteristics, which will then be used in the adaptation process.

As such, the aforementioned tasks are performed by the presentation part, the personalization part and the user modelling sub-layer of the system respectively. Since these tasks depend on each other, our design employs an integrated set of cooperating software components (tools) to realize the necessary functionality. Furthermore, the personalized presentation layer interacts with the system via software agents (tools) in the application layer that either provide or evaluate data (see Figure 1).

Presentation tools and the portal work with the domain and user models and consist of a presentation and a personalization layer respectively. Individual presentation tools are depicted in the centre and forward output to the web portal, which in turn provides an interface to the client web browser (right). The user modelling layer is shown at the bottom and includes both client and server side logging and user characteristic evaluation. All of the already described functionality can be examined at the following two levels of abstraction.

The primary purpose of the portal level is to act as means of integration for the system's functionality by providing a common environment for individual tools. The portal provides a common global navigation interface, authentication, authorization and user management services to individual tools (e.g., information about the current user session). It defines the overall high-level functionality and layout of items such as global menus, fields, links and that of individual tools, as well as the overall navigation structure of the whole site. With respect to personalization, the layout of individual tools can be adapted by changing the order of the displayed portlets, by adaptively adding or hiding items in global menus or by adding or hiding portlets. Furthermore, the portal provides a "skinnable" interface, which allows users to choose their preferred presentation style.

The purpose of individual tools is to realize or provide access to the functionality of the system. Presentation tools are responsible for the presentation of information and user interaction with each tool being responsible for the handling of its part of the user – system interaction. Thus the tool level defines the internal low-level functionality and

layout of items (e.g., controls, menus, text areas, images) for individual presentation tools and the functionality of processing tools which are not directly used for presentation (e.g., for user modelling or business logic). Individual tools provide tool-specific adaptation of visualization and functionality based on the common user model inferred from implicit user feedback throughout the system via both client-side and server-side logging agents.

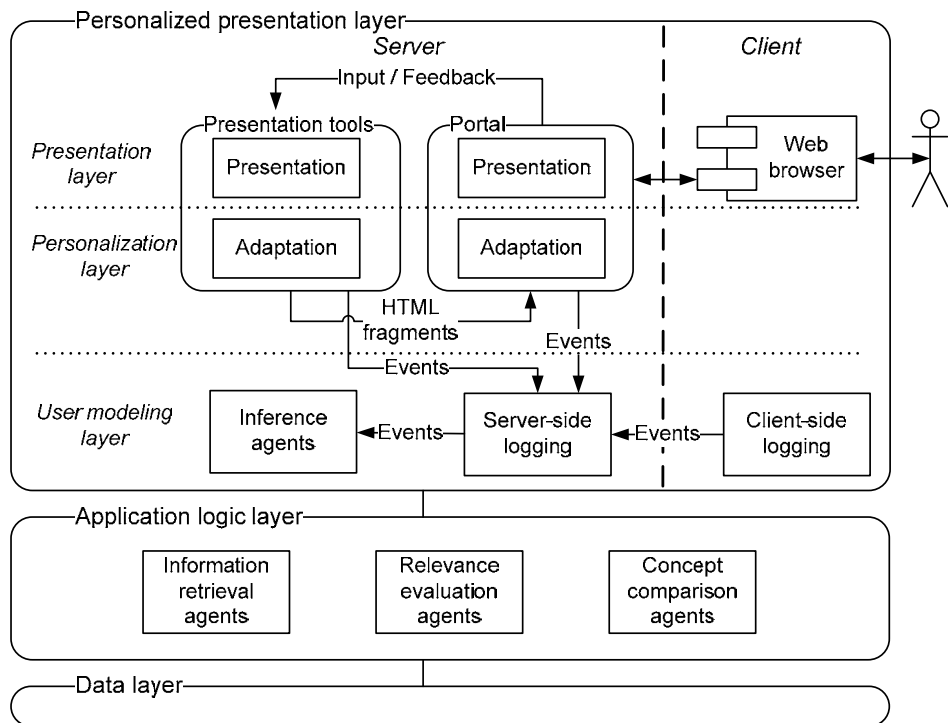


Figure 1. Architecture of the personalized presentation layer in NAZOU and MAPEKUS.

### B.1.2 Practical Architecture Implementation

We used a single tool for portal level functionality – *JOP* – Job Offer Portal which is the primary user interface and processes all user inputs while returning the corresponding outputs. Both *JOP* and the underlying tools employ the Sesame 1.2 ontological repository to store data, but also use the MySQL relational database to store simple table-based relational data during pre-processing.

The main purpose of *JOP* is to integrate different functionality and provide a form-based user interface tailored for ontology editing with support for dynamic form generation for a given ontology. *JOP* also allows users to register, log in and customize the global layout of the user interface. Since *JOP* is based on Apache Cocoon which supports

flexible inclusion of existing functionality into the portal solution, the individual functionality and results of all other tools are easily integrated into the user interface provided by *JOP*. Each tool (or set of tools) is represented by a coplet (cocoon portlet) whose position within the overall layout of the portal can be adjusted by the user, who can also minimize, maximize or hide it completely.

On the tool level we developed two cooperating presentation tools, with each tool being responsible for the adaptation of its content according to data from user model. We also developed several simpler tools for the editing of the user profile and for forms filling, both of which are integrated into our portal solution. *Factic* – Faceted Semantic Browser is the main presentation tool that allows users to navigate the information space by choosing restrictions on the displayed content. It is fully integrated into *JOP*, which supplies it with information about the current user for adaptation purposes. As input it takes user actions and returns the logical description of the content that should be displayed. Its output can be further processed by a set of XSL transformations to directly create valid XHTML output or alternatively it can be sent to Prescott for further processing.

*Prescott* is a presentation tool able to visualize domain dependent content (e.g., job offers) in a flexible and configurable manner using the Fresnel presentation ontology. It defines various views on domain content using “lenses”, which can be defined dynamically based on user preferences. As input, Prescott takes the logical description of the content (e.g., the URIs of job offer instances) and returns an XHTML fragment with the visualization of the content. Additionally, Prescott can take advantage of user characteristics stored in a user model to choose the most appropriate lens to apply on the ontology individuals that should be displayed. Prescott is invoked mainly by *Factic* every time the user changes the selected restrictions or decides to view details of a job offer.

To fulfil the requirements on the data collection stage, we developed the client side monitoring tool *Click*. This JavaScript based tool captures and logs browser events and sends them to the server. Server side logging is enhanced by the *SemanticLog* tool, which combines information from presentation tools and logs acquired by *Click* to create a comprehensive log of user actions with added semantics which are suitable for further processing. The consecutive data processing is performed by the *LogAnalyzer* tool, which estimates user characteristics and stores them in the user model. Since the used method implies the mainly domain-dependent nature of the revealed characteristics, *LogAnalyzer* needs better “understanding” of the displayed domain content and uses the services provided by the *ConCom* tool, which compares ontological concepts by using various comparison strategies.

We developed the above mentioned presentation and user modelling tools and integrated them into the Job offer portal thus successfully using the proposed personalized layer architecture design. To verify the integration aspects of the proposed architecture we integrated additional tools that, for example, acquire and present job offer clusters from the application layer or display personalized instance ratings. *Factic* employs the *CriteriaSearch* and *JDBSearch* tools for result ordering, advanced search in the data collection and similar offer search, for which also *ConCom* is used. *Factic* was also

integrated with the *TopK* tool, which provides personalized search results relevance evaluation and ordering based on user preferences acquired either via *Factic* and *SemanticLog* itself or via the *UPreA-TopK-IGAP* user modelling chain. Additional tools use *Factic* to show their results, such as *Ontocase*, *Ontea* and *ClusterNavigator*. The integration and interaction of individual tools during request processing is shown in Figure 2, the invocation of application-layer tools from *Factic* is hidden.

Client requests are first received by *JOP* which forwards them to appropriate presentation tools or processes them directly in which case it next sends a response to the client. The *Factic* presentation tool prepares the response by querying the domain and user models and invoking Prescott to generate XHTML fragments as necessary. It then logs the respective event and resulting display state via the *SemanticLog* web service. Lastly it combines individual XHTML fragments into the final response and sends it to *JOP*, which in turn forwards it to the client.

Furthermore, *JOP* and *Click* independently log events that resulted from user actions via *SemanticLog*, which notifies *LogAnalyzer* about new events for processing. *LogAnalyzer* in turn processes these events asynchronously taking advantage of *ConCom* and updates the user model with newly identified user characteristics.

