Towards Computerized Adaptive Assessment Based on Structured Tasks

Jozef Tvarožek[1], Miloš Kravčík[2], and Mária Bieliková[1]

[1] Faculty of Informatics and Information Technologies,
Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava, Slovakia
{jtvarozek,bielik}@fiit.stuba.sk
[2] Open University of The Netherlands,
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Milos.Kravcik@ou.nl

Abstract. In an attempt to support traditional classroom assessment processes with fully computerized methods, we have developed a method for adaptive assessment suitable for well structured domains with high emphasis on problem solving and capable of robust continuous assessment, potentially encouraging student's achievements, reflective thinking, and creativity. The method selects problems according to the student's demonstrated ability, structured task description schemes allow for a detailed analysis of student's errors, and on-demand generation of task instances facilitates independent student work. We evaluated the proposed method using a software system we had developed in the domain of middle school mathematics.

1 Introduction

Most classroom assessments today are carried out using traditional paper & pencil methods. Paper as a delivery medium allows the students to elaborate and justify their answers in a very liberal way. While linear and adaptive computerized tests are widely used in web-based education [1] and testing community [2], they do not provide sufficient freedom of expression required to assess student's progress in solution paths of the problems and thus are not a viable option for classroom assessment. Although not primarily designed for assessment, using an intelligent tutoring system (ITS) brings some hope. ITS obviously gives the student more expressiveness during the interaction on problems but since the authoring process is time-consuming and requires sophisticated analysis [3], the system usually contains only a limited set of questions.

Recently, the Assistments system [4] seems as a more suitable alternative for classroom use. Being a so-called pseudo-tutor, a simplification of the original ITS concept, the system provides a practice environment for students giving them the opportunity to learn while solving problems and reporting their progress on a scale representing a nation-wide test performance. The problems, also called assistments, are organized in sections and within a section the assistments are optionally presented in linear or random order. A single assistment is a tree

of scaffolding questions branched from the top-level question. While providing accurate predictions for the nation-wide MCAS test, the Assistments system in its present state does not account for issues with exposure of assistments, personalized sequencing, and open-ended student answers and therefore is usable only as an instructional assistance as originally intended.

In this paper, we present a novel method for adaptive assessment which has been proposed as a part of a broader effort to bring classroom assessment to its full potential by computerized methods utilizing adaptivity, suitable answer interfaces, automatic task generation, and collaborative approaches. The method is appropriate for well structured domains with high emphasis on problem solving such as middle school mathematics, high school and university level programming, data structures and algorithms courses.

The system we had developed based on the proposed method incorporates four major aspects we argue are important in any robust assessment system:

1. for an assessment task to identify the student's solution path,
2. personalized sequencing of tasks during examination,
3. suitable answer interfaces depending on the task type,
4. on-demand generation of new tasks.

Our assessment tasks are structured in the form of a tree comparable to the structure of an assistment. A node in the tree represents a solution path; multiple branches at a node can be defined modeling a possible error in the student's solution at the respective granularity. Tasks are described in schemes using a high-level object language facilitating on-demand task generation and effective judging of open-ended answers. Schemes are calibrated using a psychometric *Item Response Theory* (IRT) [5] model, and a standard *Computer Adaptive Testing* (CAT) [6] algorithm for adaptive selection is employed.

In the next section, we provide an overview of research on related problems. We describe the proposed method for adaptive assessment in detail in section 3. In the evaluation, in section 4, we explore the feasibility of the judging process, demonstrate the adaptive selection, and summarize the students' attitudes towards the assessment in the domain of middle school mathematics. Summarizing thoughts and proposals for future work are to be found in section 5.

2 Related Work

Based on an extensive survey of the research literature on assessment, the article [7] concluded that innovations which include strengthening the practice of formative assessment (evaluation carried out in the course of an activity in such a way that the information obtained is used to improve learning and/or instruction) produce significant, and often substantial, learning gains. The formative assessment experiments produce typical effect sizes between 0.4 and 0.7. Such effect sizes are larger than most of those found for educational interventions.

One of the assessment environments used today is SIETTE [8], a web-based tool in which teachers define tests, and students can take these tests on-line.

