A Scalable Solution for Adaptive Problem Sequencing and Its Evaluation

Amruth Kumar

Ramapo College of New Jersey,
Mahwah, NJ 07430, USA

Abstract. We propose an associative mechanism for adaptive generation of problems in intelligent tutors. Our evaluations of the tutors that use associative adaptation for problem sequencing show that 1) associative adaptation targets concepts less well understood by students; and 2) associative adaptation helps students learn with fewer practice problems. Apart from being domain-independent, the advantages of associative adaptation compared to other adaptive techniques are that it is easier to build and is scalable.

Keywords: Programming tutor, Adaptive Problem Sequencing, Evaluation.

1 Introduction

Learning is most effective when it is adapted to the needs of the learner [4]. In a tutor, various aspects can be adapted to the needs of the learner, including the problem sequence, feedback type, feedback amount, and the level of detail of the open student model. Vector spaces [17] and learning spaces [10] are the popularly used mechanisms for adaptation of problem sequence in tutors. These approaches are domain-independent. But building vector spaces and learning spaces is labour-intensive. Moreover, adding new problems or concepts to a vector space or learning space entails significant redesign of the space.

Alternatively, we propose a scalable solution for adaptive problem sequencing. In this approach, we index problems by concepts. We specify proficiency criteria for each concept in the domain model and maintain the student model as an overlay of the domain model. We use simple algorithms to select the next concept for the student, and the next problem for the concept.

In this paper, we will first describe the tutors for which we developed associative adaptation. We will describe the domain and overlay student models used in these tutors. Next, we will describe the associative mechanism for adaptive generation of problems. Finally, we will describe evaluations that support our claims that associative adaptation targets concepts less well understood by students, and it helps students learn with fewer practice problems.

2 Programming Tutors

We have been developing web-based tutors to help students learn C/C++/Java/C# programming language concepts by solving problems. To date, we have developed

V. Wade, H. Ashman, and B. Smyth (Eds.): AH 2006, LNCS 4018, pp. 161 – 171, 2006.

tutors on expression evaluation, selection statements, loops, pointers in C++, parameter passing mechanisms, scope concepts and their implementation, and C++ classes. Our tutors target program analysis (solving expressions, predicting the output of programs and debugging programs) in Bloom's taxonomy [3] in contrast to program synthesis (writing a program), which has been the traditional focus of intelligent tutors (e.g., LISP Tutor [16], ELM-ART [18]).

Consider the tutor on selection statements. The tutor presents a program that involves one or more selection (if/if-else) statements, and asks the learner to predict the output of the program. The tutor grades the learner's answer. In addition, it provides explanation of the step-by-step execution of the program as part of its feedback [12].

Limited problem set has been recognized as a potential drawback of encoding a finite number of problems into a tutor [13]. Therefore, our web-based tutors generate problems as instances of parameterized templates. Each problem template is indexed by one or more concepts, and the template is used to generate problems for only these concepts.

2.1 The Domain and Student Models

We use a single unified domain model for all our programming tutors. This domain model is the concept map of the programming domain, i.e., a taxonomic map with topics as nodes, and *is-a* and *part-of* relationships as arcs. In this model, we associate two measures with each node to determine whether the student has mastered the corresponding concept:

- M_1 - The minimum number of problems the learner must solve on that concept. Typically, $M_1 = 2$.
- M_2 - The percentage of problems that the learner must solve correctly on a concept in order to be considered proficient in it. Typically, $M_2 = 60\%$.

We use an overlay of the domain model as our cognitive student model. But, instead of saving M_1 and M_2 in the student model, we save {G,A,R,W,M} to record the student's progress - the number of problems generated (G), attempted (A), correctly solved (R), incorrectly solved (W) and missed (M) by the student on that concept. Currently, our tutors use two inequalities to interpret this data and determine whether a student has mastered a concept: $A \geq M_1$ and $R / A \geq M_2$.

3 Associative Adaptation of Problem Generation

Recall that we use a concept map as our domain model, associate proficiency criteria with the concepts in the domain model, use an overlay student model, associate the student's progress statistics with the concepts in the student model, and index problem templates by concepts. We will now present the algorithm for associative adaptation of problem generation.

