On the Impact of Adaptive Test Question Selection for Learning Efficiency

Abstract. *In this paper we present a method for adaptive selection of test questions according to the individual needs of students within a web-based educational system. It functions as a combination of three particular methods. The first method is based on the course structure and focuses on the selection of the most appropriate topic for learning. The second uses Item Response Theory to select the k-best questions with adequate difficulty for a particular learner. The last is based on the usage history and prioritizes questions according to specific strategies, e.g. to filter out the questions that were recently asked. We describe how these methods evaluate user answers to gather information concerning their characteristics for a more precise selection of further questions. We describe an evaluation of the impact of a proposed method through two different types of experiments in the domain of learning programming, which both showed that our method for adaptive test question selection increases the overall learning outcome, especially for lower than average performing students.*

Keywords: *Learning programming, web-based educational system, adaptive test questions, domain model, metadata*

1 Introduction

Learning supported by information technologies has become the standard and expected way of providing education. The Web seems to be one of the most efficient media to learn from, if we are able to get over the traditional ”just-put-it-on-the-Web” approach and provide learners with educational content adapted to their individual needs and current skills (Brusilovsky & Peylo, 2003).

Apart from the any-time and any-where access, web-based education brings another substantial advantage over the traditional methods of education using books and other printed materials – interactivity in the form of exercises and questions with instant feedback. The learning does not need to be simply a passive reading of materials, but instead can be a mixed approach, consisting of questions (with immediate feedback and

explanations of correct answers) chosen appropriately by the learning system. The possibility of active interaction with the educational system presents a substantial incentive for learners to actually use the system (Sun & Hsu, 2007).

Another significant advantage of alternating explanatory texts with interactive questions is that answering test questions provides a feedback for the educational system itself (Brusilovsky & Millán, 2007). The educational system can use this feedback to estimate the level of user knowledge. This information can be used during the question selection process to select questions with an appropriate level of difficulty for the user. Moreover, the estimation of the learner's knowledge can also assist in providing effective navigation through the educational content, e.g. a recommendation of next topic for study for a particular user.

Although there are many educational web-based systems that consider estimated learner's knowledge for providing personalized content or navigation, they consider learning objects as elements to be adapted using various approaches based on heuristics or rules expressing knowledge on adaptation. Our aim is to consider test questions as a part of educational material and improve adaptation by considering their difficulty and other attributes that cannot be considered for general educational materials. We present a novel method for learning programming in an undergraduate study programme based on mixing textual study materials with a test question selection according the Item Response Theory (IRT). We combine topic selection and prioritization following the answer history to achieve an adaptive selection of questions for the needs of each individual student. We applied the method within a web-based educational environment in several subjects of programming in a bachelor study programme *Informatics* – Functional and Logic Programming, Procedural Programming and Object-Oriented Programming. We present results of experiments on procedural and functional programming.

The paper is structured in the following manner. Section 2 discusses related work in the field of adaptive web-based educational systems and testing systems. Section 3 gives a description of the proposed method for adaptive test selection. Section 4 presents an evaluation of the proposed method in the context of two different experiments. Finally, we give our conclusions.

2 Related work

There are many educational web-based systems which try to benefit from the fact of "being online" and instead of just providing the same information as can be found in text books, they focus on enhancing the learning experience by an added value in the form of interactivity and adaptation to the individual learner.

Most of the adaptation, which can be found in state-of-the-art systems, is focused on navigation support and content presentation. One such system is NavEx (Yudelson & Brusilovsky, 2005), which provides a personalized access to annotated programming examples. The NavEx system tracks student's knowledge of course concepts and uses this knowledge to annotate the navigation links as being "not ready to be browsed" or "ready to be browsed", in a 5 degree scale. The evaluation of whether a student has mastered a concept or not is done by a simple mechanism based on counting clicks on annotations of presented examples, with the assumption that if the student clicked on an annotation, he actually read it *and* understood its content.

The web-based educational system ALEA (Bieliková, 2006), was created to support courses of functional and logic programming. Compared to the previously mentioned system NavEx, it incorporates different types of content (learning texts, schemas and templates, examples), which are organized in a hierarchical structure with additional relations between them – e.g., prerequisite or similarity relation. The user modeling part of ALEA is based not only on the fact that learner has seen the concept (has clicked on it), but also on additional attributes such as the time that he spent reading the concept. The learner also has the possibility to explicitly tell the system that he understood the presented concept.

The method presented in (Šimún, Andrejko, & Bieliková, 2008) uses user knowledge and interest for recommending learning objects. The authors proposed a multi-layered user model on top of the domain model, consisting of learning objects linked to concepts by various kinds of relations such as similarity, prerequisite, hierarchy, and a way how expressed user interests on one learning object can be spread over new concepts and learning objects by using these relations. Our domain model follows the basic principles proposed, while at the same time we stress more strongly the relationships between concepts and learning objects.

QuizGuide (Brusilovsky, Sosnovsky, & Shcherbinina, 2004) is a system with adaptive navigation support similar to NavEx, which provides access to self-assessment quizzes. The user model used for setting link annotations is built by checking student's answers to displayed questions. Each question is manually assigned one of the three levels of complexity, which afterwards influence the contribution of correctly answering the question to the mastery of topic related to the question.

QuizPack (Brusilovsky & Sosnovsky, 2005) represents a tool for dynamic generation of questions from question templates and evaluation of user answers. It is used in domain of programming – a teacher provides questions which are in this case represented as parameterized fragment of code and the expression to evaluate students answer. Questions are then randomly selected from the template base and the user's answer is evaluated. Random selection can be sufficient in domains where the difficulty of the questions is almost the same, or it is useful to solve most of the questions, because they involve some sort of practice (e.g. multiplication of small numbers).

The previously mentioned systems model the learners by employing explicit feedback and a clickstream-based heuristics, which is a working and widely used solution. From the analyzed educational systems, only QuizGuide uses an evaluation of the learner's knowledge through questions and quizzes, but does not incorporate any standardized way of selecting the questions (in fact, it is the student himself who chooses the difficulty of questions he wants to solve).

Having in mind the usage of questions incorporated in an educational material as the driving force for learning improvement, knowing the user knowledge level is crucial for adapting the difficulty of the question to the particular learner. Psychometric Item Response Theory (IRT) represents an approach that combined with other techniques, enables such dynamic and adaptive question selection. Several existing systems use IRT. System SIETTE (Conejo, Guzmán, Millián & others, 2004) is a web-based adaptive *testing* (i.e., not used for learning purposes) system, which adapts to student's needs using Computerized Adaptive Testing (CAT) and three parameter logistic IRT (3PL IRT). It enables teachers to define tests and students to take them on-line. It uses Java applets so the student can interact with the system by means of applet during the question answering. Inspire (Papanikolaou, Grigoriadou, Kornikolakis, & others, 2003) is

also a web-based system that employs user's progress acquired during learning. It uses IRT with 3PL model, but keeps discrimination factor constant with a value of two.

One of the problems in current adaptive selection of questions is that there are several methods and many different models of IRT which can be used, but there are only few experiments comparing which one is better for various domains and why. Another problem is that basic models describe user knowledge only by one value, which in many courses is not sufficient. There are also difficulties in question base calibration, especially in more complex IRT systems. Our method not only uses the IRT model, which employs several values to express student's knowledge – one value for each course topic, but also combines model-based approaches with heuristics-based approaches, considering semantics expressed by an educational course structure and an answer history.

3 Adaptive selection of questions

Our aim is to select one question from the set of existing questions which is most appropriate for a particular student and a particular content (e.g., the student actually learns programming loops). We propose an approach for adaptive selection of questions that works as a combination of three methods applied sequentially as filters. Each of them tightens the collection of questions that could be potentially used and in the end only most beneficial one remains. Even if the methods are proposed as independent, they are chained in such a way that they use the results of particular predecessor in the sequence.