SIETTE uses traditional multiple-choice questions while custom item formats can be implemented by a Java applet. To further enhance the system, possibilities of adding instructional support by adaptive hints are explored in [9].

Automatic item generation. Item pools in CATs need regular refreshing because even with a relatively few items compromised a substantial gain can be achieved [6]. Methods of automatic item generation are explored to lessen the costs of creating new items [10]. Items are usually generated from so-called item models, prototypes, or schemes, by instantiating parameters with random values. IRT parameters of the generated instances may be slightly different but provided that we preserve the item structure and calibrate the instances together as a single item no statistically significant differences in ability estimates have been observed [11, 12]. A more sophisticated method, generating math word problems using frame semantics, is explored in [13].

Adaptive item selection. Selecting the next item in an adaptive test is a nontrivial task. The number of times an item is administered might differ significantly between items if we choose to select the most informative item only [6]. The often administered tasks are easily disclosed and may compromise the whole adaptive test. Therefore, methods for controlling the exposure of items, limiting items' usage, are employed. Normally, the simple method of randomly selecting one of the k most informative items is used. The sophisticated b-blocking-a-stratified method [14] stratifies available items into layers according to the discrimination parameter b. Balanced exposure is ensured by selecting less discriminating items early in the examination when the estimate is still inaccurate and using high discriminating items later when we need to pinpoint the estimate in a relatively narrow ability range.

3 Method for Adaptive Assessment

Let us describe the main parts of our assessment system which is divided up into several independent modules (see Figure 1). A task conceived by an expert is processed into a parametric task description scheme described using a high-level object library. Tasks are parametrized to provide sufficient abstraction for the generator module to create new task instances on-demand.

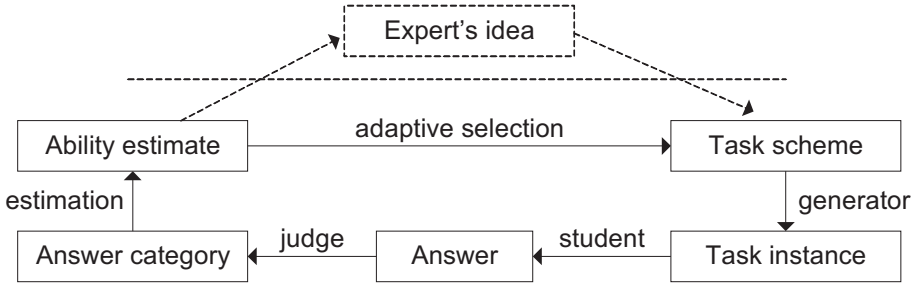

Fig. 1. Assessment system architecture.

During the assessment phase, the system selects the task scheme that provides the most information for the current student's ability estimate. Using the selected task scheme, an unused task instance is generated and displayed to the student for answering. Students' answers are semi-automatically assigned to predefined categories. Having determined the category, the system either (1) asks the student a deeper question regarding her solution path, or (2) finishes the instance administration providing the task outcome which is subsequently used to update the ability estimate. Depending on the amount of error in the updated estimate, the selection module either (i) selects another task scheme at an appropriate level of ability to continue with, or (ii) finishes the assessment process providing the final ability estimate together with the amount of error.

Finally, estimates are transformed into grading levels required by the institution and the students are allowed to assert tasks' difficulties, confront their answers with the correct ones, and compare with their peers. Raw answers are further analyzed by domain experts to extract new patterns and solution paths not previously anticipated and to increase automatic judge efficiency.

3.1 Task Descriptions

Tasks are described in schemes consisting of:

1. *Static descriptions created during the authoring phase* - encompasses types and ranges of scheme parameters, set of constraints, and display templates of descriptions of subtasks and possible solution paths which are organized in the form of a tree (see Figure 2).
2. *Dynamic descriptions continuously maintained by the system* - psychometric parameters and usage indicators, both being required for the adaptive selection. Psychometric values correspond to the psychometric model used, while multiple models can be used simultaneously.