The Algorithm

1. Let the set of all the concepts in the topic be $C = \{C_1, C_2, \ldots, C_m\}$, where $C_1, C_2, \ldots, C_m$ are individual concepts.
2. For each concept C_i, extract all the problem templates that match the concept. Let the resulting set of templates be $T_i = \{T_{i1}, T_{i2}, \ldots, T_{iq}\}$, where $T_{i1}, T_{i2}, \ldots, T_{iq}$ are individual templates that match C_i.
3. Identify the list of concepts that the learner has not mastered. Let this set be $C_s = \{C_1, C_2, \ldots, C_n\}$, $n \leq m$. If the set C_s is empty, the student has mastered this topic, exit.
4. Select the next concept C_j from the set C_s.
5. Select the next problem template T_{jk} from the set of templates T_j corresponding to the concept C_j and generate the next problem as an instance of the template.
6. After the learner has attempted the problem, update {G,A,C,W,M} for the concept C_j in the student model, as well as any other concept affected by the template T_{jk}. Repeat from Step 3.

We define **persistence** p as the maximum number of problems a tutor generates back to back on a concept before moving on to the next concept. Persistence p affects problem generation as follows:

- $p = 1$ means that the concept is changed from one problem to the next. This may not reinforce learning due to rapid switching of concepts.
- $p = 2$ or 3 helps reinforce learning since the tutor presents 2-3 problems back to back on a concept.
- $p > 3$ may make the tutor predictable and boring. The student may begin guessing the correct answer to problems, which would negatively affect learning.

Sub-algorithm for Step 4: Given the last concept was C_i, the algorithm to select the next concept is as follows:

1. If C_i has been mastered, return the next concept C_{i+1} in the list. If $i + 1 > n$, the number of concepts not yet mastered, set $i = 1$, and return C_1
2. If p problems have been generated back to back on the concept C_i, return C_{i+1}. If $i + 1 > n$, set $i = 1$, and return C_1
3. Else, return C_i.

Sub-algorithm for Step 5: We use the round-robin algorithm for selecting the next problem template for a concept. If the last template used by the tutor for a concept is T_{ij}, the next time it revisits the concept, it uses the template $T_{i,j+1}$. If $j + 1 > q$, $j = 1$.

This associative algorithm is independent of the domain: it can be used for any domain wherein 1) appropriate concepts can be identified; 2) the student model is maintained in terms of concepts; and 3) problem templates are indexed by concepts. This associative adaptation algorithm has several advantages over vector spaces [17] and learning spaces [10] that have been popularly used to implement adaptation:

- The associative system is easier to build - there is no need to place all the problem templates in an exhaustive vector or learning space.

- The associative system is scalable - We can add new concepts and problem templates to the tutor without affecting any existing templates and/or modifying the vector/learning space.

The learning path of individual learners in the problem space is determined by matching the problem templates in the template knowledge base with the un-mastered concepts in the student model. An associative system automatically supports all the learning paths - even those that may not have been explicitly modelled in a vector or learning space. Therefore, the resulting adaptation is more flexible. Associative adaptation is similar to the adaptation mechanism used in ActiveMath [14] to determine the information, exercises, and examples presented to the learner, and the order in which they are presented.

3.1 An Example

Consider the tutor on arithmetic expressions. For this example, we will consider only the following concepts: correct evaluation and precedence of +, * and / operators. Let the following table represent the initial student model, where m / n denotes that the student has correctly solved m out of the n problems (s)he has attempted on the concept:

Student Model	+	*	/
Correct Evaluation	2/2	1/2	0/2
Precedence	0/2	2/2	1/2

Assuming $M_1 = 2$ and $M_2 = 60\%$, the student has not yet mastered the following concepts: correct evaluation of * and /, and precedence of + and /. Assume that the next problem template for the correct evaluation of * yields the expression 3 + 4 * 5, and the student correctly solves the entire expression. Since the expression includes the correct evaluation and precedence of + and * operators, the student gets credit for all four concepts:

Student Model	+	*	/
Correct Evaluation	3/3	2/3	0/2
Precedence	1/3	3/3	1/2

Since the student just mastered the correct evaluation of *, the tutor considers the next concept, viz., correct evaluation of /. Assume that the next problem template for the correct evaluation of / yields the expression 5 + 10 / 4, and the student correctly solves the entire expression. Since the expression includes the correct evaluation and precedence of + and / operators, the student gets credit for all four concepts:

Student Model	+	*	/
Correct Evaluation	4/4	2/3	1/3
Precedence	2/4	3/3	2/3

If persistence p = 2, the tutor generates a second problem on the correct evaluation of /. Note that even if the student solves the second problem correctly, correct evaluation of / will remain un-mastered (2/4 < 60%). Even so, since persistence p = 2, the tutor will pick another concept for the subsequent problem and return to the correct evaluation of / later, in a round-robin fashion.

