TEACHING AI SEARCH ALGORITHMS IN A WEB-BASED EDUCATIONAL SYSTEM

Foteini Grivokostopoulou and Ioannis Hatzilygeroudis
University of Patras, School of Engineering
Department of Computer Engineering & Informatics
26500 Patras, Hellas (Greece)

ABSTRACT

In this paper, we present a way of teaching AI search algorithms in a web-based adaptive educational system. Teaching is based on interactive examples and exercises. Interactive examples, which use visualized animations to present AI search algorithms in a step-by-step way with explanations, are used to make learning more attractive. Practice exercises, which are interactive exercises where immediate feedback is given when a student makes an error, but further help is optional, are also used. So, the student can try by him/herself to correct the error or ask for help from the system. Finally, the student can take a test consisting of assessment exercises, which are interactive, but no help is provided. The result of the test determines the student's knowledge level. Evaluation of the system through a pre-test/post-test and experimental/control group method gave very promising results about learning capabilities of the method. Also, results of a questionnaire show that the majority of the students liked the system very much.

KEYWORDS

Web-based interactive system, interactive e-learning, AI search algorithms, adaptive system

1. INTRODUCTION

Numerous e-learning systems have been developed for education during last decades. A requirement of e-learning systems is their ability to provide personalization that is, interact with learners by taking into account their differences in knowledge levels, learning styles, preferences, goals and so on (Wade and Ashman 2007). Intelligent Tutoring Systems (ITSs) constitute a popular type of intelligent educational systems. They perform their tasks mainly based on Artificial Intelligence (AI) methods. Employed AI methods provide the representation and reasoning basis needed to accomplish the goals of ITSs. More specifically, AI techniques are used to represent theoretical and practical knowledge regarding the teaching subject. Such knowledge provides the foundation for developing mechanisms for customizing the learning process, given information related to learners. AI techniques also represent information gathered from learners' interaction with the system. ITSs are able to intelligently analyze learner responses to presented problems and questions. They are also able to intelligently analyze questions posed by learners to the system itself, regarding the teaching subject. Such intelligent analysis enables provision of intelligent support/assistance/feedback concerning specific parts of the teaching subject, for which learner knowledge is assessed to be incomplete and/or misconceived. ITSs have been used with great success in many challenging domains to offer individualized learning to the students and have demonstrated remarkable success in helping students learn challenging content and strategies (Woolf 2009). Adaptive Educational Hypermedia Systems (AEHSs) are also systems that offer personalized education. They are specifically developed for hypertext environments such as the WWW (Brusilovsky et al 1998). Enhancing AEHSs with aspects and techniques from ITSs creates another type of personalized education systems: Adaptive and Intelligent Educational Systems (AIESs) (Brusilovsky and Paylo 2003).

In an Artificial Intelligence (AI) curriculum, a fundamental topic is "AI search algorithms". It is vital for students to get a strong understanding of the way the algorithms work and also of their implementation on various search problems. However, many students have particular difficulties in understanding and implementing AI search algorithms. Students often have difficulties in understanding algorithms, which are explained to them on the blackboard.

In this paper, we present how an adaptive and intelligent web-based system assists in teaching/learning search algorithms in the context of an AI course. The system provides interactive exercises and examples to assist students to learn and practice by graphically implementing blind and heuristic search algorithms. The system provides a step-by-step guidance and assistance during the application of the algorithms.

The rest of the paper is organized as follows. In Section 2 related work is presented. Section 3 presents the domain knowledge structure and the user interface of the system. Section 4 presents the teaching method and learning materials. Section 5 presents an evaluation of the teaching method and user interface. Finally, Section 6 concludes the paper and provides a direction for future work.