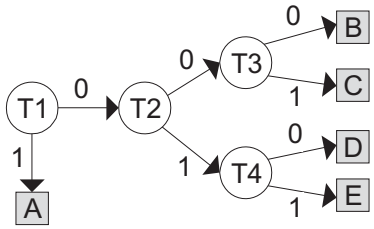

Fig. 2. Example of a scheme description tree having for each subtask only two possible answers. The task outcomes are represented by the leaf nodes A, B, C, D, and E.

Content authors specify scheme descriptions using an object language extended by a high-level library which provides them with abstract objects and operations. During the authoring process, the parameters and constraints are described by code fragments (see Example 1).

Example 1. Fibonacci sequence specification exported in XML. The author specified the sequence length and the generator code which is found in the CDATA section, resulting in a sequence of ten objects – numbers each having the value of the sum of the previous two:

```
<array name="fib" length="10">
 <singleton type="Integer" generator="code"><![CDATA[
   if (fib.Index <= 1) return Number.Integer (1);
   return Number.Integer (fib[fib.Index-2].Value+fib[fib.Index-1].Value);
 ]]></singleton>
</array>
```

Descriptions of subtasks and possible solution paths are specified using the XHTML markup language extended by a custom rendering element to allow the content authors to include a suitable rendering of the selected parameters.

Task instance generation process accounts for both procedural and declarative nature of scheme specifications. Using a pruned backtracking method, generation module instantiates the parameters in the order of appearance in the specification. If a parameter cannot be successfully constructed within k attempts during this process the instantiation process returns back to the previous parameter and tries another value. Increasing the value of k – while slowing the process – gives opportunity to produce more instances. We have found the value k=10 sufficient for high performance instance generation even in the presence of tens of parameters and constraints if they were specified efficiently. Slow instance generation at this value hinted at an ineffective code or parameter ordering.

Descriptions are fixed once the parameters in the schemes are instantiated, producing a task instance. Parameters in display templates are rendered for web delivery using MathML and SVG formats. Parametrization allows for not only a simple numerical variations between different instances but word and entity variations were employed along with preserving the structure of wording.

3.2 Assessment Process

The adaptive examination process requires a set of task schemes with precalibrated psychometric parameters. For the first run, parameters are determined manually by the content author. After each subsequent examination the parameters are recalibrated using all available student answers.

The 2PL IRT model [15] is used as a baseline for adaptive selection as long as more student answers are not available to grant a more sophisticated multidimensional IRT model. The structuring of tasks into trees does not allow for a straightforward use of a dichotomic model. As a helper, the system uses the 50% criterion, by which the student is awarded a correct answer for the current task if and only if he succeeds in answering at least half of the presented subtasks correctly (see Figure 3 for an example of a wrong task answer).

The adaptive selection of task schemes is initialized with an initial ability estimate of 0. To select the next task, the adaptation process considers all unused task schemes and identifies the maximum information value M that can

be attained by the most informative scheme at the current ability estimate. To balance the scheme exposure, a small set of task schemes having the information value close to the value of M is picked and a random scheme from this set is selected for administration. The adaptive selection of task scheme occurs at the beginning of the examination and each time the student reaches a leaf node in the solution tree of current task instance.