Note that a student may master a concept without attempting any problem on it, e.g., precedence of / operator in the above example. A student could revert from mastered to un-mastered state by solving subsequent problems incorrectly.

4 Evaluation of the Adaptive Tutor

Numerous evaluations have shown that our tutors help students learn, e.g., in one controlled test comparing a tutor with a printed workbook, improvement in learning with the tutor was larger and statistically significant compared to improvement with the printed workbook [7]. Evaluations have also shown that the explanation of step-by-step execution provided as feedback by the tutors is the key to the improvement in learning [12]. We wanted to evaluate whether associative adaptation helped improve the effectiveness of the tutors. The hypotheses for our evaluations were:

1. Associative adaptation targets the concepts less well understood by students.
2. Associative adaptation helps students learn with fewer problems.

In spring and fall 2005, we evaluated our tutor on selection statements. Students used the tutor on their own time, as non-credit-bearing assignment in a course.

Protocol: We used the pre-test-practice-post-test protocol for evaluation of the tutor:

- **Pre-test** –The pre-test consisted of a predetermined sequence of 21 problems covering 12 concepts. Students were allowed 8 minutes for the pre-test. The tutor administered the pre-test. The tutor did not provide any feedback during the pre-test.

  The tutor used the pre-test to initialize the student model, as proposed by earlier researchers (e.g., [1,6]). However, the test was not adaptive as proposed by others (e.g., [2, 15]), because we wanted to compare the pre-test score with the score on a similarly constructed post-test to evaluate the effectiveness of the adaptive tutor. Stereotypes [1,8] and schema-based assessment [9] are some of the other techniques proposed in literature to initialize the student model.
- **Practice** – The tutor provided detailed feedback for each problem. The tutor used the associative adaptation algorithm to present problems on only those concepts on which the student had not demonstrated mastery during the pre-test. It used persistence = 2, M_1 = 2, and M_2 = 60%. The practice session lasted 15 minutes or until the student learned all the concepts, whichever came first.
- **Post-test** –The post-test consisted of 21 problems, covering concepts in the same order as the pre-test. Students were allowed 8 minutes for the post-test. The tutor administered the post-test. It did not provide any feedback during the post-test. The test was not adaptive.

The three stages: pre-test, practice and post-test were administered by the tutor back-to-back, with no break in between. The students did not have access to the tutor before the experiment.

Analysis: In Table 1, we have listed the average and standard deviation of the number of problems solved, the raw score, and the score per problem on the pre-test, practice and post-test for the 22 students who used the tutor in spring 2005. Note that the raw score increased by 62% (from 6.39 to 10.33) from the pre-test to the post-test. However, the number of problems solved by the students also increased by 33% (from 10.55 to 14.05), and both these increases were statistically significant (2-tailed $p < 0.05$). In order to factor out the effect of the increase in the number of problems on the increase in the raw score, we calculated the average score per problem. The average score per problem also increased by 44% from pre-test to post-test and this increase was statistically significant.

Table 1. Results from the spring 2005 evaluation of the tutor on selection statements

Spr. 05	Pre-Test			Practice			Post-Test		
N = 22	**Prob.**	**Score**	**Ave**	**Prob.**	**Score**	**Ave**	**Prob.**	**Score**	**Ave**
Average	10.55	6.39	0.59	17.27	9.41	0.59	14.05	10.33	0.75
Std-Dev	3.54	3.86	0.27	10.56	4.71	0.25	4.10	4.10	0.22
***p*-value of pre-post difference**							0.0000	0.0000	0.0007

The above results do not take into account the following confounds:

- Recall that the practice provided by the adaptive tutor was limited to 15 minutes. This meant that the students often ran out of time and did not get practice on all 12 concepts.
- It was likely that students already knew some of the concepts during the pre-test – learning of these concepts could not be credited to the use of the tutor.