2. RELATED WORK

There are systems developed to assist teaching and learning in the context of AI curriculum. In (Hatzilygeroudis et al 2005, Hatzilygeroudis et al 2006) a web-based intelligent educational system in the context of an AI course that uses AI techniques for teaching and assessing learners is presented. PATHFINDER (Sanchez-Torrubia et al 2009) is a tool for actively learning Dijkstra's algorithm. The highlighting new feature provided by this tool is an animated algorithm visualization panel. It shows, on the code, the current step the student is executing and also where there is a user's mistake within the algorithm running. TRAKLA2 (Malmi et al 2004) is a system for automatically assessing visual algorithm simulation exercises. TRAKLA2 provides automatic feedback and grading and allows resubmission, which means that students can correct their errors in real time. In (Kordaki et al 2007), an educational system, called Starting with Algorithmic Structures (SAS), designed for teaching the concept of algorithm and those of basic algorithmic structures to beginners is presented. Students come across the implementation of algorithms in real life scenarios and the system offers feedback to correct their answers. Also, in (Naser 2008), a visualization tool for teaching searching algorithms is presented. However, none of the above works

3. ARTIFICIAL INTELLIGENCE TEACHING SYSTEM

We developed the Artificial Intelligence Teaching System (AITS), to assist learning and teaching in the context of the "Artificial Intelligence" course in our Department. It is an adaptive and intelligent web-based educational system that teaches AI aspects and uses AI techniques for personalized learning and assessment of the students. Among others, it teaches AI search algorithms. We present in this paper the way it does it at an educational level.

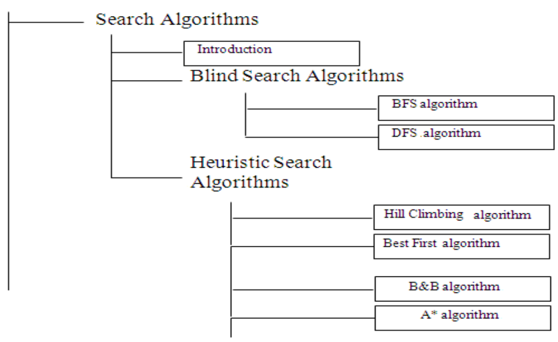

Figure 1. Part of the Tree of the AI Search Algorithms Domain.

3.1 Domain Knowledge Structure

AITS uses a tree-like structure to represent the domain knowledge related to the AI search algorithms. In Figure 1, a part of the tree structure of the AI search algorithms domain is presented. The subject itself is the root of the tree. The subject is divided in sections and the sections into subsections. Each sub-section deals with a number of concepts, which are the leaves of the tree. Subsections may have some concepts in common. The domain knowledge tree, described above, is displayed, as far as the subsections level, at the navigation area, at the left side of the user interface. From that tree the student can choose a learning goal (subsection). Each subsection corresponds to a learning page. Each learning page deals with a number of concepts. So, the concept pages constitute the teaching material of the domain.

3.2 User Interface

A student, through the user interface, initially subscribes to the AITS. After subscription, the student can, at any time, enter the system with his/her personal data. The interface of AITS is illustrated in Figure 2. It consists of two main areas: *navigation area* and *content area*, and also an *info bar*, which accommodates various information. This information concerns the username with which the student is connected to the system, the residence time of the system and the current date and time. Also, it offers to the student facilities for changing personal elements, changing the learning style, view statistics and general information about the system, annotation capabilities of the system, and logout from the system. Especially for the statistics, the student has the opportunity to see all the answers he/she gave and his/her knowledge level for each subsection, subject, and concept.