Figure 1 depicts the whole approach of question selection and application of each selection method in the order as they are applied. According to their ability to adapt to particular user, the methods employ and modify a user model data, which accumulate all gathered information about the user including his preferences and knowledge level.

The overview of how each method selects a question and evaluates user answers is summarized in Table 1.

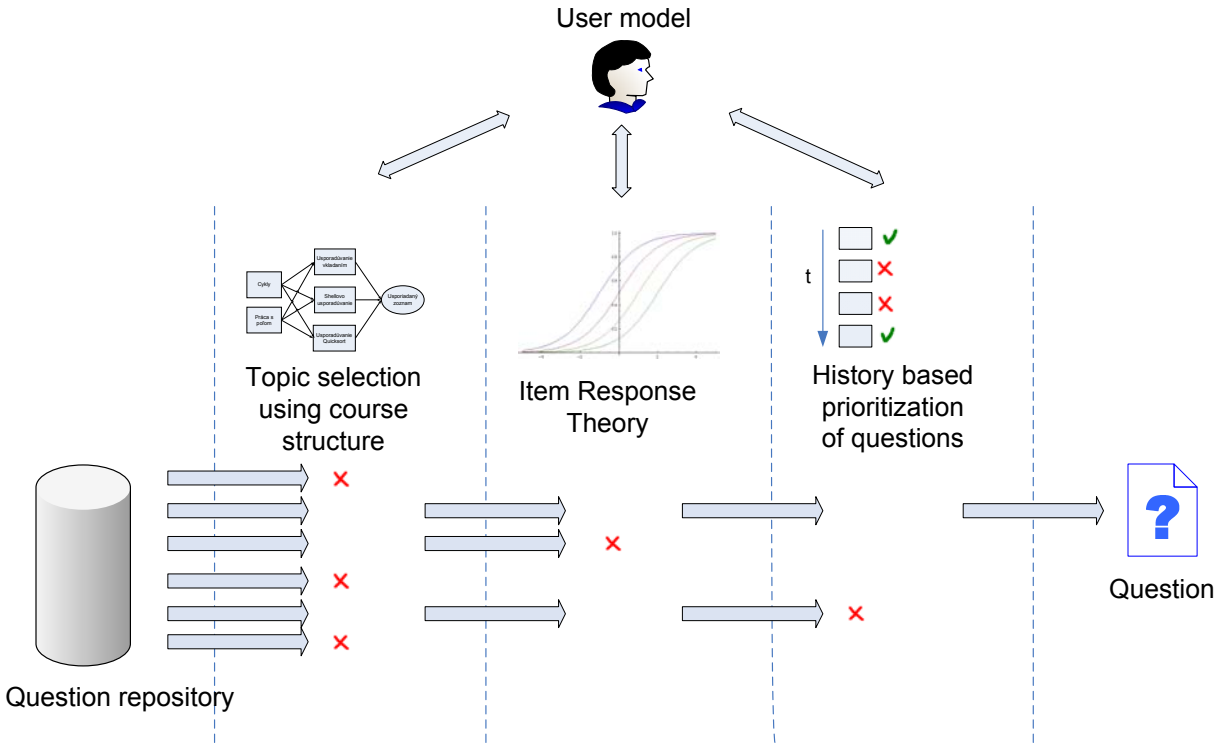

Figure 1. Adaptive question selection.

Table 1. Overview of selection and user feedback evaluation for each method.

Order	Method name	Question selection	Answer evaluation
1	*Topic selection using course structure*	Selects the most appropriate topic for a question	Employing user knowledge estimation as a result of second, IRT based method it updates topics for the user appropriately
2	*Item Response Theory*	Selects *k*-best questions with most appropriate difficulty for the particular user	Updates the estimation of user knowledge of the question topic
3	*History-based prioritization of questions*	Selects one question that was not recently asked or fulfills other history-based strategy	Updates the log with time and correctness of the user answer

3.1 Course structure based selection

One of the reasons why we do not rely solely on the IRT is that we use the IRT in combination with the course domains (educational materials) which are richly articulated in topics and subtopics and therefore cannot be described by means of only one or few

numeric values. So we have also opted for a method for educational material recommendation based on course structure.

Our method uses the part of a domain model that forms a structure we define to be a ‘prerequisite graph’ (see Figure 2). For each course topic, the prerequisite graph defines the additional topics that a student must know before proceeding further. Topics are represented by concepts in our domain model. The prerequisite graph is an acyclic oriented graph with two types of nodes, and associates required prerequisites for each topic. Conjunctive nodes (depicted by squares) require all prerequisites to be mastered, disjunctive (depicted by ellipses) are fulfilled by mastering at least one of them. The prerequisite graph is created by domain experts (teachers), however, this process is significantly supported by software tools which are responsible for mining concepts as well as relationships among them (Šimko & Bieliková, 2009). In an ideal case, the teacher just checks and confirms the generated graph.

Each course topic can be in one of three states (see Figure 3):

- *unavailable*,
- *opened* or
- *mastered*.

This involves using results from the IRT method in an evaluation phase to quantify user knowledge of each topic, represented by a floating-point value. There are also two explicitly defined values shared among all topics, *master value* and *failure value*.

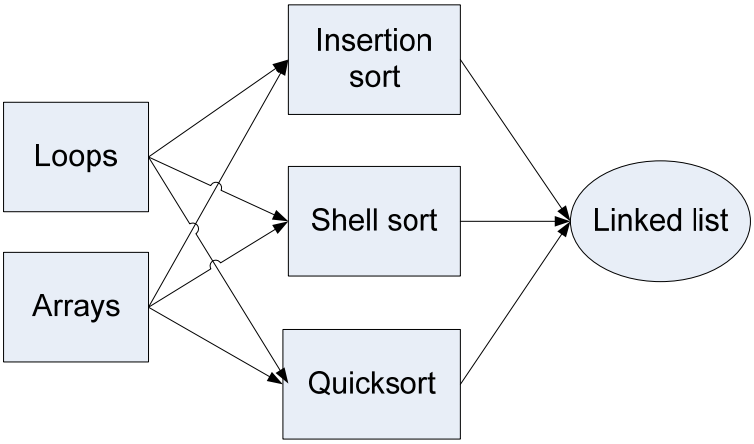

Figure 2. Prerequisite graph.

In the beginning of the educational course, all topics are unavailable except of those that have no prerequisites. Topics are mastered if student's knowledge exceeds a master value (Figure 3). This event triggers a check for the prerequisite fulfillment and can open unavailable topics. There is also an inverse process and the opened topic can be closed if user knowledge of the topic is lower than a predefined failure value. The states of a prerequisite may also degrade over time and move from mastered to opened.

During selection we choose an opened topic either randomly or explicitly.

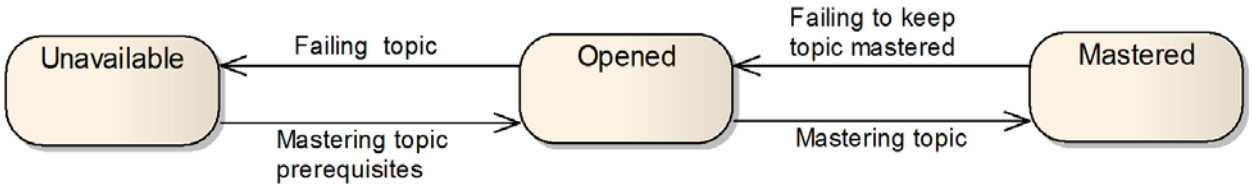

Figure 3. State diagram of course topic.