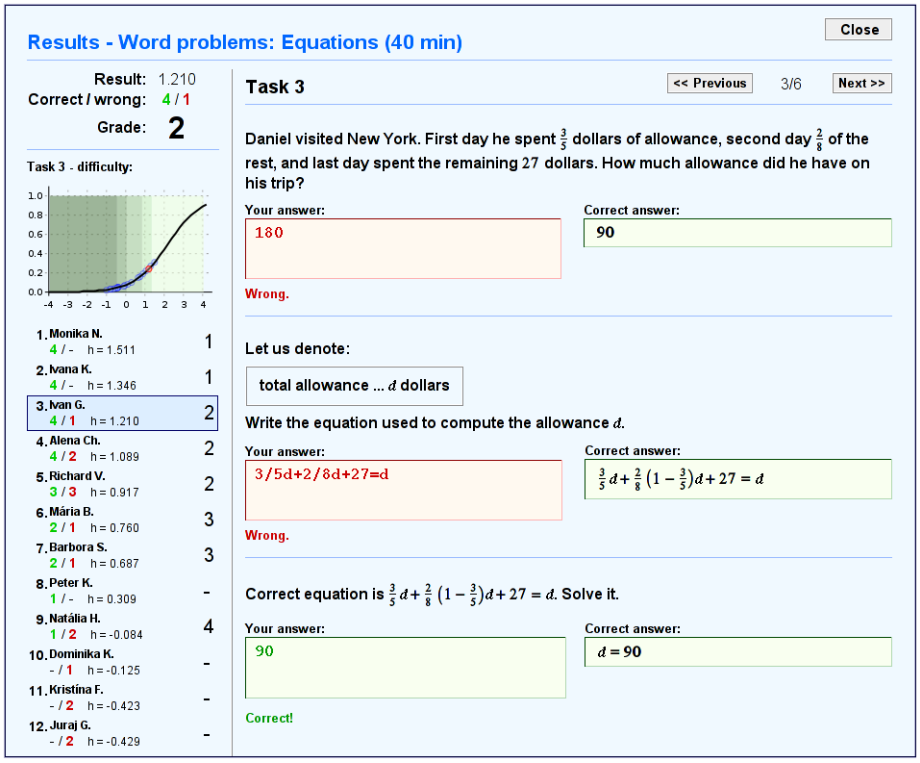

Fig. 3. After the examination is finished, students examine tasks' difficulties and compare answers with the correct ones. The student in the figure received a score of 1.210 by answering 4 tasks correctly and 1 task incorrectly (on display).

The examination finishes after a predefined test information value has been attained or none of the available schemes provides at least a threshold value of information. Consequently, grades are awarded only to students whose examination process resulted in at least a given test information value (see Figure 3). This threshold value is determined operatively by the teachers.

For a scheduled examination, students sign in to the system before the actual examination takes place for the assessments of all students in the class to commence simultaneously. Tasks are presented to the students one by one, each task is given a dedicated web page on which new subtasks appear as the student progresses in the solution tree. Students provide their answers in an open-ended format depending on the task type – free-text answer, drawing applet, etc.

For structured answer formats such as the geometry drawing applet, the answer is already specified in the domain's object model and the correctness of the automatic judge comparison procedure is thus straightforward.

For unstructured answer format such as the free-text answer, we at first try to transform the supplied answer into predefined answers described by structured templates of objects (numbers, equations, etc.) and proceed with the automatic comparison if possible. If no structure from the predefined set can be identified, the set of previously encountered unstructured patterns is consulted and the raw answer is classified using a vector-based machine learning algorithm. To prevent contamination of training examples, classification outcomes are later manually reviewed for correctness. Finally, if no sufficiently close match is found the answer is passed to a human judge to decide.

4 Evaluation

We have conducted experiments in the domain of middle school mathematics on a set of 45 students during the 2006-2007 school year. Our objective was to explore the feasibility of semi-automatic judging of student answers, compare adaptive task selection with human teachers, and qualitatively evaluate the suitability of this type of assessment using feedback from students and teachers.

A set of 7 task schemes of varying difficulties in the topic of word problems on linear equations was prepared. Normally a whole class examination of this topic contains 5 tasks with an allotted time of 35-40 minutes. Prepared tasks had all a similar structure such as the one depicted in Figure 2. Beginning with the subtask T1 containing the description of the problem, the student either submits a correct answer (A), or she is asked (in subtask T2) to provide the linear equation she had used in her solution. If the provided equation is correct she is asked to recompute its root (in subtask T4), or the correct equation is presented (in subtask T3) and its solution is demanded.

Table 1. Breakdown of students' answer patterns and an estimate of successful recognition by a contemporary machine learning algorithms based automatic judge.