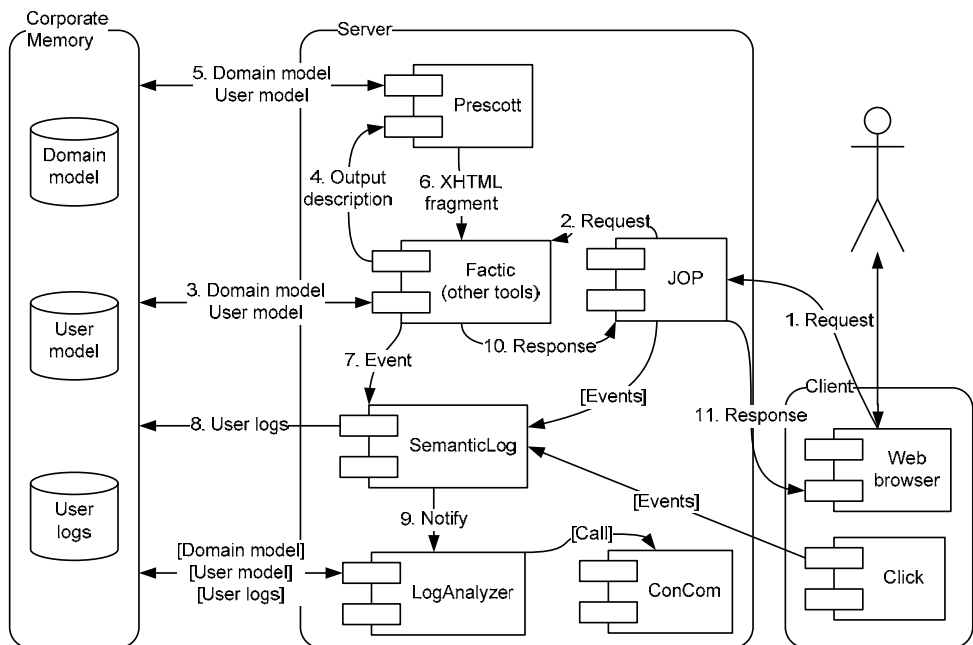


Figure 2. Request processing of our personalized presentation evaluation framework.

The user characteristics are used by several software tools from the application layer which contribute to further refinement of user interests. The *Top-k* aggregator tool retrieves the most relevant job offers with respect to user preferences (e.g., salary, education

requirements, place) based on ordered lists of user preferences. The Aspect tool searches for similar documents (job offers) based on a probabilistic model for soft clustering. Using the described approach for the comparison of domain and user ontology instances the devised clusters are presented to the user based on her characteristics.

### B.1.3 Factic implementation and operation

We designed and implemented the adaptive faceted semantic browser as a software tool called *Factic* and integrated it into the personalized presentation layer of a web-based information portal (see Figure 1).

The *Factic* presentation tool is depicted on the top left and can be divided into the presentation and adaptation parts. As input, Factic takes user input/feedback from the Portal module, to which it also sends the results of its processing in the form of (X)HTML fragments. The portal serves for the integration of individual presentation tools (e.g., Factic) and acts as a proxy towards the client web browser depicted on the right.

Presentation tools as well as the Portal tool perform user action logging with semantics by means of the user modelling layer depicted at the bottom, which performs both client-side and server-side logging and user characteristic analysis.

Factic is implemented in Java as an Apache Cocoon coplet (i.e., Cocoon portlet) and uses Sesame to store ontological data, MySQL to store relational data and as a back-end for Sesame. For the *SemanticLog* logging service we use Apache Axis as a web service container and Apache Tomcat as a servlet container.

Since Cocoon is based on XML and the pipes and filters architectural pattern where every request is processed by a given pipeline, *Factic* itself was implemented as a Cocoon pipeline generator thus taking full advantage of its XML processing capabilities. Figure 3 depicts the design and request processing of Factic, which employs a two-step transform view, where the initial logical XML output description is transformed by a set of XSL transformations into the final XHTML document (top) and sent to the client web browser (right). Individual user requests are handled as described by Sequence 1 and Figure 5. The resulting rendered user interface is shown in Figure 4.

Sequence 1: HandleRequest

Input: URL request

Output: XHTML response

1. Session: Preprocess request, update session state
2. Core: Process request, create and execute query
3. DataProviders: Retrieve domain and user data
4. Core: Process results, evaluate adaptation and annotation
5. Session: Log event via the SemanticLog logging service
6. Generator: Generate logical output description in XML
7. Cocoon: Transform XML output to formatted XHTML response



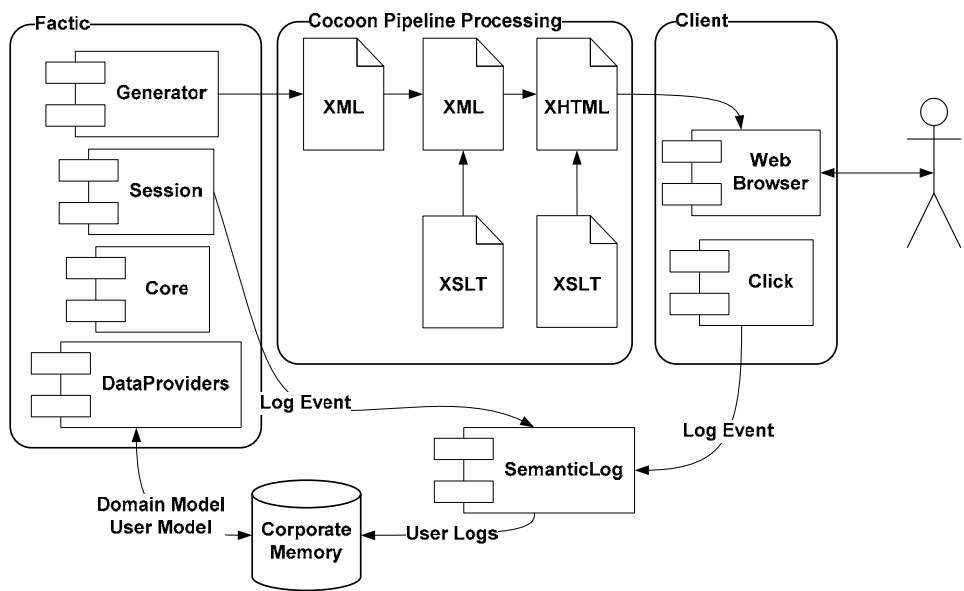


Figure 3. Design of the Factic presentation tool. Both Factic and Cocoon operate on the server side of the system. Factic generates the initial data for the processing pipeline while Cocoon performs the successive XML transformations.

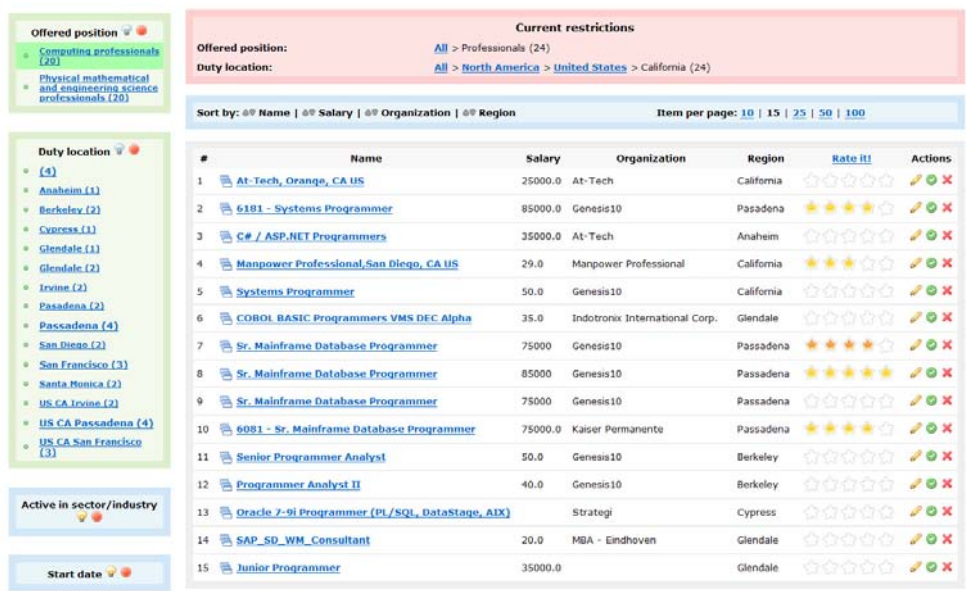


Figure 4. Example of our first Factic prototype showing facet personalization (left), list view search results (centre) with query-by-example via rating and similarity search (right).