3.2 Item Response Theory based selection

Being a method from psychometrics rather than computer science, IRT (Embertson & Resise, 2000) defines a model which can be used for adaptive question selection (also known as Computer Adaptive Testing (Linacre, 2000)). Instead of traditional testing, where all questions are chosen before testing, the estimation process in IRT is derived by maximizing the likelihood of a person's observed response pattern in a model of test

behavior (Embertson & Resise, 2000). We do not use one value to express student's knowledge of the whole course, but one value for each course topic. Because a course topic is selected by first method in the chain, we can employ the user knowledge of this topic in our IRT model.

In the IRT model, every question is characterized by *Item Characteristic Curve* (ICC; see Figure 4). It determines the relation between a student's knowledge expressed by floating-point value (x-axis) and the probability that his answer will be correct (y-axis). ICC can be modeled by various functions, in our case it is a commonly used three parameter logistic function also known as 3PL (Equation 1).

$$P(\theta) = c + \frac{1-c}{1+e^{-a(\Theta-b)}} \qquad (1)$$

where *θ* stands for student's knowledge, *a* for discrimination factor, *b* for question difficulty and *c* for probability that student will guess the correct answer, also called *guessing factor*. We set parameters *a* and *b* manually for each question during its authoring by selecting from predefined values. Parameter *c* is computed based on the question type and the number of possible answers (we cover the following question types: single choice, multi choice, multi choice with known number of correct answers, ordering and pairing).

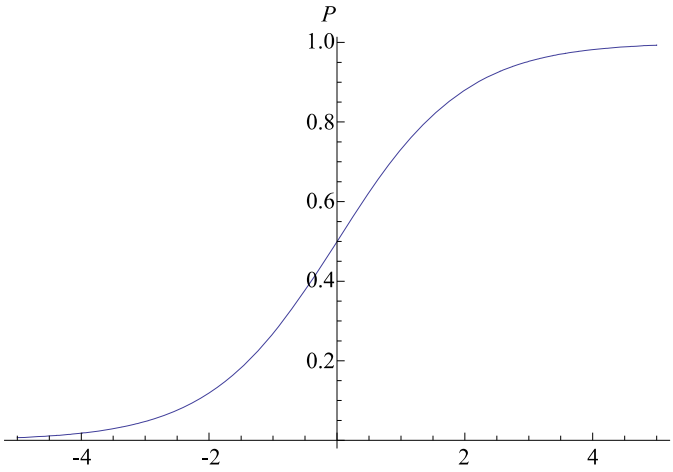

Figure 4. Item Characteristic Curve.

The student's knowledge *θ* is defined by IRT as an arbitrary floating value between -∞ and ∞, where 0 usually represents the knowledge of an average student (it depends on

the calibration). For practical purposes, the value range of student's knowledge is often trimmed to interval <-3, 3>. The discrimination factor affects how steep the item characteristic curve is – higher value means higher slope. When the values asymptotically reach infinity, the characteristic function will have just two values, zero and one.

In a process of question selection it is important to express the usefulness of each available question in the view of how will the answer help to more precisely determine the level of user knowledge. This is achieved by the *item information function*.

Figure 5 shows item characteristic curve (ascending curve) and its information function which determines the relation between user knowledge and the amount of information that will be gained from his answer. It shows that this question is most appropriate for a user with knowledge of 0.5. The important fact is that from the user's point of view, selecting questions with highest information function makes the system appear as adapting to his knowledge level.

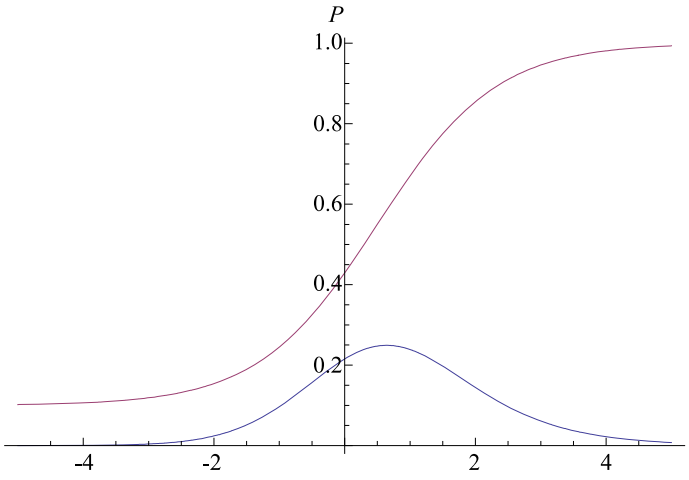

Figure 5. Item Information Function

The item information function is computed directly from the ICC. Equation 2 shows the form used for 3PL model.

$$I(\theta) = \left[a_i^2 \frac{1 - P_i(\theta)}{P_i(\theta)} \right] \left[\frac{(P_i(\theta) - c_i)^2}{(1 - c_i)^2} \right] \qquad (2)$$

where P_i is characteristic function with parameters a_i, b_i and c_i evaluated for user knowledge θ.

Evaluating the user's answer in IRT actually means computing more precise estimation of user knowledge. It is achieved by computing the maximum of likelihood function which determines the relation between the knowledge value and the probability that it equals the knowledge level of a user. Equation 3 presents the formula for computing the likelihood function.

$$L(u_1, u_2 ... u_i \mid \theta) = \prod_{i=1}^{n} P_i(\theta)^{u_i} Q_i(\theta)^{1-u_i} \qquad (3)$$

where L is the likelihood function for vector of user's answers u (u_i has zero value if the answer to the i-th question was incorrect, otherwise it is equal to one), θ is an estimation of user knowledge, and n stands for count of questions from which we compute the likelihood function, P_i is a characteristic function for i-th question and Q_i is evaluated as $1 - P_i$.

However, this basic approach cannot be applied to vectors with all correct or all incorrect answers. Therefore we use a method, which is called *expected a posteriori* (Embertson & Resise, 2000), based on the Bayesian method.

The method uses values computed during its initialization. This process consists of two following steps:

1. *Discretization of knowledge axis* – Interval <-3,3> on knowledge axis is evenly divided using value 0.1 as a step. These values are inspired by (Embertson & Resise, 2000). We will get 61 discrete points which we will express as Q_r, where r represents the index of a point (value between 1 and 61).
2. *Computation of density of normal distribution* – A value called *weight* is computed for each discretization point (Equation 4).

$$W(Q_r) = \left[\frac{1}{\sqrt{2\pi}}\right] e^{\frac{-Q_r^2}{2}} \qquad (4)$$

Afterwards, we are able to estimate user knowledge based on his *N* previous answers. The estimation proceeds in two steps:

1. *Computation of log-likehood function* – A value of *log-likehood* function is computed for all discretization points (Equation 5), where variable u_i stands for user's answer on *i*-th question (it has value 1 if it was correct or 0 otherwise), P_i is characteristic function for *i*-th question and Q_i is evaluated as 1 - P_i.

$$\log L(u_1, u_2..u_N \mid \theta) = \sum_{i=1}^{N}\left[u_i \log\left(P_i(\theta)\right) + (1-u_i)\log\left(Q_i(\theta)\right)\right] \qquad (5)$$

2. *Knowledge estimation* – Estimation of user knowledge is computed using following equation:

$$\theta = \sum_{r=0}^{61} \frac{W(Q_r) \times Q_r \times L(Q_r)}{\sum_{p=1}^{61} W(Q_r) \times L(Q_r} \qquad (6)$$

   where *L represents* log-likelihood function evaluated in discretization points.

Unlike the common realizations of IRT where one question with maximum information function is selected, our method selects defined number of questions with highest information function and passes them to the third method.

3.3 History-based selection

The aim of the history-based selection of a question is to prevent the use of the same question for a particular user and to improve the quality of the question selection. Each answer on the question is stored in the user model together with the time required to answer and whether the answer was correct or not.