In order to take these into consideration, we re-analyzed the data by concepts instead of problems. For each student, and each concept, we calculated the problems solved and average score on the pre-test, practice and post-test. Next, we grouped the concepts for each student into four categories:

- **Discarded Concepts:** Concepts on which the student did not attempt any problem during the pre-test or during the post-test because of the time limit on the tests;
- **Known Concepts:** Concepts on which the student demonstrated mastery during the pre-test, i.e., attempted $M_1 = 2$ problems and solved $M_2 = 60\%$ of the problems correctly;
- **Control Concepts:** Concepts on which the student solved problems during the pre-test and the post-test, but did not demonstrate mastery during the pre-test and *did not solve any problems during practice* due to the time limit imposed on the practice session – this provided the datum for comparison of test data.
- **Test Concepts:** Concepts on which the student solved problems during the pre-test and the post-test, but did not demonstrate mastery during the pre-test and *did solve problems during practice* – since the tutor provides feedback during practice to help the student learn, data on test concepts could prove or refute the effectiveness of using the tutor for learning.

Table 2. Classifying student concepts as discarded, known, control or test

Problems Solved	Pre-Test	Practice	Post-Test
Discarded	0	*	*
Discarded	*	*	0
Known	$A \geq M_1$ & $R / A \geq M_2$	*	*
Control	+	0	+
Test	+	+	+

The four types of student concepts are summarized in Table 2, where * represents 0 or more problems solved, and + represents 1 or more problems solved. For our analysis, we ignored the discarded student concepts since they represented incomplete data. We ignored the known student concepts – the tutor cannot be credited for the learning of the concepts that the students already knew during the pre-test. On the remaining student concepts, since each student served as part of both control group (on concepts for which the student did not get practice) and test group (on concepts for which the student did get practice), we consider this a within-subjects design.

In Table 3, we have listed the average and standard deviation of the number of problems solved and the average score per problem on the pre-test, practice and post-test for the 56 control student concepts and the 135 test student concepts as defined above. Note that the average score of the control group remained steady whereas the average score of the test group increased by 48% and this increase was statistically significant. This supports the results from our prior evaluations that practicing with the tutor promotes learning.

Table 3. Control versus Test Student Concepts from spring 2005 evaluation of Selection Tutor

Spring 05	Pre-Test		Practice	Post-Test		*p*-value	
	Prob.	**Ave**	**Problems**	**Prob.**	**Ave**	**Prob.**	**Ave**
Control (N = 56 student-concepts)							
Average	1.02	0.88	0	1.11	0.87	0.02	0.68
Std-Dev	0.13	0.30	0	0.31	0.31		
Test (N = 135 student-concepts)							
Average	1.07	0.46	1.83	1.35	0.68	0.000	0.000
Std-Dev	0.26	0.47	1.14	0.48	0.43		
***p*-value**	0.05	0.000		0.000	0.000		

Note that there is a statistically significant difference between the control and test groups on the number of problems solved and the average score on the pre-test. The average score of the test group of student concepts is significantly lower than that of the control group of student concepts. This supports our hypothesis that associative adaptation in our tutor targets the concepts less well understood by students.

Finally, we conducted a repeated measures one-way ANOVA on the average score, with the treatment (adaptive practice versus no practice) as between-subjects factor and pretest-post-test as the repeated measure. Our findings were:

- There was a significant main effect for pre-test versus post-test [$F(1,189) = 7.391$, $p = 0.007$] - post-test scored significantly higher than the pre-test.
- There was a significant interaction between the treatment (adaptive practice versus no practice) and time repeated measure [$F(1,189) = 10.211$, $p = 0.002$]: while the average score with no practice stayed steady, with adaptive practice, it showed a significant increase.

We repeated our evaluation of the tutor in fall 2005. In Table 4, we have listed the average and standard deviation of the number of problems solved, the raw score, and the score per problem on the pre-test, practice and post-test for the 16 students who used the tutor. Note that the raw score increased by 94% and the number of problems solved by the students increased by 53% from pre-test to post-test, and both these increases were statistically significant (2-tailed $p < 0.05$). The average score per problem also increased by 23% from pre-test to post-test and this increase was statistically significant.