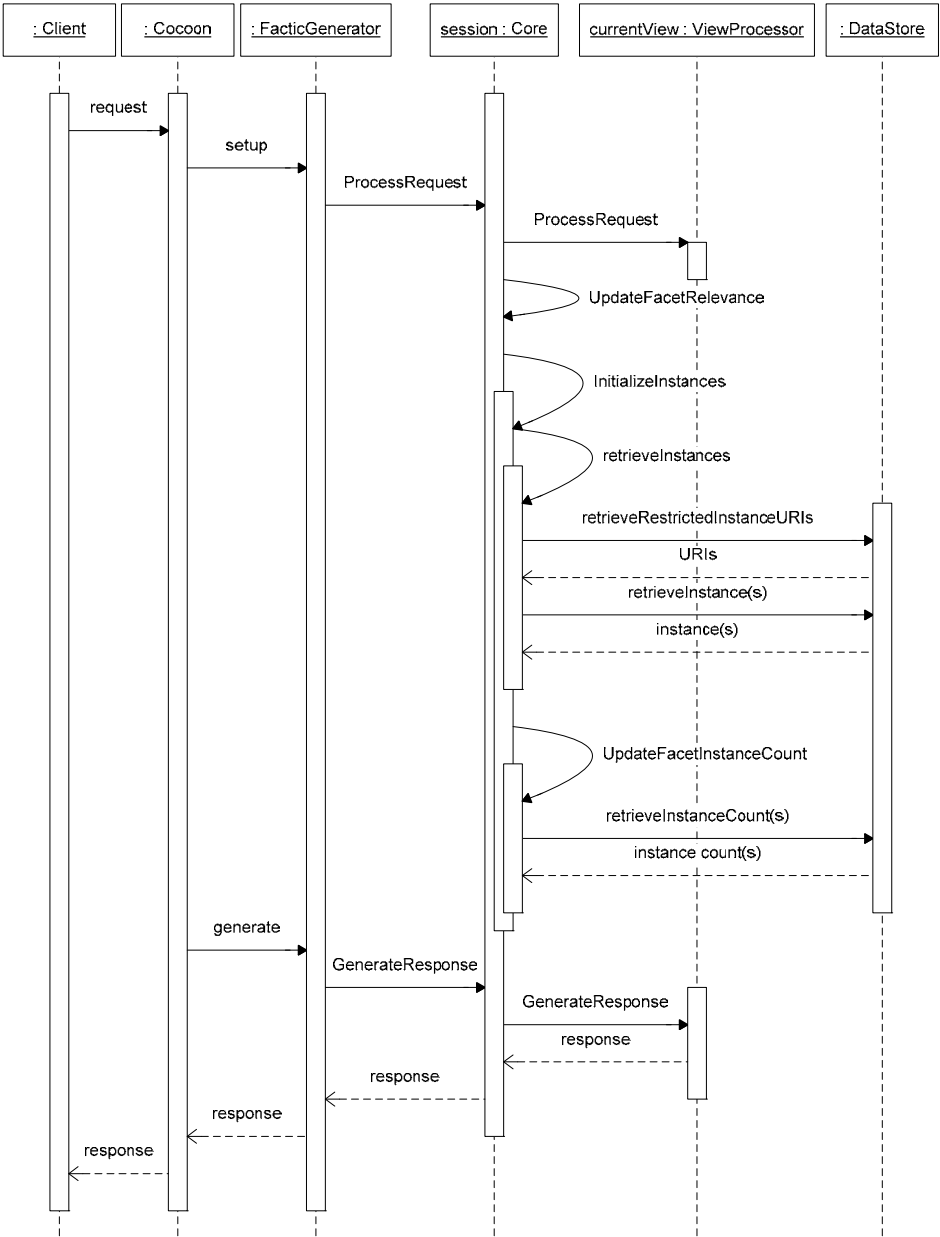


Figure 5. Typical request processing performed by Factic.

## B.2 Second faceted browser prototype

We used our first prototype primarily to evaluate our personalization approach in the domain of job offers and scientific publications, where we achieved encouraging results. Still, we encountered several issues during the implementation and tuning of our prototype as well as severe bottlenecks with scalability and practicality:

- The use of the Apache Cocoon framework proved to be a mistake, as the development and debugging of applications in this open-source Java framework was atrocious. The framework itself contained several critical bugs, was missing vital functionality (e.g., user management). There was only limited if any support with limited or to-be-done documentation.
- The performance of the Apache Cocoon framework with the XML/XSLT pipeline architecture was rather slow. Java itself was lacking in memory management efficiency and did not offer a 64bit development / runtime environment thus lacking sufficient memory for caching.
- The integration of individual interface widgets in Apache Cocoon was difficult and communication was not efficient. Support for asynchronous operation was limited thus user interaction always resulted in slow full page requests. Additionally client-server roundtrip latency made some operations unnecessarily long although they could have been completed quickly on the client side.
- The logging of user actions together with the display state of the GUI via web services was very slow because a lot of data had to be serialized/deserialized via SOAP. We originally solved this by using a hybrid logging approach where some of the data are logged via web services (e.g., client-side logging) and some data are logged directly by means of an API (e.g., display state of the GUI). Still, the logging of the entire view state was not practical.
- The cost of ontological queries was high and consequently, the processing of ontological queries was slow. We were unable to resolve this problem with the first prototype although we somewhat improved overall performance by caching data. Furthermore, the ontological repository Sesame 1.2 was rather immature (i.e., slow, unoptimized and contained bugs that prevented correct evaluation of queries).
- SeRQL, the recommended query language for Sesame, and thus Sesame lacked several important features such as COUNT() or ORDER BY. These must thus be emulated by our application which further reduces performance. There was also no SPARQL support at the time.

In order to build on our experience with the first *Factic* prototype and address the aforementioned issues, some of them quite serious, we developed a second prototype as a client-side Silverlight C#/ .NET application running inside a web browser in the digital image domain. This simplified deployment enabled us to process and store

information on the client device. The second *Factic* prototype is primarily centred around end-user specific functionality such as visualization, personalization, user modelling and profile management. We also process end-user specific data (e.g., activity logs and the derived user models) on the client thus reducing unwanted privacy exposure (with optional sharing with the server-side).

Consequently, *Factic* works as an intelligent front-end to (multiple) search and/or information providers thus effectively delegating querying, indexing and crawling services to third-party providers (e.g., in the future Google, Sindice, DBpedia). Our modular architecture allows us to easily incorporate new views and also gives us the flexibility to add new data sources provided that they contain enough semantic metadata to generate user interface widgets (see Fig. 6).

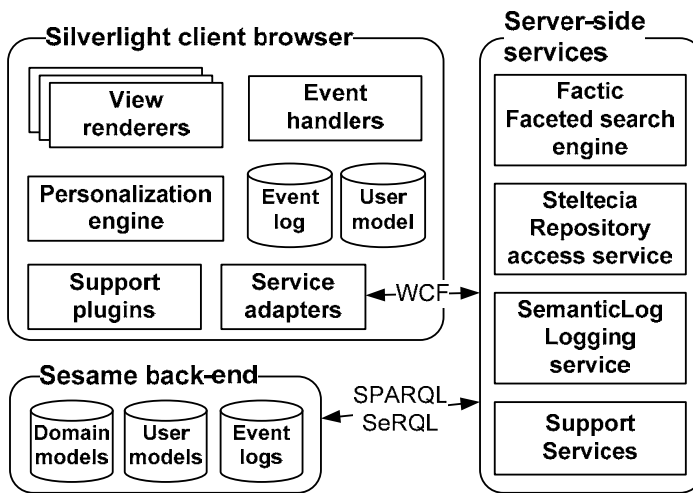


Figure 6. The client renders the user interface and provides personalization support (top). The server includes web (WCF) services for faceted search (*Factic*), ontological repository access (*Steltecia*), and event logging (*SemanticLog*) for global statistics tracking (right). All services store their data in a common ontological repository in *Sesame*.

The Silverlight client browser handles all user interaction and acts as a front-end to server-side web (WCF) services, which serve as search providers, content providers or as support services. These include the *Factic* faceted search engine, the *Steltecia* service for ontological repository access, and the optional *SemanticLog* event logging service for global statistics tracking along with additional support services for individual plug-ins, e.g. for creating thumbnails of web resources or providing full text search.

While the Silverlight client runs in a typical web browser (e.g., IE or Firefox), the server-side services are all deployed in the IIS 7 web server as WCF services. As a database back-end we employ *Sesame* 2.0, which improved over *Sesame* 1.2 in various ways, e.g. by improving performance and SPARQL support. While the *Sesame* repository is still deployed in the Apache Tomcat application server, we also employ the MS SQL database

to store temporary data (e.g., for pre-processing) and Apache Lucene to provide indexing and full-text search capability.

The request processing of our second *Factic* prototype is fully asynchronous, where user actions invoke asynchronous requests to server-side services. The client application is still responsive to user input while only relevant information is transferred and the respective user interface widgets are updated, which significantly improved perceived response times from an end-user perspective.

We use our second *Factic* prototype mainly to evaluate our facet generation and multi-paradigm exploration approaches. We thus integrated the faceted browser with additional widgets for view rendering, orientation support and exploration (see Fig. 7):

- *Photo view*, which provides an end-user grade visualization experience for photos by supporting common features from popular web-based photo galleries (e.g., Flickr or Picassa), such as image slide shows, zooming, rotation.
- *Graph view*, which provides a graph-based visualization of the information space and allows users to explore the resources and their relations by zooming, panning or expanding the nodes in the graph view.
- *Search history tree*, which provides orientation support during search and browsing sessions by showing what queries have been performed the user in the past, what results were visited and by providing quick links to return to any of these previous browsing states.
- *Annotation view*, which enables (authorized) users to edit existing or create new content in the respective information space. This includes the annotation of individual images but also the editing or creation of other resources such as authors, events associated with images.



Figure 7. Example of the user interface of our second *Factic* prototype in Silverlight. Facets shown in the centre, search history tree on the left, result matrix on the right.



## Appendix C      Ontological Models and Datasets

---

This appendix provides an overview of the used data models and data sets adapted from individual projects documentations in project NAZOU (job offers), MAPEKUS (scientific publications) and digital images.

### C.1      Project NAZOU: Job Offers Domain Model

The domain ontology defines an explicit conceptualization of a job offer and its related concepts (prefix *jo:*) represented in OWL format. It also references or extends other domain independent ontologies:

- The *region* ontology (prefix *r:*) describes regions, countries, languages and currencies, which are used in the respective regions;
- The *classification* ontology (prefix *c:*) describes hierarchical classifications of industrial sectors, professions, education, qualifications and various orderings;
- The *offer* ontology (prefix *ofr:*) describes a generic offer, which is represented by the class *ofr:Offer*. Each offer has a source (*ofr:OfferSource*), a validity interval (*ofr:ValidityInterval*) and an identification of the tool/method of its acquisition and insertion into the ontology (*ofr:OfferCreator*). Fig. 1 shows the relations between these classes.