Answer type	**N**	**prob.**	**Automatic judge**
Numerical	76	100%	76
Empty	60	100%	60
Identical string	12	100%	12
Equation object	77	100%	77
Unknown text	71	50%	35
Request for help	19	80%	15
Other numerical	90	80%	72
Total	**393**	**88%**	**347**

In the examination which took 40 minutes, we have collected 174 task answers in total, with a mean value of 3.867 task per student, and a total of 393 subtask answers. Table 1 breaks down the types of encountered answer patterns. Structured answer types: Numerical, Empty, Identical string, and Equation object were judged automatically, while the unstructured: Unknown text, Request

for help, and Other numerical (e.g. "My result is w=47.") were judged manually by a teaching assistant during the examination process. We argue that using contemporary machine learning algorithms it is viable to construct a classifier successfully classifying the unstructured answers with the estimated probabilities stated in Table 1. In any case, the workload of the teaching assistant was low during this process, receiving about 3 answers to judge per minute.

Table 2. IRT parameters of the task schemes in the experiment.

Task	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**
difficulty	0.514	0.600	0.368	-1.628	0.665	2.202	2.038
discrimination	1.674	0.574	1.607	0.443	1.747	1.154	1.220

Tasks schemes were calibrated using the 2PL IRT model (see Table 2). We have 5 schemes with a good discrimination value in $[1.154, 1.747]$ interval. With #2 and #4 having discrimination values of 0.574 and 0.443 respectively, we are expecting these schemes to be selected scarcely. Note that we do not have any schemes at difficulty level value near 0 and below except the low discriminative #4. Therefore it is expected that no appropriate tasks will be selected for students at these difficulty levels (average and low ability profile).

The adaptive selection of tasks is demonstrated on three student profiles – high, average, and low achieving student (see Table 3). Let us first examine the high ability student answering every question correctly. As expected, the adaptation process selects progressively harder tasks with good information values, granting her a final ability estimate of 1.926. Note that the resulting estimate is not infinite since we employ the Bayesian EAP (expected a posteriori) estimation procedure [16] with prior normal distribution of student abilities.

For the average ability student, we selected a student from our sample that is able to answer tasks #3 and #4 correctly and thus has an ability value of 0.105. As the opening task, she "accidentally" receives the #3 providing a correct answer awarding her a high ability estimate in the first step. Afterwards however, she is not that lucky and gets all the other tasks wrong. Similarly with the low ability student. Note the low information values in the 4th and 5th step of average and low ability student selection process hindering a more precise measurement. In fact, a precise measurement was not possible because of the lack of tasks at appropriate difficulty levels [2].

Finally, teachers and students were interviewed to provide a qualitative feedback. Excluding some occasional negative feedback in the high end and positive feedback in the low end of the ability scale, the students' positive attitudes were proportional to the attained ability estimate. Worth mentioning, students of all ability levels especially liked the structured approach presenting them with easier questions after a wrong answer, giving them the opportunity to ultimately feel success after the final, though possibly the easiest, question was answered correctly. Teachers valued that the proposed system assesses their students independently of any subjective input thus perceivably providing them with objective formative assessment throughout the year.

Table 3. Adaptive selection for high, average, and low ability student.

Step:	**0**	**1**	**2**	**3**	**4**	**5**
High ability student (with ability +inf)						
Task selected:		5	1	3	6	7
Task information:		0.554	0.662	0.454	0.257	0.350
Answer:		correct	correct	correct	correct	correct
Ability estimate:	0.000	0.798	1.128	1.304	1.639	**1.926**
Average ability student (with ability 0.105)						
Task selected:		3	5	1	6	7
Task information:		0.592	0.762	0.664	0.088	0.099
Answer:		correct	wrong	wrong	wrong	wrong
Ability estimate:	0.000	0.663	0.239	-0.20	-0.57	**-0.095**
Low ability student (with ability -inf)						
Task selected:		3	1	5	2	7
Task information:		0.592	0.391	0.252	0.070	0.040
Answer:		wrong	wrong	wrong	wrong	wrong
Ability estimate:	0.000	-0.442	-0.653	-0.771	-0.865	**-0.887**

5 Conclusions and Future Work

In this paper, we presented an adaptive assessment method proposed in an effort to empower traditional classroom assessment processes with computerized methods. It adaptively selects tasks according to the student's ability. Higher achieving students receive harder and lower achieving students easier questions, giving each the opportunity to demonstrate her ability level.