This method is applied after the IRT and it selects the question using one of following strategies:

- *Selection of least recently used questions*: This strategy compares the time when each question was last used. Questions that were not correctly answered are preferred in a way that length of the time from their last use is multiplied by the factor greater than one (this value was calibrated during experiments). It means that it is possible that an incorrectly answered question can be preferred even if it was used more recently than other, correctly answered questions.

- *Using correctly answered questions again*: Questions which are answered incorrectly many times in the beginning and then answered correctly only few times are preferred. This strategy is used when there is a long time of user's inactivity within the system and when there is a significant chance that the user has forgotten what he had learned.

To decide which strategy will be used we define a probability value which determines how often the associated strategy will occur. For the best performance we estimated that *selection-of-least-recent-used-question* strategy should be used 95% of the time, and the remaining occasions should use the *using-correctly-answered-question-again* strategy.

3.4 Domain and user models

We distinguish between educational content presented to students and knowledge elements contained within them. Educational content is divided into educational segments known as *learning objects*. We consider following learning object types in the domain of learning programming:

- explanation,
- exercise,
- example,
- question.

Learning objects are often arranged in the educational course in a structure similar to a printed book – they form chapters, subchapters, etc. If they reference each other,

explicit links are created, allowing a non-hierarchical relationship modeling (e.g., learning objects similarity).

Domain knowledge is represented via knowledge elements or topics referred to as concepts. Each learning object is associated with several concepts and each concept can be related to several learning objects. The relationships are assigned a weight from the interval <0;1> denoting the degree to which the learning object "contains" the concept. For example, a learning object can be related to the "loops" concept by a weight of 0.8 and to the concept of "file I/O" by a weight of 0.2.

Concepts themselves are interconnected via a *related-to* relationship, resembling the structure of a lightweight ontology. The relationship weight denotes the degree of mutual relatedness. Relatedness measures represent any form of cognitive connection between two concepts. For instance, the "loops" concept is related to the "conditions" concept. By mastering the "conditions" concept, the user also learns part of the "loops" concept (as we use stop conditions to control the flow of *for* and *while* loops). All of the aforementioned relationships are gained by text analysis of available course materials (learning objects) (Šimko & Bieliková, 2009).

We refer to the concept structures described above as a domain *metadata* because it contains data about the content of domain resources (learning objects). Together with learning objects and associations it forms a domain model (see Figure 6), which allows us to perform reasoning over the learning domain and thus achieve a more advanced functionality of the educational system, such as recommendation.

As we mentioned already, we are able to compose the major portion of the domain model automatically. We significantly reduce teacher's (course author) effort and address the problem of domain model authoring complexity that is one of the major bottlenecks of adaptive educational systems usability.

The domain model also serves as a basis for user modeling. We employ the principle of the overlay user model (Brusilovsky & Millán, 2007), which adds user-related information to domain elements. Concept structure is used to store the domain knowledge of each learner, and learning objects are then used to store the history of user's interaction with the content. The actual knowledge is modeled in a discrete way. As a learner gradually

learns from learning objects, interacts with the educational system, and answers the questions, his knowledge of related concepts changes.

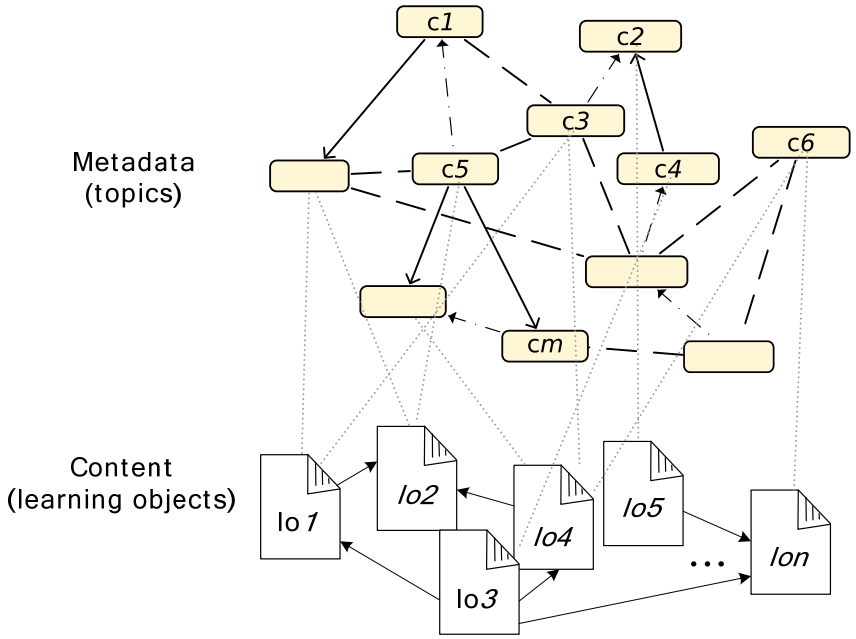

Figure 6. The domain model consists of metadata and content layer.

4 Evaluation

4.1 Learning environment and supporting tools

We realized the proposed method in a web-based learning system *Flip* developed within the project PeWePro[1] (Návrat & Bieliková, 2009). *Flip* consists of the following main parts related to the method presented in this paper (Vozár & Bieliková, 2008):

- *Learning course editor*: enables the educator to create the concept hierarchy, prerequisite structure, learning texts in XTHML format, and test questions in

[1] PeWePro project, `http://pewepro.fiit.stuba.sk`

Question and Test Interoperability 2.1 (QTI) standard format[2] (Radenković, Krdžavac, & Devedžić, 2007) enriched with IRT parameters.

- *Adaptive questions selection*: is the actual learning environment (*Flip*), which realizes adaptive question selection method and user answer evaluation.

Figure 7 displays a screenshot of the Flip user interface. A student is presented with a tree structure on the left, following the structure of the course (using hierarchical relationships between learning objects), and with actual texts and questions in the central area (with one multi-choice question displayed actually on Figure 7).