The domain ontology itself consists of about 700 classes of which 670 belong to hierarchical classifications with a maximum depth of 6 levels. The ontology contains a test base of about 1,000 job offer instances acquired by different means prior to system evaluation (e.g., manual input or web wrappers). Due to the overall number of classes and schema instances, we only describe here a set of the main selected classes.

#### Class JobOffer

The primary class of the domain ontology is the *JobOffer* class, whose instances represent a single job offer (Fig. 2). The *JobOffer* class has these data properties:

- *jo:hoursPerWeek* – the number of work hours per week as *xsd:float*;
- *ofr:name* – the title of the offers as *xsd:string*;
- *jo:startDate* – date when the contract should start as *xsd:date*;
- *jo:startDateASAP* – *xsd:boolean* indicating immediate start of work;
- *jo:subordinateCount* – the number of subordinate employees as *xsd:int*.

Object properties associated the *JobOffer* class with these additional domain concepts:

- *jo:ApplyInformation* – information about application submission;

- *jo:Benefit* – benefits offered for the position (car, flat, insurance, etc.);
- *jo:ContractType* – the contract type (permanent, temporary, contract);
- *jo:JobTerm* – the job term required (full-time, half-time, etc.);
- *c:ManagementLevel* – describes the level of management required for the position with respect to the overall corporate management structure;
- *jo:Organization* – description of the organization offering the position;
- *jo:Prerequisite* – prerequisites that a successful candidate must meet;
- *c:ProfessionClassification* – classification of the position based on the United Nations profession classification;
- *r:Region* – location where the position is offered;
- *jo:Responsibility* – describes the responsibilities the candidate will have;
- *jo:Salary* – describes the offered salary;
- *jo:TravelingInvolved* – indicates whether the position requires travelling.

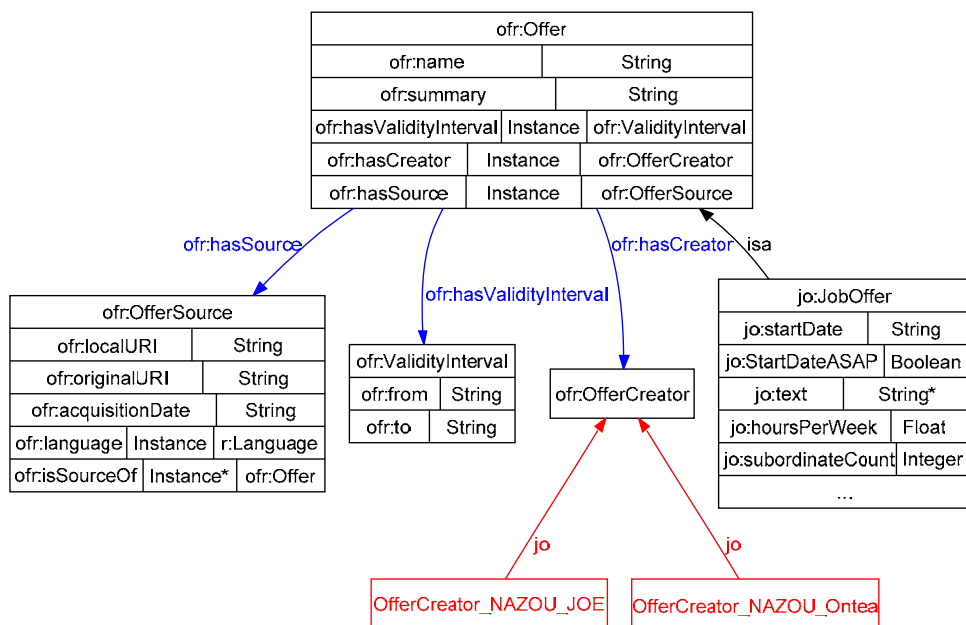


Figure 1: Relations between the Offer, OfferSource and JobOffer classes.

## Class Prerequisite

The *Prerequisite* class models requirements that applicants of a job offer must meet. Requirements are divided into:

- *required* – applicants must satisfy these requirements to be accepted;
- *preferred* – applicants satisfying these requirements will be preferred.



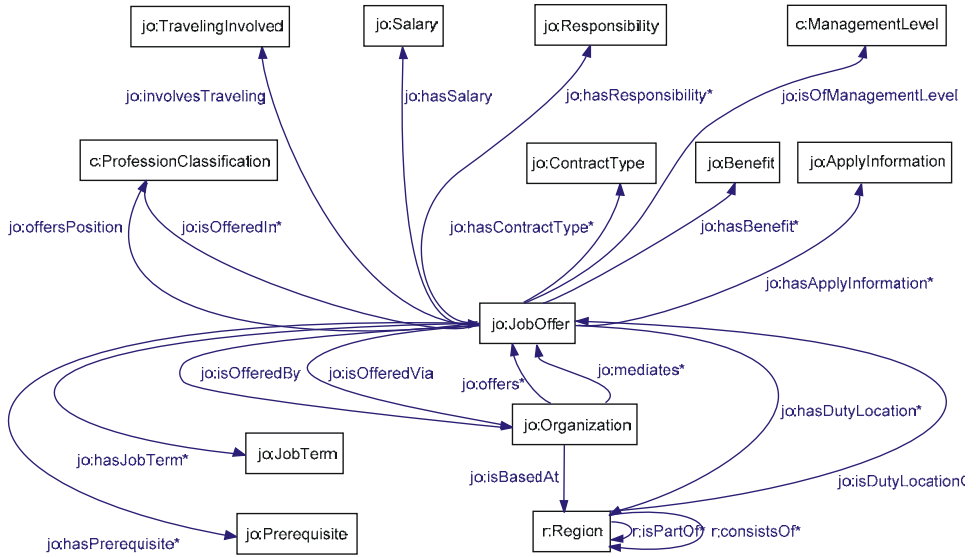


Figure 2: Object properties of the *JobOffer* class.

Employer requirements associate a *JobOffer* instance with classes in the *Classification* ontology and can be of three types:

- Requirements on the education of the candidate – *QualificationClassification*;
- Requirements on the experience of the candidate – *ExperienceClassification*;
- Requirements associated with the candidate – *PersonalAttributeClassification*.

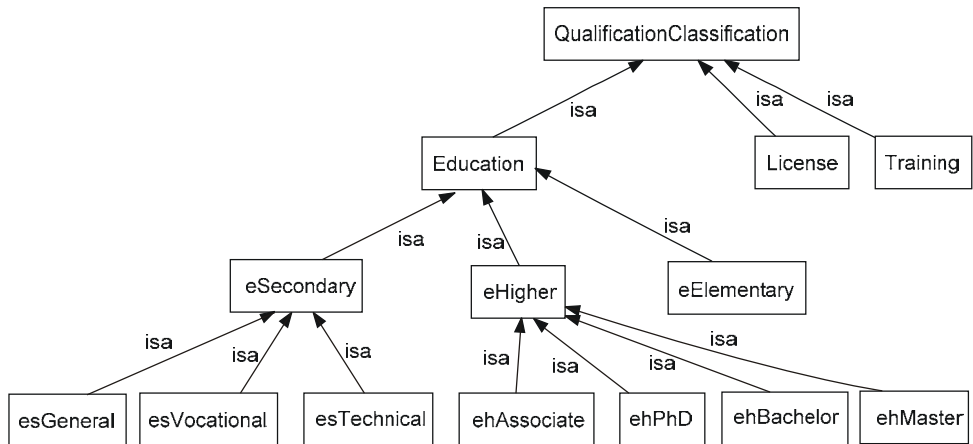


Fig. 3 Qualification classification describing educational requirements.

### Class QualificationClassification

The qualification classification in Fig. 3 is comprised of the classification of *Education*, *License* and *Training* requirements. Education describes the type and level of education of the candidate, License describes the authorization or licenses the candidate has, Training describes the courses or training the candidate has.

### Class ExperienceClassification

The *ExperienceClassification* class models the requirements on prior candidate work experience. Each experience has a specific level (*hasLevel*) and may be expressed by a value with an associated measure (e.g., the number of kilometres driven per year). Experience can also be described with respect to a specific knowledge domain (e.g., study programme), to a skill, a sector (e.g., academic or private), to an industry sector or a profession (Fig. 4).

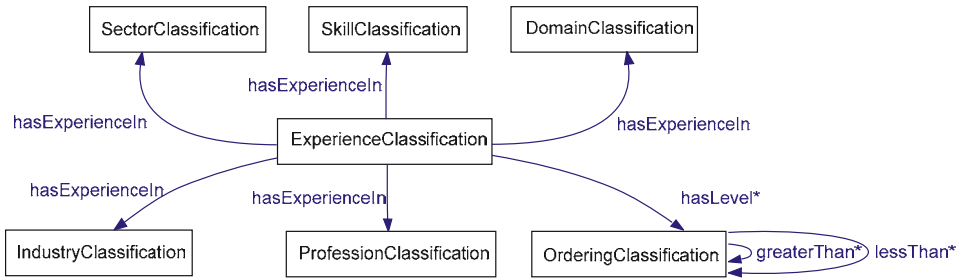


Figure 4: Relations of the experience classification with respect to other classifications.

### Class PersonalAttributeClassification

The classification of personal attributes models the requirements which are related to personal prerequisites or properties of individual job candidates such as analytical thinking, communication skills, perfect eyesight.