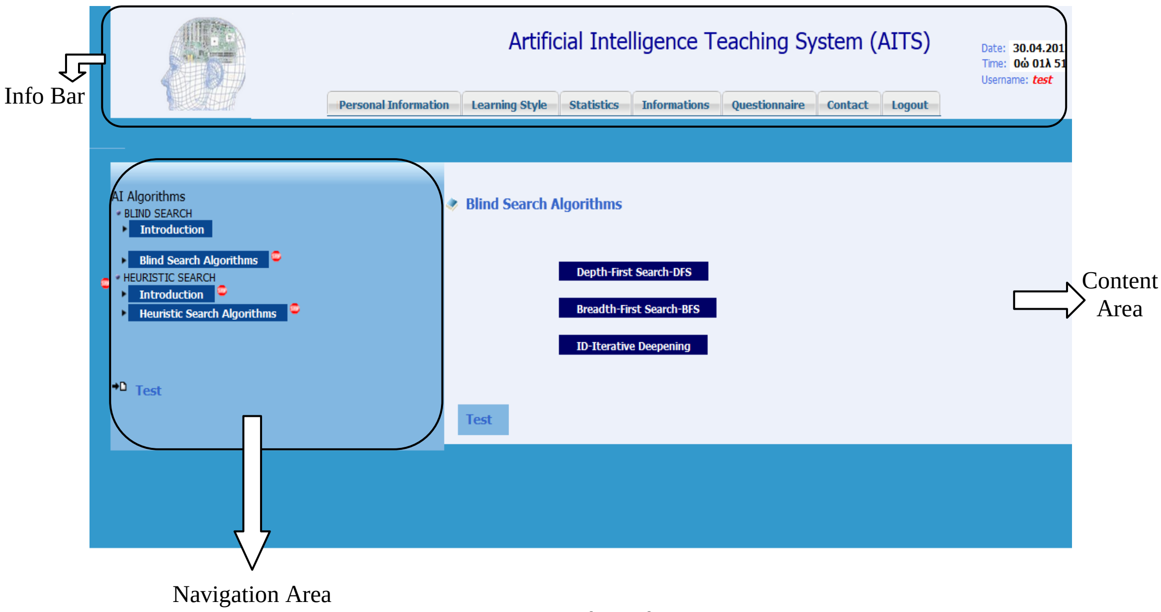

Figure 2. User Interface of the System

4. TEACHING METHOD AND LEARNING MATERIALS

Teaching is based on the traditional theory-examples-exercises paradigm (although the students can follow their own way). That is, for each topic, the theory is first presented. Then, some examples are given. Finally, the student is called to do some exercises. Theory consists in presenting a number of concepts. Those concepts are presented in a simple-to-complex way. That is, the simple concepts are presented first and the complex concepts (that require the knowledge of one or more simpler concepts) are presented afterwards. This is depicted in the ordered list presented to the user, which however is randomly accessible. Furthermore, the student can review a previous concept at any time. He/She is also not forced to follow the system's way of teaching, but can make his/her own choices for studying a learning page. Any time a student has finished the study of a particular learning page, he/she can take the corresponding concept-level test.

The system offers theory, examples and exercises about AI search algorithms. More specifically, the system offers *theory* about blind search and heuristic search algorithms (breadth-first, depth-first, hill-climbing, branch-and-bound, A* etc), *interactive examples* and two types exercises: *practice exercises* and *assessment exercises*. Practice exercises are interactive exercises that are equipped with hints and guidance during the learning sessions, aiming to help the student in learning AI search algorithms. On the other hand, assessment exercises are interactive exercises that are used to examine the student's progress and comprehension. For example, a test on an AI search algorithm concept consists of a number of interactive assessment exercises.

After a student enters the system can study the theory about blind search and heuristic search algorithms and see interactive examples on the selected subjects. The student can select to interactively deal with a practice exercise to be helped in learning an algorithm. Finally, he/she can take a test consisting of assessment exercises to see what his/her knowledge level is.

4.1 Interactive Examples

The system provides interactive examples of different types on AI search algorithms. Interactive examples necessitate the step-by-step application of an algorithm. Each example consists of a tree or a graph with an explanation for each step of an algorithm. By clinking a button, the next step of the algorithm and corresponding details of the current step are shown. The examples are using animation to present the application of algorithms. In Figure 3, an interactive example on depth-first search (dfs) algorithm is presented.