Table 4. Results from the fall 2005 evaluation of the tutor on selection statements

Fall 05	**Pre-Test**			**Practice**			**Post-Test**		
N = 16	**Prob.**	**Score**	**Ave**	**Prob.**	**Score**	**Ave**	**Prob.**	**Score**	**Ave**
Average	7.69	5.00	0.66	15.0	11.29	0.76	11.75	9.72	0.81
Std-Dev	3.89	2.96	0.26	3.92	3.49	0.17	3.96	4.28	0.20
***p*-value of pre-post difference**							0.0002	0.000	0.003

When we analyzed the data by student concepts instead of problems, and divided the set of student concepts into control and test groups as described earlier, we obtained the results in Table 5. On control student concepts, the average changed from 0.81 to 0.76 from the pre-test to the post-test, and the change was not statistically significant ($p = 0.55$). On test student concepts, the average changed from 0.61 to 0.86, and the change was statistically significant ($p = 0.0000$). Once again, this supports the results from our prior evaluations that the tutors promote learning.

Table 5. Control versus Test Student Concepts from fall 2005 evaluation of Selection Tutor

Fall 05	**Pre-Test**		**Practice**	**Post-Test**		***p*-value**	
	Prob.	**Ave**	**Problems**	**Prob.**	**Ave**	**Prob.**	**Ave**
Control (N = 26 student-concepts)							
Average	1.15	0.81	0	1.46	0.76	0.002	0.55
Std-Dev	0.37	0.35	0	0.51	0.40		
Test (N = 87 student-concepts)							
Average	1.00	0.61	1.55	1.15	0.86	0.0000	0.0000
Std-Dev	0	0.47	1.20	0.36	0.31		
***p*-value**	0.04	0.02		0.006	0.23		

Once again, note that there is a statistically significant difference between the control and test groups on the number of problems solved and the average score on the pre-test. The average score of the test group of student concepts is significantly lower than that of the control group of student concepts. This once again supports our hypothesis that associative adaptation in our tutor targets the concepts less well understood by students.

We conducted a repeated measures ANOVA on the percentage of problems solved correctly, with the treatment (adaptive practice versus no practice) as between-subjects factor and pretest-post-test as repeated measure. Our findings were:

- The main effect for pre-test versus post-test was tending to statistical significance [$F(1,111) = 3.45$, $p = 0.066$] - post-test scored higher than pre-test.
- There was a significant interaction between the treatment (adaptive practice versus no practice) and time repeated measure [$F(1,111) = 7.837$, $p = 0.006$]: while average score with no practice declined modestly, with adaptive practice, it showed a significant increase.

In fall 2004, we evaluated *for* and *while* loop tutors. We used a within-subjects design: the same group of students used the non-adaptive version of the tutor on *while* loops one week, and the adaptive version on *for* loops the next week. In the non-adaptive version, the tutor presented problems for all the concepts, regardless of the learning needs of the student, in a round-robin fashion, with p = 3. Table 6 lists the average on the pre-test and post-test for the non-adaptive and adaptive versions of the tutor. One-way ANOVA analysis showed that the difference from the pre-test to the post-test was statistically significant in both the groups.

Table 6. Evaluation of non-adaptive versus adaptive versions of loop tutors – fall 2004

Average correctness of answers	Pre-Test	Post-Test	Change	Significance
Without adaptation (N = 15)				
Average	0.47	0.65	0.17	p = 0.014
Standard Deviation	0.24	0.20	0.24	
With adaptation (N = 25)				
Average	0.55	0.69	0.14	p = 0.0002
Standard Deviation	0.21	0.20	0.16	

Table 7. Problems Solved by the Control and Experimental Groups during 15-minute Practice

Problems Solved	Non-Adaptive Group	Adaptive Group	Statistical Sig.
Minimum	28	1	
Maximum	86	60	
Average	45.80	24.22	p= 0.00017
Std-Dev	15.44	14.56	

However, students solved far fewer problems during practice with the adaptive tutor than with the non-adaptive tutor, and this difference was statistically significant ($p < 0.05$) - the minimum, maximum and average number of problems solved by the two groups during practice is listed in Table 7. Given that the improvement in learning was similar for both the groups, this supports our hypothesis that associative adaptation helps students learn with fewer practice problems. Our results are in accordance with earlier results in computer-aided testing, where adaptive systems were shown to more accurately estimate a student's knowledge, and with fewer questions than non-adaptive systems [2, 19]. For this evaluation, we did not consider the time spent by the students on practice since all the students were required to practice for 15 minutes with the non-adaptive (control) tutor.