## C.2 Project MAPEKUS: Scientific Publications Domain Model

The entities and relations in the publications domain are conceptualized via the publication ontology, represented in the OWL language. The ontology itself was created based on an in-depth analysis of the scientific publications domain and populated with instances using several digital libraries serving as data sources (ACM DL, DBLP and SpringerLink). We collected information about publications, authors, organizations, keywords, references and citations (Table 1). The quality of data varied between the sources and also overall due to duplicate, erroneous or incomplete data, which had to be filtered and merged in the pre-processing stage.

Table 1. Overview of the scope of the data acquired from individual sources.

Resource \ Data source	ACM	DBLP	Springer
Instance	1,805,369	1,680,875	102,410
Object properties	10,815,691	10,684,530	153,360
Data properties	2,197,522	3,656,164	201,910
Authors	708,475	612,581	57,504
Organizations	6,793	-	6,232
Publications	899,402	1,062,210	35,442
Keywords	106,176	-	-

The domain ontology of publications takes advantage of domain independent ontologies:

- The *region* ontology (prefix *r:*) describes regions, countries, languages and currencies, which are used in the respective regions;
- The *party* ontology (prefix *p:*) describes stakeholders in relations, such people or organizations.

The main classes of the publication ontology are (Fig. 5):

- *Publication* – corresponds to a generic publication, its subclasses are individual publication type, e.g. books, articles or proceedings;
- *Person* – describes authors and editors of publications;
- *Event* – describes conferences and other meetings related to publications;
- *Organization* – includes universities, research labs and publishers;
- *IndexTerm* – represents the topic of a publication, based on the hierarchical taxonomy of terms provided in the ACM digital library.

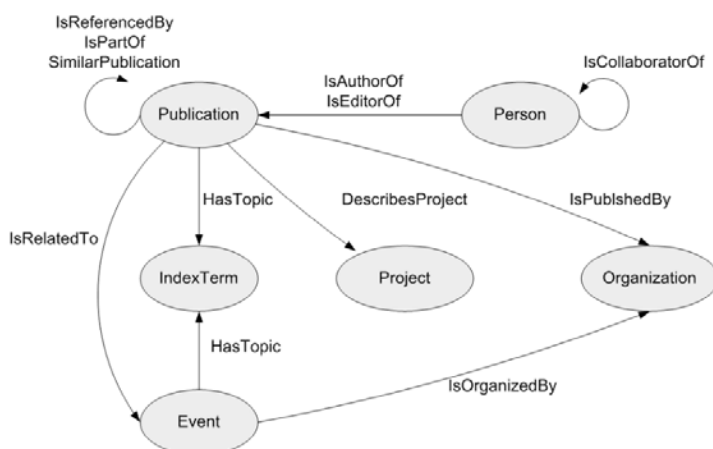


Figure 5: Relations between the main classes of the publication ontology.

## Class Publication

The Publication class is the main class of the ontology and has these subclasses that correspond to specific kinds of publications (Fig. 6):

- *Article* – a class corresponding to a journal article;
- *Book* – a class corresponding to a book with these subclasses:
  - *Anthology* – a collection of works by various authors,
  - *Biography* – a biographical work,
  - *Monograph* – a scholarly work on a single topic;
- *Journal* – corresponds to a periodic publication for scholarly readership;
- *Paper* – corresponds to scholarly work with these subclasses:
  - *ConferencePaper* – a conference paper,
  - *TechnicalPaper* – a technical paper describing, e.g. a developed system;
- *Proceedings* – a set of scientific articles published, e.g. at a conference or other gathering, usually as a book;
- *Poster* – corresponds to a poster;
- *TechnicalReport* – a formal report detailing the contribution in applied science and development, describing the details and achieved results in a problem area.
- *Thesis* – a work describing the results of research performed by a student with these subclasses:
  - *MasterThesis* – work realized as part of the second level university study,
  - *PhDThesis* – work realized as part of the third level university study.

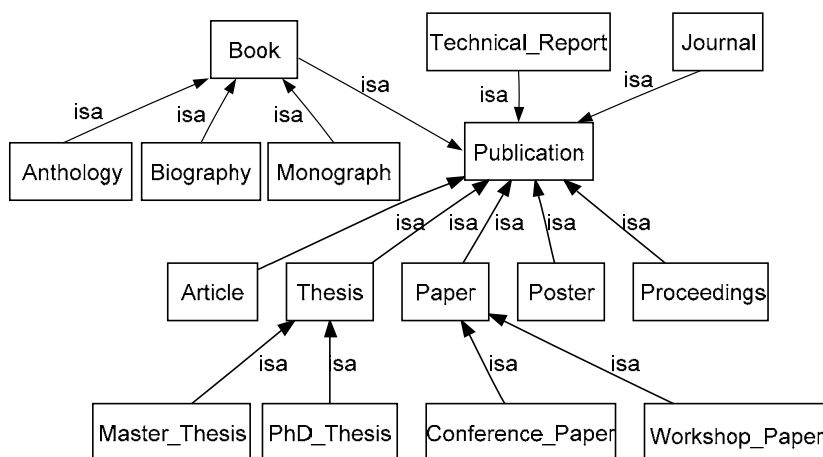


Figure 6: Hierarchical taxonomy of publication types.

Figure 7 depicts data and object properties of the *Publication* class. Data properties are:

- *pages* – type *xsd:string* – the number of pages on which the publication is if it is part of another publication (e.g., conference proceedings);
- *date* – type *xsd:string* – publication date;
- *abstract* – type *xsd:string* – abstract describing the publication's content;
- *firstPage* – type *xsd:int* – the number of the first page if it the publication is part of another publication;
- *year* – type *xsd:int* – the year of publication;
- *source* – type *xsd:string* – the original source of the publication (e.g. its DOI);
- *web* – type *xsd:string* – a link pointing to the resource on the web;
- *title* – type *xsd:string* – the title of the publication.
- *lastPage* – type *xsd:int* – the number of the last page if it the publication is part of another publication.

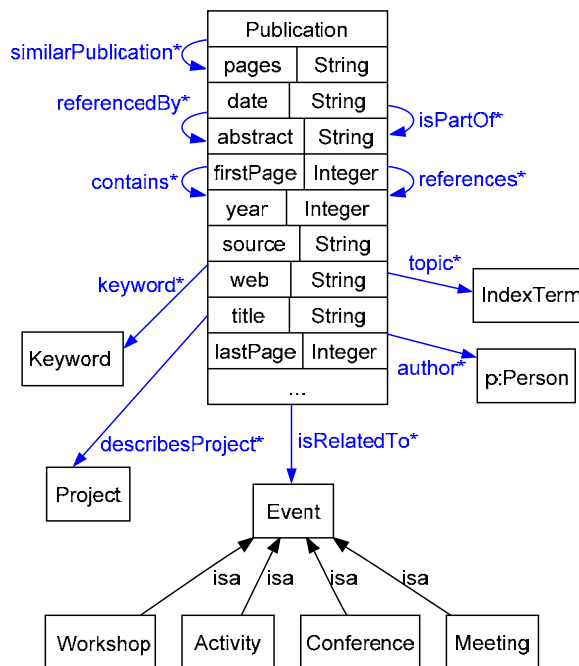


Figure 7: Properties of the Publication class.

Object properties are:

- *keyword* – class *Keyword* – describes a keyword;
- *describesProject* – class *Project* – describes a research/development project;
- *isRelatedTo* – class *Event* – an event which is related to a publication (e.g., a conference);

- *author* – class *Person* – the author of a publication;
- *topic* – class *IndexTerm* – an indexed term describing the publication;
- *contains* – class *Publication* – links to publications that are parts of the parent publication (e.g., articles in a journal);
- *isPartOf* – class *Publication* – links to parent publication (inverse to contains);
- *references* – class *Publication* – links to referenced publications;
- *referencedBy* – class *Publication* – links to citing publications (inverse to references);
- *similarPublication* – class *Publication* – links to similar publications.

### Class Event

The Event class represents events that occur in some community, e.g. conferences, workshops, meetings (Fig. 8). The Event class has these properties:

- *startDate*, *xsd:date* denoting the date when the event starts;
- *endDate*, *xsd:date* denoting the date when the event ends;
- *web*, *xsd:string* linking the original web resources about the event.

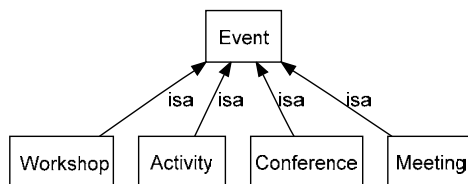


Figure 8: The Event class and its subclasses.

### Class IndexTerm

The *IndexTerm* class and its subclasses correspond to the taxonomy of research areas in computer and information science taken from the ACM digital library. Fig 9. shows the hierarchical tree structure of the subclasses with the leaves corresponding not to classes but to individual instances.

### Class Author

The *Author* class is derived from the *Person* class (from the *Party* ontology) and represents people who have authored at least one publication. It has these derived properties:

- *givenName*, an *xsd:string* representing the given name of a person;
- *familyName*, an *xsd:string* representing the family name of a person.

## Class Editor

The *Editor* class represents editors of at least on publication. Similarly as the *Author* class, it is derived from the imported class *Person* and thus has the same properties.