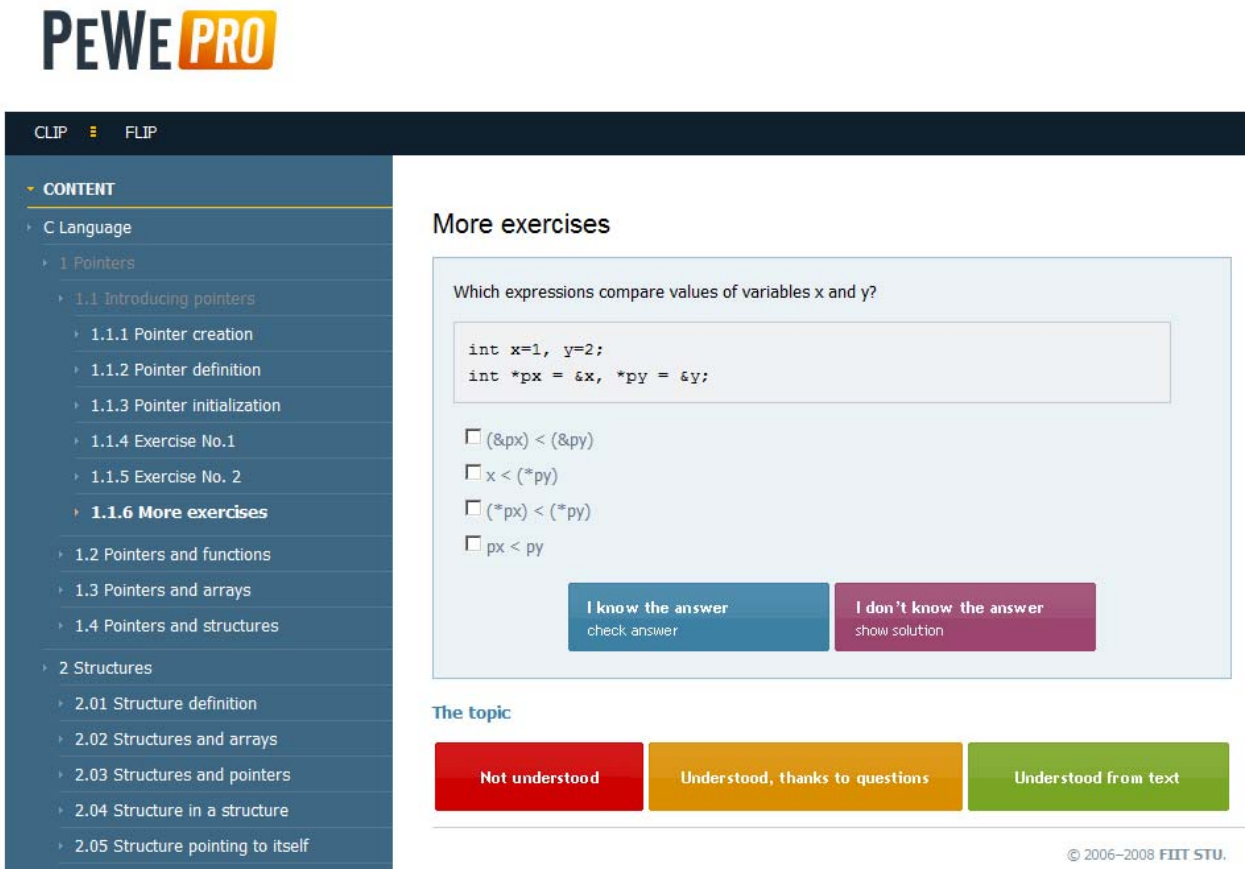

Figure 7. Screenshot of the learning environment interface showing a course structure on the left side and multi-choice question along with controls in the central area.

[2] IMS Question and Test Interoperability Overview, Version 2.1 Public Draft Specification, 2006, `http://www.imsglobal.org/question/qtiv2p1pd2/imsqti_oviewv2p1pd2.htm`

Below the texts and questions are three buttons for expressing student's explicit feedback on the displayed content. The student can explore the course content either by clicking directly on any topic within the course structure, or by using the aforementioned three buttons, when the system chooses the most appropriate topic to study according to the actual learner's knowledge and the prerequisite graph mentioned in section 3.1.

4.2 Uncontrolled long-term experiment

4.2.1 Experiment scenario

To perform a large-scale evaluation, we proposed to our undergraduate students to use *Flip* as an additional learning resource for their winter term course of Procedural Programming. We filled the system with texts, questions and exercises covering the topic of pointers in the C programming language. In a total of 38 questions there were multi-choice and single-choice questions, as well as questions where students needed to supply the correct answer or fill-in the missing words. We also supplied four more advanced exercises, where students were asked to code a small program, and were given an option to request a hint and/or to show the correct solution.

The students were using *Flip* to learn and practice the topic in order to prepare themselves for the mid-term exam (post-test) on pointers. During this time, all their activities within the system were logged. Right before the post-test exam (which was conducted as a real exam impacting students' final grade in order to increase students' motivation) we distributed a simple questionnaire, which took about 5 minutes to complete and collected students' answers. In total, we gathered answers from 264 students taking the exam.

We could not evaluate the impact of the underlying method of adaptive question selection on the individual level, due to uncontrolled setup of the experiment with following limitations:

- No pre-test – there might have been experienced students, who were familiarized with the topic prior to the course start, so they did not feel the need to learn and/or to use *Flip*, and yet they still managed to correctly answer the pointer questions even better than their classmates. However, we had no possibility to conduct a pre-test on such a large group and as an official part of the course,

which would eliminate this phenomenon. As a small replacement, we included a question in the questionnaire where we asked the students to evaluate their subjectively perceived mastery of the subject. The majority of students were not very confident about their knowledge – in the majority of answers they assigned themselves grades[3] C (22.3%), D (34.9%) and E (27.5%). Only 3.3% of students felt that their knowledge could be evaluated with grade A, and 4.08% of students evaluated themselves as failing (F grade).

- Unrestricted access to other learning resources – there is a chance that students, who were not using Flip found other study materials of superior quality compared to study texts and questions in Flip and were using it more exhaustively compared to the usage of Flip by other students.

- No control group – we could not divide students into disjoint groups, each with a different version of the system, which would help us to evaluate impact of the questions on learning efficiency. Students would very likely notice the difference, as we could not prevent various types of communication and collaboration during preparation for the exam.

Instead of individual level evaluation (which is presented in the second experiment), our goal for this experiment was to evaluate a contribution of the presence of the system on the overall results when compared to past years, when the system was not available. More, we were also interested in subjective evaluation of the system, acquired from the way students used the system as well as from the questionnaires given to the students before the exam.

The graph on Figure 8 shows the users' activity in the system from October 27 to November 4, 2008, one week from the deployment of the system till the post-test. We can clearly see the time at the beginning when the URL of the system was given to students. Then the activity naturally decreased, but we can see that as the exam was approaching (the exam was scheduled on November 5, 2008), students were using the system more and more again. From the fact, that students were coming back to the

[3] Students were familiar with the ranking system used at our faculty. Grade A represents at least 94% mastery of a given subject, grade B at least 84%, grade C 72%, grade D 62% and grade E 56%.

system and were using it repetitively, we can conclude that they perceived *Flip* as useful. This was furthermore confirmed by answers from the questionnaire, where almost 60% of students declared they were using *Flip* and 40% of all students found this learning resource as the most useful (compared to lecture notes and any other learning materials). Only 2% of students declared that *Flip* system did not help them at all to grasp the topic of C programming language pointers.

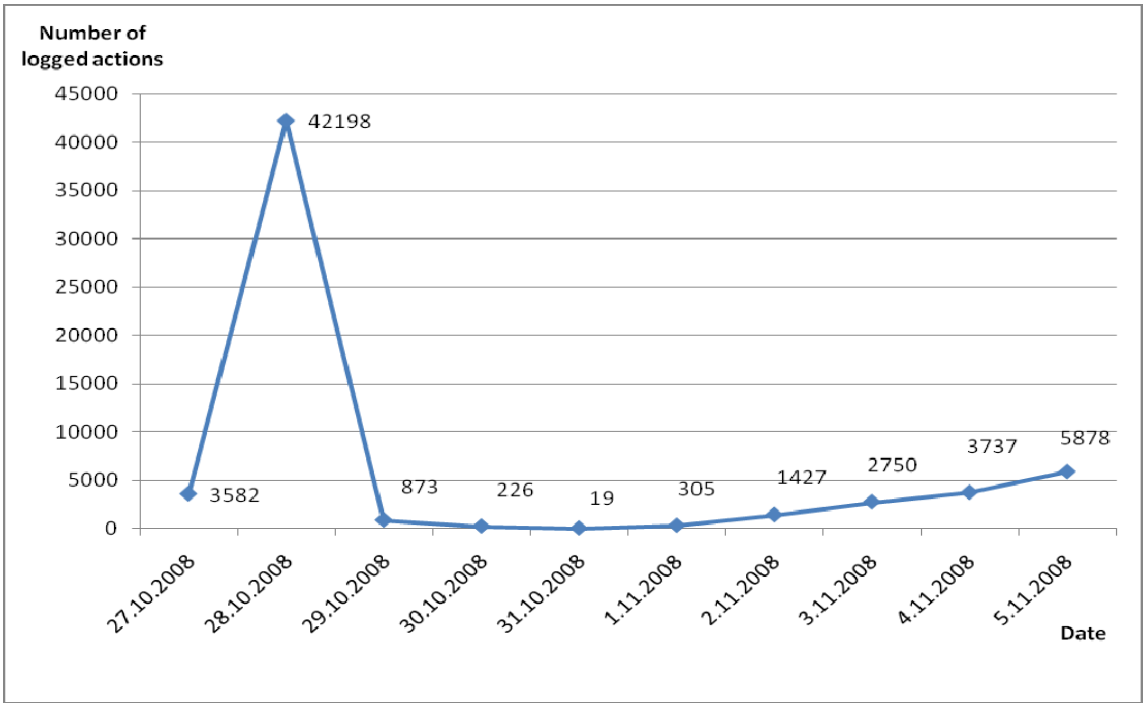

Figure 8. The graph shows the users' activity within the system from October 10 to November 5,2008 using a linear scale.