5 Conclusions

We proposed an associative mechanism for adaptive generation of problems in web-based intelligent tutors. Our evaluations show that:

1. Associative adaptation targets concepts less well understood by students - the average pre-test score on the concepts targeted by adaptation is significantly lower than the average on the concepts not targeted by adaptation.
2. A tutor with associative adaptation helps students learn with significantly fewer practice problems than a non-adaptive tutor.

Associative adaptation is easier to build and is scalable. Unlike vector spaces [17] and learning spaces [10], there is no need to exhaustively enumerate and organize all the problem templates. New concepts and problem templates can be added to the tutor without affecting any existing templates and/or modifying the previously constructed vector/learning space. This feature permits incremental development of tutors, which is invaluable when developing tutors for large domains.

Acknowledgements

Partial support for this work was provided by the National Science Foundation's Educational Innovation Program under grant CNS-0426021.

References

[1] Aimeur, E., Brassard, G., Dufort, H., and Gambs, S. CLARISSE: A Machine Learning Tool to Initialize Student Models. S. Cerri, G. Gouarderes, F. Paraguacu (eds.), Proc. of ITS 2002, Springer (2002). 718-728.
[2] Arroyo, I., Conejo, R., Guzman, E., & Woolf, B.P. An Adaptive Web-Based Component for Cognitive Ability Estimation., Proc. of AI-ED 2001, IOS Press (2001). 456-466.
[3] Bloom, B.S. and Krathwohl, D.R.: Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain, New York, Longmans, Green (1956).
[4] Bloom, B.S.: The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher, Vol 13. 1984. 3-16.
[5] Brown, J.S. and Burton, R.R. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, Vol 2 (1978). 155-191.
[6] Czarkowski, M. and Kay, J. Challenges of Scrutable Adaptivity. U. Hoppe, F. Verdejo and J. Kay (eds.), Proc. of AI-ED 2003, IOS Press (2003). 404-406.

[7] Dancik, G. and Kumar, A.N., A Tutor for Counter-Controlled Loop Concepts and Its Evaluation, Proceedings of Frontiers in Education Conference (FIE 2003), Boulder, CO, 11/5-8/2003, Session T3C.
[8] Kay, J.: Stereotypes, Student Models and Scrutability. Proc. of ITS 2000. G. Gauthier, C. Frasson and K. VanLehn (eds.). Springer (2000). 19-30.
[9] Kalyuga, S. Rapid Assessment of Learner's Knowledge in Adaptive Learning Environments, Proc. of AI-ED 2003, IOS Press, (2003). 167-174.
[10] Kurhila, J., Lattu, M., and Pietila, A. Using Vector Space Model in Adaptive Hypermedia for Learning. Proc. of ITS 2002, Springer (2002). 129-138.
[11] Kumar, A.N. Using Enhanced Concept Map for Student Modeling in a Model-Based Programming Tutor. International FLAIRS conference on Artificial Intelligence, Melbourne Beach, FL, May 11-13, 2006,
[12] Kumar, A.N., Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors, Technology, Instruction, Cognition and Learning (TICL) Journal, to appear.
[13] Martin, B. and Mitrovic, A. Tailoring Feedback by Correcting Student Answers. Proc. of ITS 2000. Springer (2000). 383-392.
[14] Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M. and Ullrich, C. ActiveMath: A Generic and Adaptive Web-Based Learning Environment. International Journal of Artificial Intelligence in Education, Vol 12 (2001). 385-407.
[15] Millan, E., Perez-de-la-Cruz, J.L., and Svazer, E. Adaptive Bayesian Networks for Multilevel Student Modeling. Proc. of ITS 2000. Springer (2000), 534-543.
[16] Reiser, B., Anderson, J. and Farrell, R.: Dynamic student modelling in an intelligent tutor for LISP programming, Proc. of IJCAI 1985. Los Altos CA (1985).
[17] Salton, G., Wong, A. and Yang, C.S. A Vector Space Model for Automatic Indexing. Communications of the ACM, Vol. 18(11), (1975). 613-620.
[18] Weber, G. and Brusilovsky, P. ELM-ART: An Adaptive Versatile System for Web-Based Instruction. International Journal of Artificial Intelligence in Education, Vol 12 (2001). 351-384.
[19] Wainer, H. Computerized Adaptive Testing: A Primer. Lawrence Erlbaum Associates. Hillsdale, NJ, 1990.