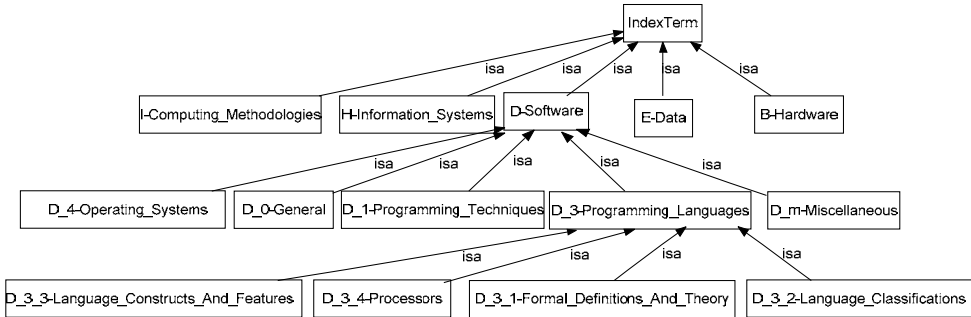


Figure 9: Hierarchy of *IndexTerms*, for reasons of clarity, only one branch is shown at each level.

## Class Publisher

The *Publisher* class represents the publishers of at least one publication. It is derived from the *Organization* class imported from the *Party* ontology.

## C.3 Projects NAZOU, MAPEKUS: User Model

The ontological user model is derived from the ontological domain model by overlaying it with user specific preferences. The user model itself has two layers, the basic domain independent layer, which can be shared between application domains, and the domain specific layer which must be customized for each domain. Our user model ontology separates the domain independent and domain specific characteristics into three subontologies as used in projects NAZOU and MAPEKUS:

- *generic-user* ontology – defines general user characteristics;
- *job-offer-user* ontology – defines characteristics associated with the domain of job offers represented by the job offer domain ontology;
- *publication-user* ontology – defines characteristics associated with the domain of publications represented by the publications domain ontology.

### C.3.1 Domain independent model

#### User class

*User*, the primary class of the user model has these data and object properties (Fig.10):

- *hasMaxAge*, *hasMinAge* – data property of type *xsd:int* represents upper and lower boundary of interval which contains user's age;
- *hasChild* – data property of type *xsd:boolean* has the value true or false depending on whether a user has at least one child or not;
- *livesInRegionOfSize* – data property of type *xsd:int* represents number of citizen in user's region of residence;
- *hasCharacteristic* – object property represents domain independent characteristics, its range are instances of type *UserCharacteristic*;
- *includes* – object property with a range instances of type *DomainSpecificUser* represents domain-specific parts of user model.

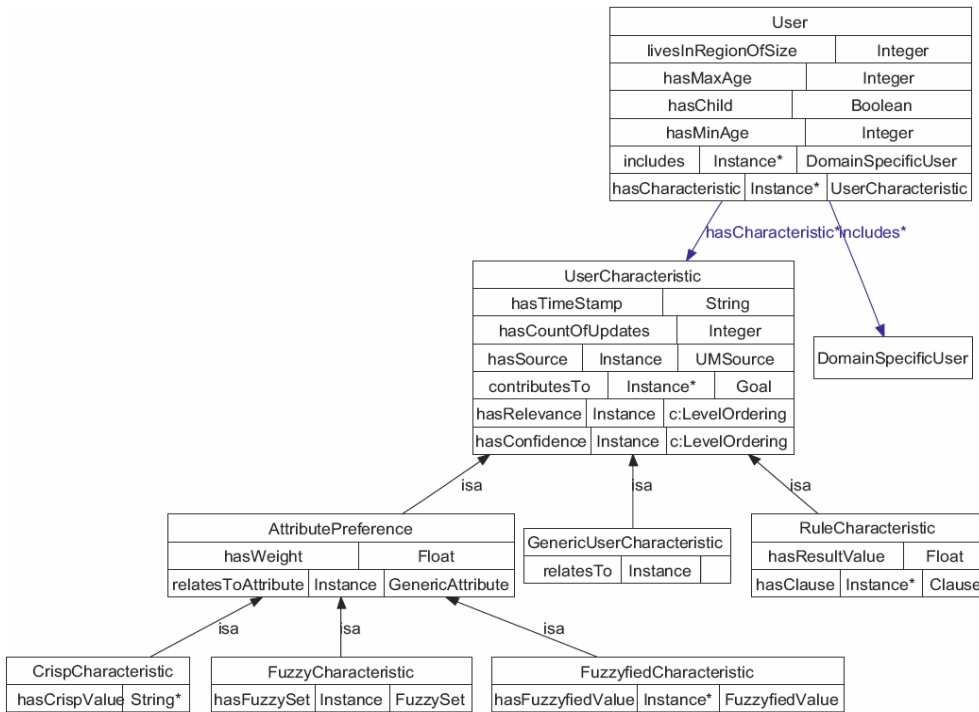


Figure 10: Domain-independent user model.

## UserCharacteristic class

Class *UserCharacteristic* has these properties:

- *hasTimeStamp* – data property of type *xsd:string* represents a time stamp of this characteristic;



- *hasCountOfUpdates* – data property of type *xsd:int* represents number of actualization (updates) of this characteristic;
- *hasSource* – object property with a range instances of type *UMSource* represents characteristic's source;
- *contributesTo* – object property with a range instances of type *Goal* represents a goal, which relates to this characteristic;
- *hasRelevance* – object property represents relevance of this characteristic in order to achieve *Goal* linked by *contributesTo* property; its range instances are of type *c:LevelOrdering*;
- *hasConfidence* – object property represents confidence of this characteristic (degree of quality of user characteristic estimation); its range instances are of type *c:LevelOrdering*.

Class *UserCharacteristic* has these subclasses:

- *AttributePreference* – represents local preferences. One subclass of this class is used for an individual characteristic. It can be *FuzzyCharacteristic*, *CrispCharacteristic* or *FuzzyfiedCharacteristic*.
- *GenericUserCharacteristic* – represents characteristic in general, which can be used to express relationship to any entity of user and domain model.
- *RuleCharacteristic* – represents global preferences in the form of rules, e.g.: *resultValue = 0.5 IF (goodSalary >= 0.7 AND goodPosition >= 0.4)*.

## UMSource class

Class *UMSource* does not have any data or object properties. It is assumed, that such instances exist in domain-specific parts of a model that represent ways how the user model gets populated with data (automatically by software tools and manually from human intervention).

## Goal class

Class *Goal* does not have any data or object properties. It is assumed that such instances exist in domain-specific parts of a model that represents user goals in the particular domain.

## AttributePreference class

Class *AttributePreference* has the following properties:

- *hasWeight* – data property of type *xsd:float*. It is used to compute weighted average if no rules are available;

- *relatesToAttribute* – object property represents an attribute which is bound to the characteristic. Its range instances are of type *GenericAttribute*.

### GenericUserCharacteristic class

Class *GenericUserCharacteristic* has these properties:

- *relatesTo* – object property with a range instances of any class from user and domain model.

### RuleCharacteristic class

Class *RuleCharacteristic* (Fig. 11) has these properties:

- *hasResultValue* – data property of type *xsd:float* represents the resulting value of the rule;
- *hasClause* – object property with a range instances of type *Clause* represents individual clauses of the rule.

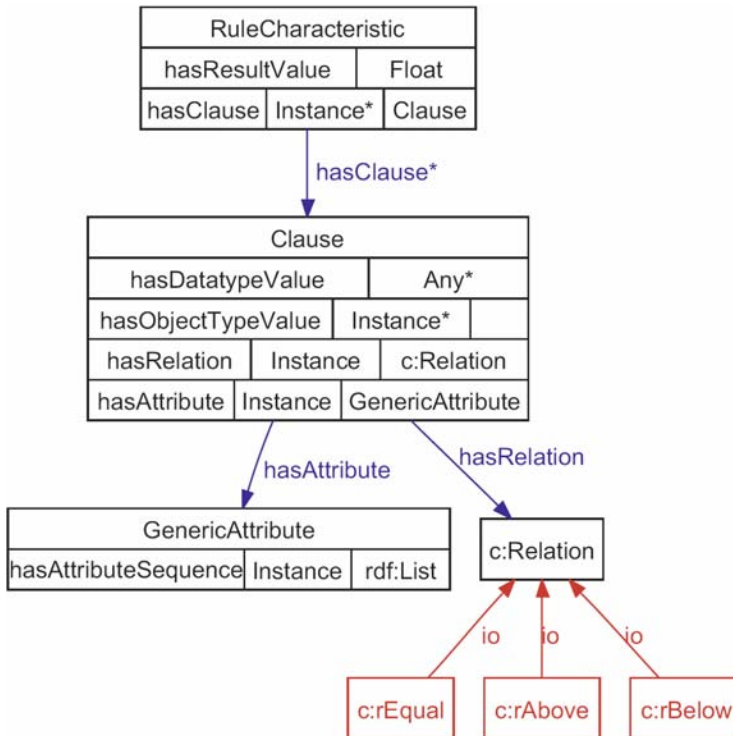


Figure 11: Part of a model representing a Rule Characteristic

### Clause class

Class *Clause* represents a clause part of the rule – a condition applied on a particular attribute. It has the following properties:

- *hasAttribute* – object property represents an attribute on to which the property is related. It has range instances of type *GenericAttribute*;
- *hasDataTypeValue*/*hasObjectTypeValue* – data/object property, which represents a value of respective attribute;
- *hasRelation* – object property with a range instances of type *c:Relation* represents a type of relation (<, >, =).