4.2.2 Results and discussion

After the questionnaire, students were given an exam, which (being an ordinary exam as a part of the course) covered a broader scope than just the pointers in the C programming language, which were available to study in Flip. We were therefore considering only 4 relevant “C pointers” questions from the exam on which students could gain 6 points. One of the questions was taken directly from Flip, without any further changes. The average score of this part of the test was **3.6** points. Figure 9 shows the distribution of the score among all students.

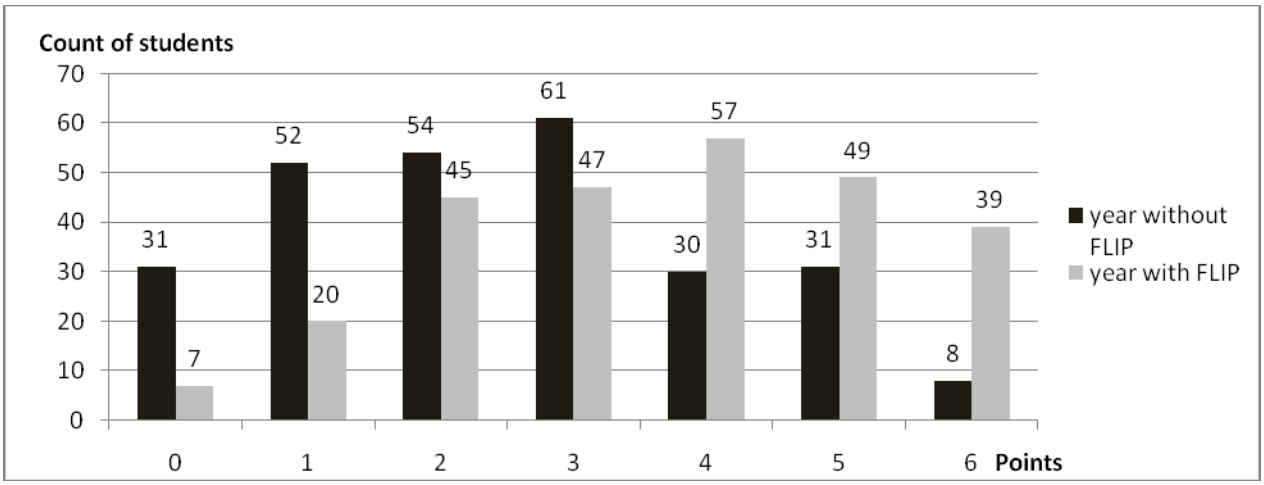

Figure 9. Post-test score distribution, when considering only questions related to pointers in C language, the topic which could be studied using Flip.

We compared the results against the one which took place one year before, when Flip was not available. Although the questions were obviously little bit different, all other conditions can be considered equal: same topic (pointers in C language), same number of questions (4) with comparable difficulty (both tests were prepared by one and the same person – an experienced pedagogue who took special care to prepare the tests with equal difficulty), same lectures and practically the same number of participants (264 vs. 267).

When comparing score distribution of this and the previous year (see Figure 9), we realized that the results of the previous year (without Flip) are much worse, with an average score of 2.5 compared to 3.6 out of possible maximum 6 (T-test confirms statistically significant difference between means with a P value of less than 0.0001). The standard deviation of the *with* and *without Flip* year was 1.61 and 1.62 respectively, which largely fulfills the requirement for homogeneity of variance property (the dataset passes Levene's test with significance equal to 0.01). Because the overall results of previous year exam were not an outlier, students in earlier years were achieving similar score and the results of these students in other subjects were similar, and from the fact that the overall exam conditions were equal (same duration, same total number of questions with consistent difficulty), we can conclude that this year's shift of overall results towards higher score is caused mainly by the fact that students took advantage of Flip.

4.3 Controlled short-term experiment

4.3.1 Experiment scenario

Knowing that Flip contributed positively to overall learning efficiency, we wanted to evaluate more precisely the impact of the method on an individual level. We conducted another, more controlled and closed user study. We gathered a group of 33 students with no prior knowledge of the functional programming paradigm and LISP language and let them study this topic in our system for a limited amount of time (70 minutes). We restricted access to any other learning materials which can be found on the web. The main focus was given on list processing, which is the basic concept of LISP language. The system was set up with 74 questions related to different LISP concepts. Similar to the uncontrolled experiment, we prepared single and multi-choice questions, as well as fill-in response questions. Right after the learning session, students participated in a post-test consisting of 7 tasks, which examined the practical as well as theoretical skills acquired during the learning session.

Students were divided into three distinct groups, each having the same number of participants:

- *Group A* (control group) – participants from this group were working with a modified version of Flip, with all interactive content turned off. The system served only static learning materials.
- *Group B* – similarly to group A, we presented a static (without interactive questions) version of Flip to participants belonging to this group. However, the presented texts were enhanced by annotations, small pieces of information which appear when hovered on with mouse, explaining key concepts of the programming language (so the learners did not need to click back and forth in order to recall what some function does (Mihál & Bieliková, 2009)).
- *Group C* – participants from this group were working with a full version of Flip, with the adaptive question selection and the content navigation. However, their study materials were not enhanced by annotations as in group B.

The students were not assigned into groups randomly, but according to their current overall study performance and weighted study average, so that groups were more or

less balanced in terms of cognitive abilities. The metric used is well known to all of our students and is used to rank students at our university for various purposes. However, it is important to mention that students were not informed to which group they were assigned. They did not even know about existence of any groups or multiple versions of the system. Moreover, they were not presented with the real goal of the experiment, as it could influence their performance. The only information they got was the instructions needed to use the system during the learning session, and then to take a test and to try to get the best possible result.

Our hypotheses (unknown to students) were as following:

1. Group C would achieve a better result in the post test compared to group A.
2. Group B would achieve a better result in the post test compared to group A.
3. Group C would achieve a better results in the post test compared to group B (we wanted to know whether questions are more valuable help than annotations or vice versa).

4.3.2 Results and Discussion

The overall results (achieved average and standard deviation for every group) of the experiment are displayed in the Table 2 and Figure 10.

We can see that every group scores around 15 points (of max 27), with adaptive question selection (group C) causing only a bit better results than in two other groups and annotations (group B) only decreasing the standard deviation, thus narrowing the bell curve.