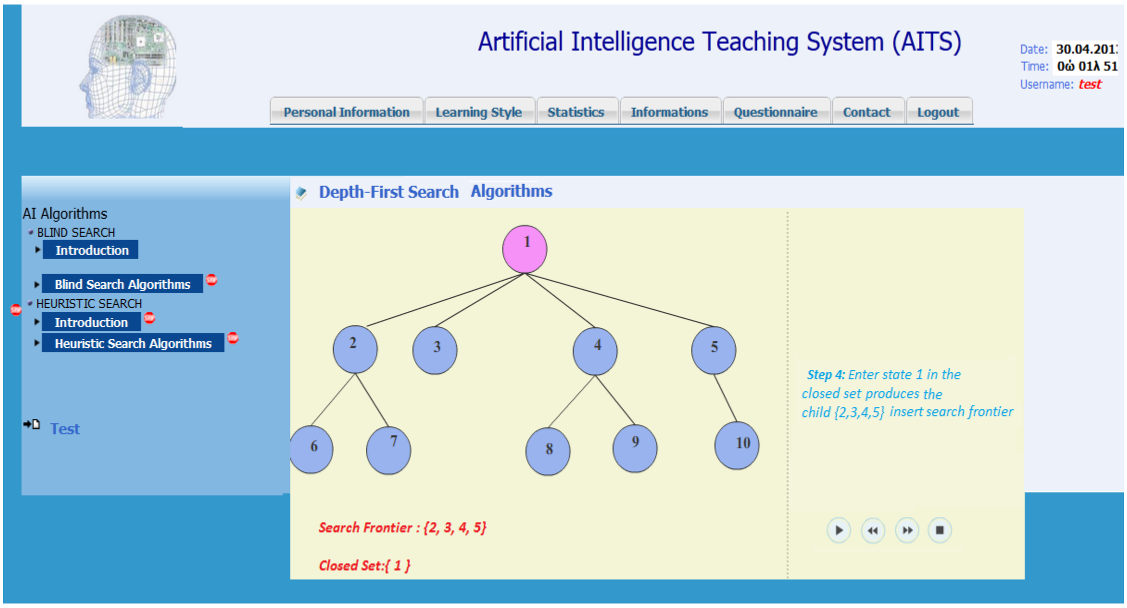

Figure 3. An interactive example on depth-first algorithm

4.2 Practice Exercises

Practice exercises are interactive exercises and their purpose is to assist students in understanding the algorithms through implementing AI search algorithms in an interactive way. A practice interactive exercise presents a graph or a tree and the student is called to reach a specific *goal node* starting from its *root* following an AI search algorithm. The student selects a *node* by clinking on it and the system provides guidance and feedback at student’s actions during his/her interaction. The student can request for system assistance when stuck, by clicking on a help button. Then, the system provides feedback on what should do next and also cues regarding the next to visit/select node.

More specifically, as soon as a student selects a node in a practice exercise, the system acts as follows:

1. If the node choice is correct (in which case node color changes to green), then wait for the student to select the next node and go to 1.
2. If it is incorrect (in which case node color changes to red)
   2.1 Provide proper feedback, i.e. adapted to the error made.
   2.2 If the student clicks on the "show solution" button, display the correct algorithm step.
   2.3 If the student selects a new node, go to 1.