### GenericAttribute class

Class *GenericAttribute* represents properties of any object. It is required to create a subclass along with instances in a domain specific part of the model. The class has the following properties:

- *hasAttributeSequence* – object property with a *rdf:List* as a type of range instances, which represents a list of properties from a base class of domain specific model (i.e., a Publication in MAPEKUS project).

### CrispCharacteristic class

Class *CrispCharacteristic* holds evidence of tolerable values of properties, if these can not be ordered naturally (e.g., places or company names). It has these properties:

- *hasCrispValue* – data property of type *xsd:string* represents a list of values, which are of user's interest.

### FuzzyCharacteristic class

Class *FuzzyCharacteristic* represents a fuzzy characteristic for properties, whose values can be naturally ordered (e.g., a salary, received degree). It has these properties (Fig. 12):

- *hasFuzzySet* – object property with a range instances of type *FuzzySet* represents fuzzy set, which assigns a number  $R, R \in [0, 1]$  to each value. Higher the number is, better the value is for the user.

### FuzzySet class

The *FuzzySet* class is defined by its member function. This function assign a number from  $[0, 1]$  to each item. If an item is assigned 0, it does not belong to the set. If an item is assigned 1, it belongs to the set. Values from open interval (0,1) are interpreted as a partial membership in the set. The shape of the set is acquired by linking all its points. Class *FuzzySet* has these properties:

- *hasPoint* – object property with a range instances of type *FuzzySetPoint* represents individual points of a fuzzy set;
- *hasType* – object property with a range instances of type *FuzzySetType* represents one of four set types.

### FuzzySetPoint class

Class *FuzzySetPoint* has these properties:

- *hasY* – data property of type *xsd:float* represents rating from 0 to 1;
- *hasX* – data property of type *xsd:float* represents numerical value of an object;
- *hasXString* – data property of type *xsd:string* represents a label of value.
- *FuzzySet* point has coordinates x, y and a label. If we want to plot a fuzzy set, we plot *hasXString* values on x-axis. If the items of a set are number, than the labels are their textual representations (x=1, xstring="1"). If the items of a set are values such as Bc., Mgr. Phd., the value *hasX* would contain symbolic numerical values (xstring = "Bc.", x = 1; xstring = "Mgr.", x = 2).

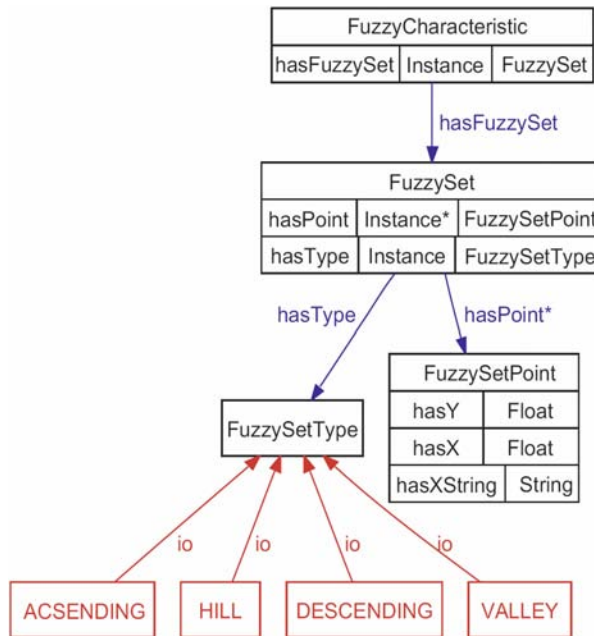


Figure 12: Representation of fuzzy characteristics.

### FuzzySetType class

Class *FuzzySetType* represents types of fuzzy sets, which expresses the shape of member function. The class has these instances:

- *ASCENDING* – the member function is ascending;
- *DESCENDING* – the member function is descending;
- *HILL* – the member function has a triangle- or trapezoid-like shape;
- *VALLEY* – inverse of *HILL*, it reaches its maximum at the edges of function's domain.

### FuzzyfiedCharacteristic class

Despite of the fact, that values of a particular properties cannot be naturally ordered (places, company names, etc.), we can assign a 0-to-1 preference to some values. It is not a fuzzy set, as we do not have an x-axis, so we do not know the order. However, we can still use it when selecting the best objects. Class *FuzzyfiedCharacteristic* has these properties:

- *hasFuzzyfiedValue* – object property represents values of properties along with their numerical rating. Its range instances are of type *Fuzzyfied-Value*

### FuzzyfiedValue class

Class *FuzzyfiedValue* has the following properties:

- *hasString* – data property of type *xsd:string* represents values of properties, which cannot be ordered naturally;
- *hasEval* – data property of type *xsd:float* taking the value from 0 to 1, which express to what extent the user prefers this value.

## C.3.2 Domain-specific model

In general, there could be several domain-specific models that would be connected with a single domain-independent model. In the NAZOU and MAPEKUS projects, we devised two domain specific models for each respective domain.

The primary class of the domain specific model in project NAZOU is *JobOfferSpecificUser*, while in project MAPEKUS it is *PublicationSpecificUser*. Both are subclasses of the *DomainSpecificUser* class from the domain independent user model thus facilitating the interconnection of the respective models.

### JobOfferSpecificUser class

Class *JobOfferSpecificUser* has these properties (Fig. 13):

- *hasCharacteristic* – object property with a range instances from a union of classes *RuleCharacteristic* and *AttributePreference*.
- *hasVisitedOffer* – object property represents records about visited offers. Its range instances are of type *VisitedOffer*.

### VisitedOffer class

Class *VisitedOffer* defines a record about interaction of user with domain content. It has these properties:

- *hasDateOfVisit* – data property of type *xsd:string* represents a date, when the interaction occurred.
- *hasJobOffer* – object property with a range instances of type *jo:JobOffer* represents a domain content (a job offer);
- *hasRating* – data property of type *xsd:int* represents user's rating of the offer.

### JobOfferAttribute class

Class *JobOfferAttribute* is a subclass of a *GenericAttribute* class and replaces it in the domain of job offers.

### PublicationSpecificUser class

Class *PublicationSpecificUser* has these properties (Fig. 14):

- *hasCharacteristic* – object property with a range instances from a union of classes *RuleCharacteristic* and *AttributePreference*.
- *hasVisitedPublication* – object property represents records about visited publications. Its range instances are of type *VisitedPublication*.

### VisitedPublication class

Class *VisitedPublication* defines a record about interaction of user with domain content. It has the following properties:

- *hasDateOfVisit* – data property of type *xsd:string* represents a date, when the interaction occurred.
- *hasPublication* – object property with a range instances of type *p:Publication* represents a domain content (a publication);
- *hasRating* – data property of type *xsd:int* represents user's rating of the publication.

### PublicationAttribute class

Class *PublicationAttribute* is a subclass of a *GenericAttribute* class and replaces it in the domain of publications.

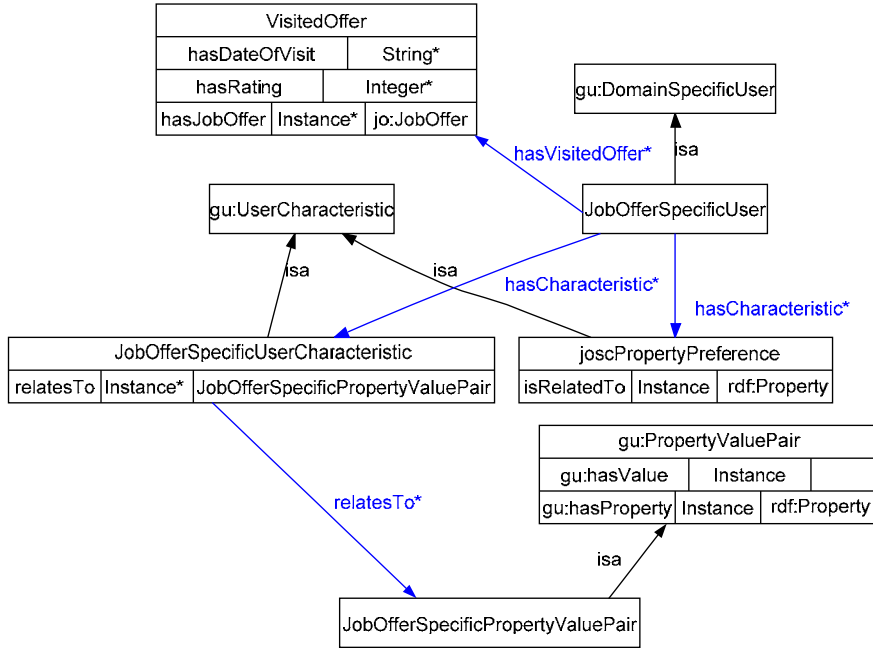


Figure 13: Domain specific user model in the job offers domain.