Table 2. Results of the controlled experiment (all students).

	average	stdev
group A	15,00	5,24
group B	14,95	3,77

group C	16,59	5,05

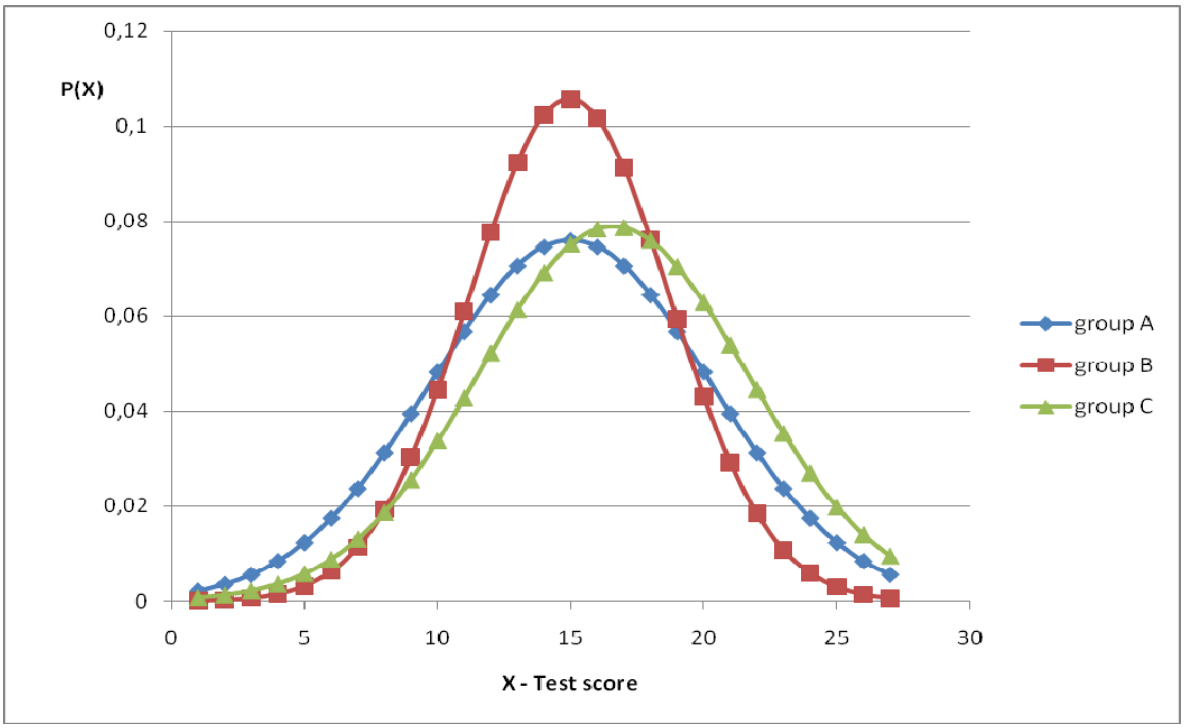

Figure 10. Normal distribution of all three groups computed from the results of the controlled experiment.

In order to find out what exactly happened, we examined the resulting score also from the point of view of student's performance using aforementioned weighted study average metric. Figure 11 and Figure 12 display score distributions for better and worse students. When considering good students (Figure 11), we can see that three bell curves are basically overlapping. This means that study texts are good enough for a good student to master a topic to some extent and neither annotations nor questions (even adaptively selected) could boost student's performance significantly – student is performing well without them anyway.

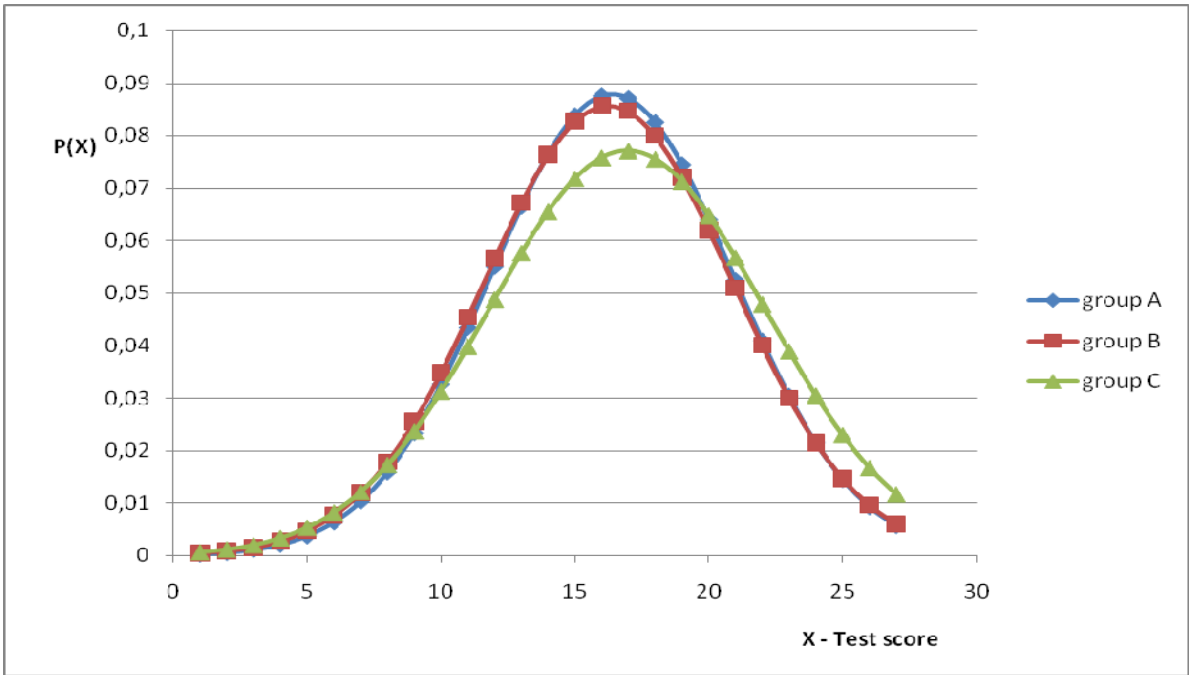

Figure 11. Normal distribution of all three groups computed from results of the controlled experiment, when taking into account only students with weighted study average below 2.16 (75th percentile and better).

However, when considering less performing students (Figure 12), we can see a distinct difference between different setups of the learning environment, with group C scoring significantly better than group A. Less performing students benefit from the additional interactive content attached to the study texts and chosen according to their current knowledge. Further, the experiment showed that carefully selected text annotations, which help to recall already explained concepts, are also useful for less performing students. We believe that this phenomenon is caused by our restriction of study time, along with the focus of the experiment on the most basic concepts of the LISP language. The annotations helped our students to save a lot of time and their repetition was sufficient to grasp the LISP basics. Questions would definitely prove their advantage against annotations when harder-to-understand concepts come into play.

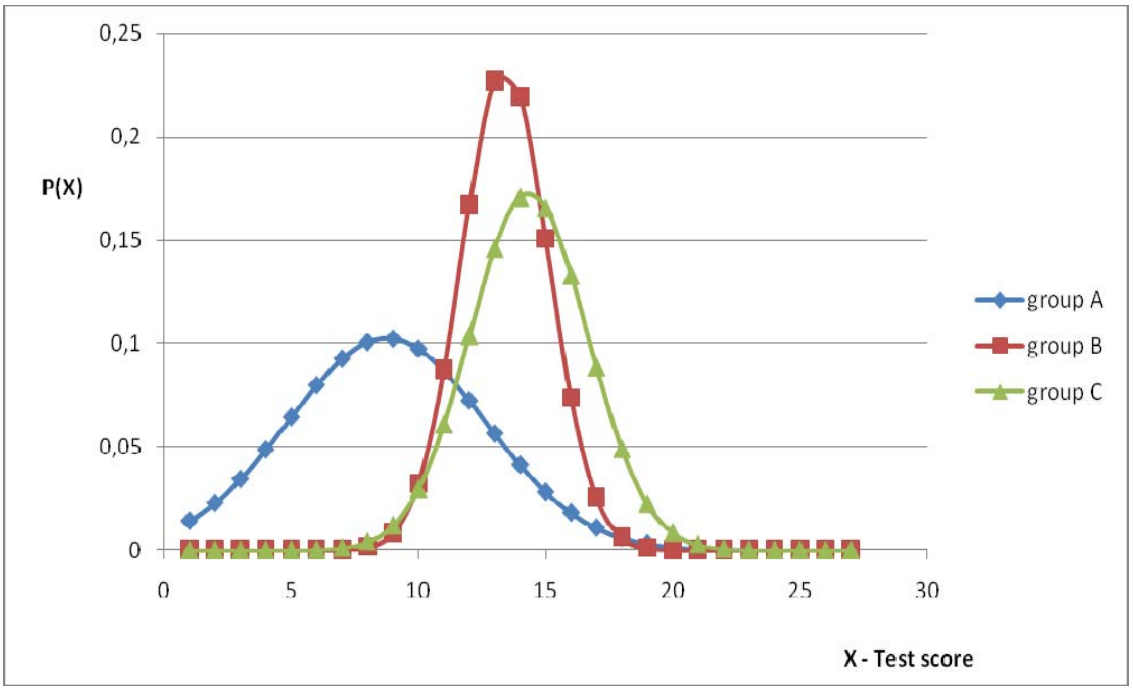

Figure 12. Normal distribution of all three groups computed from results of the controlled experiment, when taking into account only students with weighted study average above 2.0.