The proposed structuring of tasks into parametric solution trees which are described in a high-level object language using a library of domain objects makes a detailed assessment of student's solution possible. After submitting a wrong answer, the student is asked a deeper question regarding the solution path taken. Employing appropriate (e.g. polytomous) IRT models allows all of the demonstrated performance, be it right or wrong, to be included in the final ability estimate. In addition, structured parametric task descriptions facilitate automatic on-demand task generation and effective judging of open-ended answers.

We have evaluated the proposed method using a software system we had developed in the domain of middle school mathematics. By not administering a rigid set of tasks students fail to employ simple surface approaches to learning. In our observation after the experimental session students did not identify common problems to talk about at first since as much as 174 different task instances were administered even though only 7 task schemes were employed. Afterwards, students recognized the common features of the different instances each of them received, promoting higher order thinking skills.

As the next step we explore both individual-level and group-level improvements. On the individual level, observing the time spent on tasks and other in-system behavior patterns of an individual student can reveal interesting assessment and instructional opportunities. On the group level, we explore possi-

bilities of enhancing the method with collaborative activities. Having multiple students working on the same task or in the role of the judge may result in a meaningful activity and lessen the required workload of the judging procedure.

Acknowledgements

This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 3/5187/07 and the TENCompetence Integrated Project that is funded by the European Commission's 6th Framework Programme, priority IST/Technology Enhanced Learning.

References

1. Brusilovsky, P., Miller, P.: Web-based testing for distance education. In De Bra, P., Leggett, J., eds.: Proceedings of WebNet 1999, AACE (1999) 149–154
2. Mills, C., Steffen, M.: The GRE Computer Adaptive Test: Operational Issues. Computerized Adaptive Testing: Theory and Practice (2000)
3. Aleven, V., McLaren, B., Sewall, J., Koedinger, K.: The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains. ITS **2006** (2006) 61–70
4. Feng, M., Heffernan, N., Koedinger, K.: Addressing the Testing Challenge with a Web-based E-Assessment System that Tutors as it Assesses. Proceedings of the 15th international conference on World Wide Web (2006) 307–316
5. Lord, F., Novick, M.: Statistical theories of mental test scores. Addison-Wesley Reading, Mass (1968)
6. Wainer, H.: Computerized Adaptive Testing: A Primer. Lawrence Erlbaum Associates. Hillsdale, NJ (2000)
7. Black, P., Wiliam, D.: Inside the Black Box: Raising Standards Through Classroom Assessment. Phi Delta Kappan **80**(2) (1998) 139–148
8. Conejo, R.: SIETTE: A Web-Based Tool for Adaptive Testing. International Journal of Artificial Intelligence in Education **14**(1) (2004) 29–61
9. Conejo, R., Guzmán, E., Pérez-de-la Cruz, J.L., Millán, E.: An Empirical Study About Calibration of Adaptive Hints in Web-Based Adaptive Testing Environments. Adaptive Hypermedia and Adaptive Web-Based Systems (2006) 71–80
10. Irvine, S., Kyllonen, P.: Item Generation for Test Development. Lawrence Erlbaum Associates, Mahwah, NJ (2002)
11. Bejar, I., Lawless, R., Morley, M., Wagner, M., Bennett, R., Revuelta, J.: A Feasibility Study of On-the-Fly Item Generation in Adaptive Testing. Journal of Technology, Learning, and Assessment **2**(3) (2003)
12. Sinharay, S., Johnson, M.: Analysis of Data From an Admissions Test With Item Models. Educational Testing Service (2005)
13. Deane, P., Sheehan, K.: Automatic Item Generation via Frame Semantics: Natural Language Generation of Math Word Problems. Annual meeting of the National Council on Measurement in Education, Chicago, IL (2003)
14. Chang, H., Qian, J., Ying, Z.: a-Stratified Multistage Computerized Adaptive Testing with b-Blocking. Applied Psych. Measurement **25**(4) (2001) 333–341
15. Birnbaum, A.: Efficient design and use of tests of a mental ability for various decision-making problems. Randolph Air Force Base, Texas: Air University, School of Aviation Medicine (1957)
16. Baker, F., Seock-Ho, K.: Item Response Theory: Parameter Estimation Techniques. CRC Press (2004)