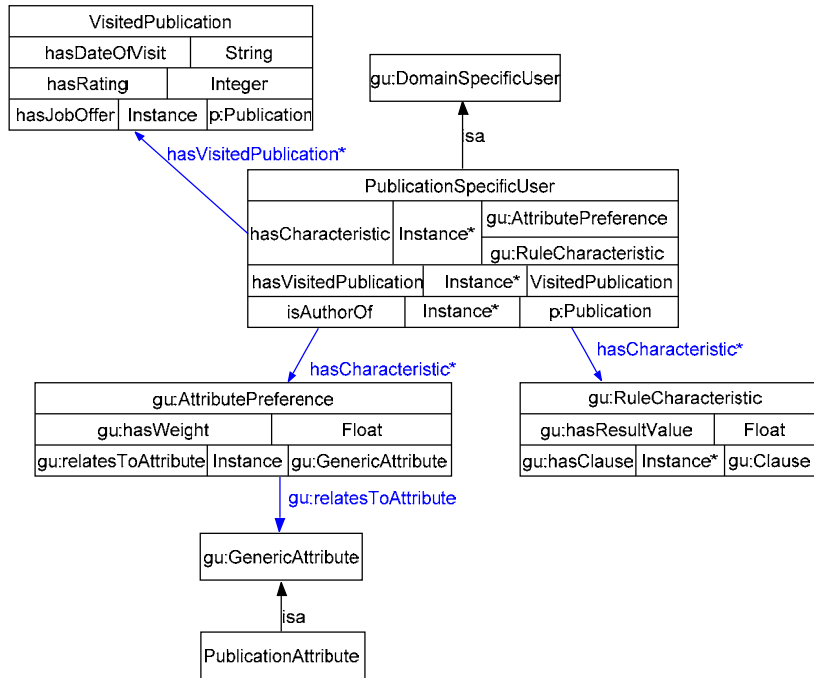


Figure 14: Domain specific user model in the publications domain.

## C.4 Project NAZOU, MAPEKUS: User Log Model

In order to store events that happen in the browser at run time, we employ an event ontology stored in a relational database. Figure 15 shows the SQL data model used to store the ontology for further processing by user modelling tools.

Individual events are stored in the *events* table which corresponds to the *Event* class in the ontology. Subsequently, each event has an associated *type* (table *typesOfEvents*), *user* (table *users*) and *session* (table *sessions*) in which the event took place. Each event is also associated with two view states corresponding to the presented view before and after the event occurred (table *displayStates*).

The view states are used to determine user characteristics based on implicit user feedback. Each view state describes the presented widgets (table *displayedItems*), their types (table *typesOfDisplayedItems*), attributes (table *displayedItemAttributes*) and types of attributes (table *typesOfDisplayedItemAttributes*). We also record the attributes of individual events (table *eventAttributes*) and their types (table *typesOfEventAttributes*).

User IDs correspond to user logins, URIs are used for identification of concepts and instances from the domain ontology with a maximum length of 100 characters.

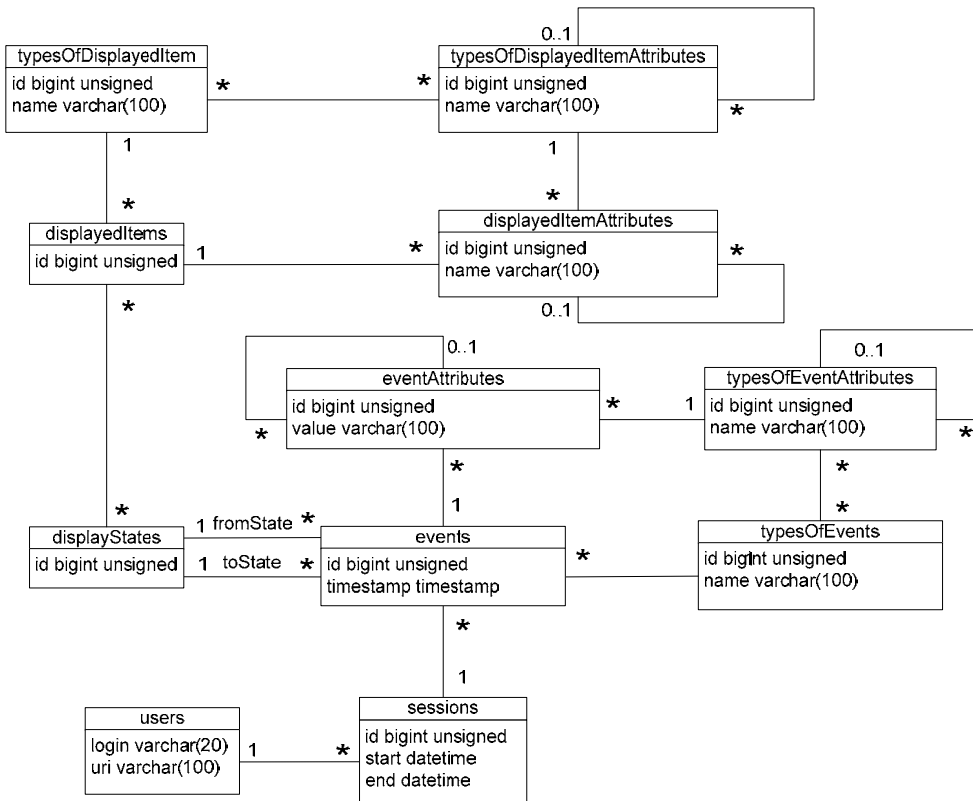


Figure 15. The data model used to store the event ontology used to represent user actions.



## C.5 Digital Image Domain Model

The domain ontology of digital images was created by analyzing existing photo galleries creating a corresponding conceptualization of the information space. To represent EXIF metadata, which are automatically added to images in cameras, we build on the popular Kanzaki EXIF ontology<sup>25</sup>, import additional domain independent ontologies and extend it with additional concepts to represent image specific metadata.

The domain ontology of images imports these domain independent ontologies:

- The *region* ontology (prefix *r:*) describes regions, countries, languages and currencies, which are used in the respective regions;
- The *party* ontology (prefix *p:*) describes stakeholders in relations, such people or organizations.

We populated the ontology with metadata about 8,000 photographs provided by various authors from our university in the form of image files in three steps:

- The files were manually organized into several sets of directories corresponding to some metadata, e.g. years, events or places in directory names.
- The whole 30 GB data set was then automatically imported into our ontology and the corresponding thumbnails were created.
- Selected images in the data set were manually annotated by users supplying information such as what is on the photo, what the weather was or what type of photo it was (e.g., cloudy weather, landscape, city scene).

The entire ontology consists of 35 classes, 50 properties (including relations and attributes), more than 32,000 individuals and in excess of 150,000 facts. For individual photos, the ontology describes EXIF metadata as supplied by the camera, information about formats in which the photos are available (e.g., resolution, aspect ratios), and optional additional annotations such as the author, the object and background of the photo, the place, overall theme and expression, lighting conditions, weather and the event to which the photo belongs (see Fig. 16).

---

<sup>25</sup> Kanzaki EXIF ontology, <http://www.kanzaki.com/ns/exif>

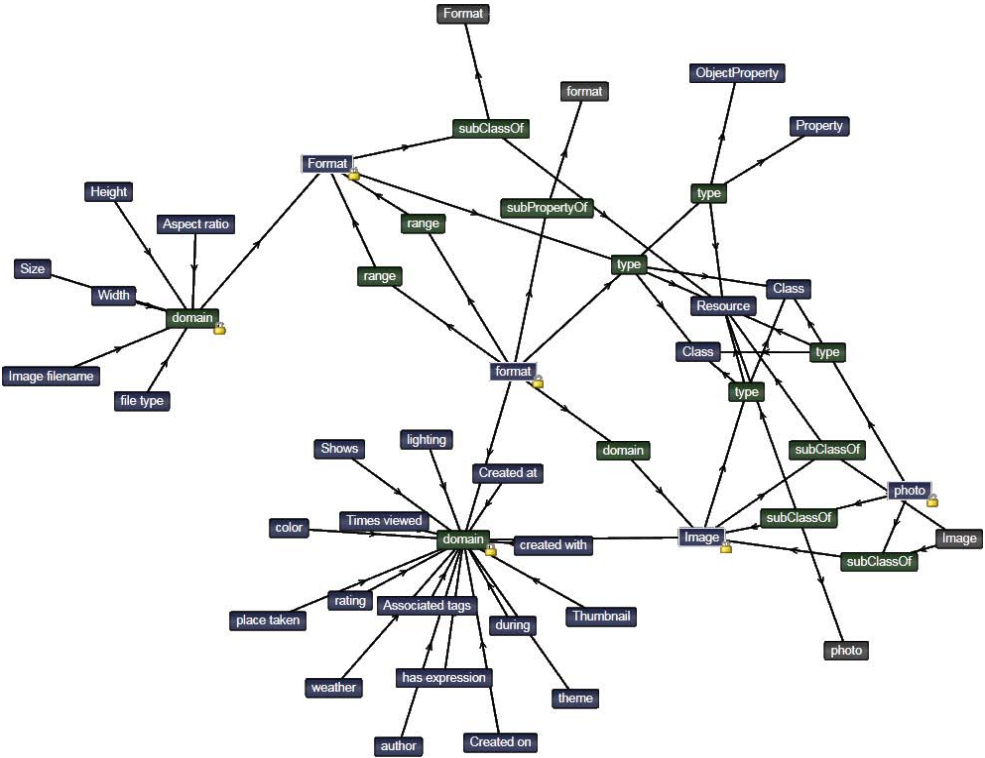


Figure 16. Visualization of a part of the digital image domain via our graph view. The main Image class (bottom right) has several properties associated with it such as author, shows or theme (bottom left).