Overall, the second experiment was successful. It shows that adaptively selected questions and adaptive content navigation are especially useful as an additional learning resource for below-average students. We believe that even good students, who were able to learn adequately from plain texts only, would have benefited from the method if the available amount of information had been much broader, or they had been given a shorter learning time.

5 Conclusions

In this paper we have described an approach to adaptive question selection and its evaluation. We have combined the Item Response Theory with a course structure based selection as our main contribution. Our approach enables us to use adaptive testing in highly articulated domains such as programming. Furthermore, the results of the Item Response Theory can be used as a feedback for recommendations for the next topic of learning.

We conducted two experiments of very different natures. The first experiment we called uncontrolled, where we allowed (but not forced) a large number of students to prepare for an exam by studying within our educational web-based system for a virtually unlimited time. In the second experiment we gathered a smaller, but highly controlled group of motivated users, who were asked to learn as much as they could in a limited time and only by using our system. We got a valuable feedback from users during the first experiment. We found that users liked the system and found it very helpful. We also observed a remarkable improvement in overall test results compared to the results from the previous year of the procedural programming course. The only difference between the two cases was the use of Flip by students.

A second experiment showed that the impact of advanced techniques applied in e-learning cannot be generalized. We need to distinguish between good students, who (in many cases) do not need any enhancements in order to score very well, and below-average students, who tend to benefit greatly from enhancements to the basic learning materials.

Our approach to enhancing the learning experience by adaptively selected questions proves to be useful and presents a substantial benefit to learners. In future work we plan to evaluate the usefulness of inter-domain recommendations based on a shared conceptual layer of our domain model – e.g., when a student masters the “loops“ concept within a C programming language course, we do not need to give too much focus on learning objects explaining “loops“ in Java. We believe that such recommendations, apart from saving a learner’s time, can make the system seem less boring and more personalized, and thus raise the possibility that students may use it more intensively.

Apart from the inter-domain recommendations, we see several other interesting extensions of our method. One such extension is the ability to adaptively select questions and learning content either automatically or using an explicitly identified goal (or context like in (Návrat & others, 2008)). In our work, we considered only one ultimate goal – to maximize the student’s knowledge. However, we can imagine a scenario in which a student does not want to be perfect in a given topic, but simply needs to pass a next-day exam. In such a case, the student needs an adaptive selection of questions, which would not waste his time by tuning his knowledge in a certain area, but would

cover all the required topics, and ensure that the student gain the minimal required knowledge to pass the exam.

References

Bieliková, M. (2006). An adaptive web-based system for learning programming. *Int. J. Continuing Engineering Education and Life-Long Learning , 16* (1/2), pp. 122-136.

Bieliková, M., & Návrat, P. (2009). Adaptive web-based portal for effective learning programming. *Communication & Cognition, 42* (1-2), pp. 79-92.

Brusilovsky, P., & Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. P. Brusilovsky & A. Kobsa & W. Nejdl (Eds.), *The Adaptive Web*, LNCS 4321, Springer-Verlag, Berlin, Heidelberg, pp. 3-53.

Brusilovsky, P., & Peylo, C. (2003). Adaptive and Intelligent Web-based educational systems. *Int. Journal of Artifical Intelligence in Education, Special Issue on Adaptive and Intelligent Web-based Educational Systems, 13* (2-4), pp. 159-172.

Brusilovsky, P., & Sosnovsky, S. (2005). Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK. *Journal on Educational Resources in Computing , 5* (3), Article No. 6. ACM, NY, USA.

Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2004). QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In J. Nall, & R. Robson (Ed.), *Proceedings of World Conference on E-Learning, E-Learn 2004*, pp. 1806-1813.

Conejo, R., Guzmán, R., Millián, E. & others (2004). SIETTE: A Web-Based Tool for Adaptive Testing. *Int. Journal of Artifical Intelligence in Education , 14* (1), pp. 29-61.

Embertson, S., & Resise, S. (2000). *Item Response Theory for Psychologists.* New Jersey, USA: Lawrence Erlbaum.

Linacre, J. M. (2000). Computer-adaptive testing: A methodology whose time has come in. In S. Chae, U. Kang, E. Jeon, & J. M. Linacre (Ed.), *Development of Computerized Middle School Achievement Tests, MESA Research Memorandum No. 69.* Seoul, Komesa Press.

Mihál, V. & Bieliková, M. (2009). An Approach to Annotation of Web-Based Learning Texts on Programming. In *Proc. of SMAP 2009 – 4th Int. Workshop on Semantic Media Adaptation and Personalization*. San Sebastian, Spain, CS IEEE, pp. 99–104.

Návrat, P., Taraba, T., Bou Ezzeddine, A. & Chudá, D. (2008) Context Search Enhanced by Readability Index. In IFIP Series. Vol. 276: *Artificial Intelligence in Theory and Practice II*. New York, Springer Science+Business Media, LLC. pp. 373-382.

Papanikolaou K., A., Grigoriadou, M., Kornikolakis, H., & others (2003). Personalizing the interaction in a web-based educational hypermedia system: the case of inspire. *User Modeling and User-Adapted Interaction, 13* (3), 213-267, Kluwer Academic Publishers.

Radenković, S., Krdžavac, N., & Devedžić, V. (2007). A QTI Metamodel. *Proc. of Int. Multiconference on Computer Science and Information Technology*, pp. 1123-1132.

Šimko, M., & Bieliková, M. (2009). Automatic concept relationships discovery for an adaptive e-course. *Proc. of 2nd Int. Conf. on Educational Data Mining,* pp. 171-179.

Šimún, M., Andrejko, A., & Bieliková, M. (2008). Maintenance of Learner's Characteristics by Spreading a Change. In M. Kendall, & B. Samways, *IFIP International Federation for Information Processing: Learning to Live in the Knowledge Society,* Vol. 281, pp. 223-226, Springer Boston.

Sun, J. & Hsu, Y. (2007). A Study of Learners' Perceptions of the Interactivity of Web-Based Instruction. *Proc. of Human-Computer Interaction*, Part IV, HCII 2007, LNCS 4553, pp. 351–360.

Vozár, O., & Bieliková, M. (2008). Adaptive Test Question Selection for Web-based Educational System. *Proc. of SMAP 2008 - 3rd Int. Workshop on Semantic Media Adaptation and Personalization*, Prague, Czech Republic, CS IEEE Press, pp. 164-169.

Yudelson, M., & Brusilovsky, P. (2005). NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples. In C.-K. Looi, McCalla G., B. Bredeweg, & J. Breuker (Ed.), *International Conference on Artificial Intelligence in Education, AIED 2005,* pp. 710-717, IOS Press.