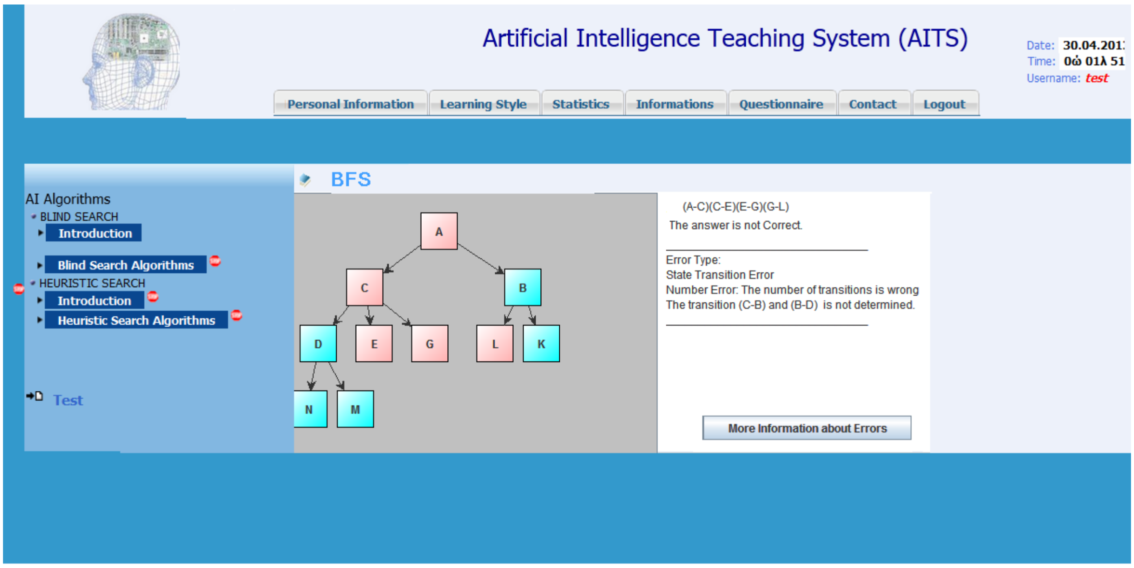

Figure 4. A practice exercise breadth-first algorithm

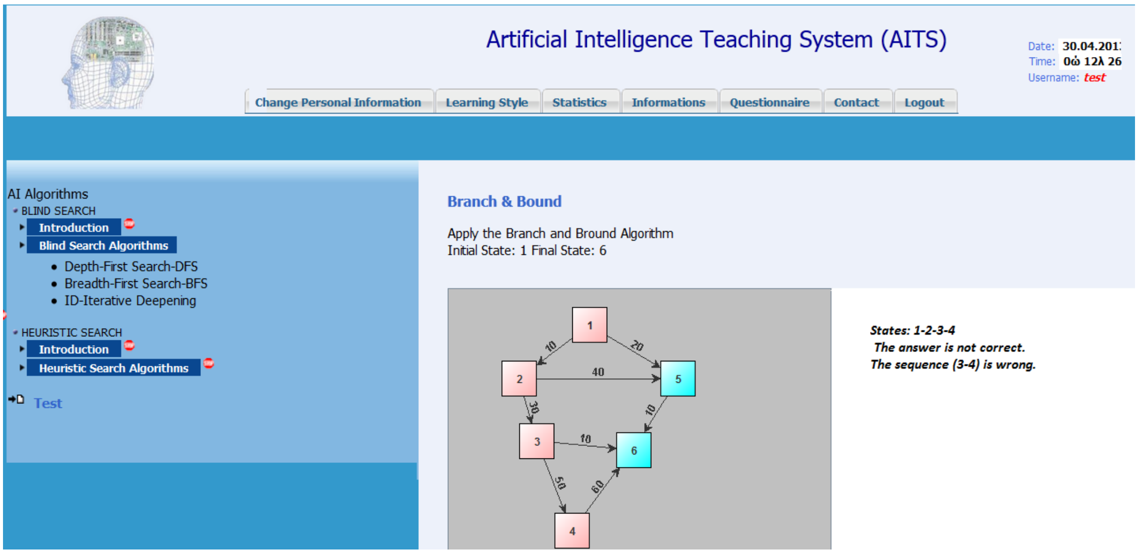

Figure 5. An assessment exercise of branch & bound algorithm

The practice interactive exercises provide immediate feedback after a student’s incorrect actions, because the main objective is to assist the student in learning about search algorithms. Also, AITS monitors student’s actions and behavior and keep records for each student, regarding the number of hints requested for each practice exercise, the total time that he/she spent to answer it correctly and the number of tries. In Figure 4, a practice exercise on breadth-first search (bfs) algorithm and the corresponding feedback and error report is presented.

4.3 Assessment Exercises

Assessment exercises are again interactive exercises that aim to assess the knowledge level of students in the corresponding concept. Also, it does not provide help either feedback during the student's interaction with the system. An assessment interactive exercise presents a graph or a tree and the student is called to reach a specific *goal node* starting from a *start node* implementing an AI search algorithm. The student selects the *nodes (states)* by clinking on them. After the student has selected the sequence of the nodes, the system provides the errors made by the student. More specifically, AITS presents the selection of the nodes that is not correct and provide detailed feedback on the errors made. In Figure 5, an assessment exercise is depicted.

5. EVALUATION

We conducted an evaluation study of the system during the AI course in the fall semester of the academic year 2011-2012. The students were attending the AI course and had attended lectures covering AI search algorithms. The participants in the evaluation study were 40 undergraduate students from those enrolled in the course. The methodology selected to evaluate the system is a pre-test/post-test, experimental/control group design, where the control group used a traditional teaching approach. The students were divided into two groups of 20 students each. The evaluation process was as follows:

(1) A pre-test was given to the students of both groups. The participants answered to exercises on AI search algorithms.

(2) Main Experiment: The control group was involved in the traditional learning and teaching process about AI search algorithms in the AI course. The students solved exercises given by the tutor individually and then discussed them with the tutor. The participants of the experimental group were given access to AITS, to study concepts of AI search algorithms for one week. The system provided exercises and feedback, as described in Section 4, during their interaction with it. Additionally, the students of the experimental group answered to a questionnaire containing questions about their experience from interacting with AITS.

(3) A post-test was given to the students of both groups. All the participants answered again to exercises about AI search algorithms.

Table 1. Descriptive Statistics of Pre-Test and Post-Test

	Group	N	Mean	SD	SE
Pre-Test	Control	20	5.705	1.448	0.32
	Experimental	20	5.825	1.37	0.31
Post-Test	Control	20	6.3	0.951	0.21
	Experimental	20	7.65	1.027	0.23

In order to reliably analyze students' performance, an independent *t*-test was used on the pre-test. The mean value and standard deviation of the pre-test were 5.825 and 1.37 for the experimental group and 5.705 and 1.448 for the control group. As the *p*-value (significance level) was $0.584>0.05$, it can be inferred that the two groups did not significantly differ prior to the experiment. That is, the two groups of students had statistically equivalent performance abilities before the experiment. In Tables 1 and 2 the descriptive statistics are presented as well as the t-test results from assessment of students' learning performance (as calculated by the SPSS v.20 tool). The results reveal that the mean value of the pre-test of the experimental group is higher than the mean value of the pre-test of the control group. The Levene's test confirmed the equality of variances of the control and experimental groups for pre-test (F=0.306, p=0.584) and post-test (F=0.182, p=0.672). Also the t-test result ($p=0.000 < .05$) shows a significant difference between the two groups. Thus, it implies that the students in the experimental group got a deeper understanding of the AI search algorithms.

Table 2. Descriptive Statistics of Pre-Test and Post-Test

Equality variance		Levene's Test for Equality of variances		t-Test for mean					
		F	Sig.	t	Df	Sig.(2-tailed)	MD	95% Confidence Interval of the Difference	
								Lower	Upper
Pre-Test	Equal	0.306	0.584	-0.269	38	0.789	-0.12	-1.022	0.7824
	Unequal			-0.269	37.8	0.789	-0.12	-1.022	0.7825
Post-Test	Equal	0.182	0.672	-4.312	38	0.000	-1.35	-1.9838	-0.7162
	Unequal			-4.312	37.78	0.000	-1.35	-1.9839	0.7161

Moreover, the questionnaire that the experimental group filled in included 9 questions. Question 1 was of multiple choices and concerned the time needed for a student to adapt to the system. Questions 2-5 were based on a Likert scale (1: not at all, 5: very much) (see Table 3), but they included a ‘please explain’ request too. Finally, Questions 6-9 were of open type and concerned strong and weak points or problems faced in using the system.

Their answers show that the students in general enjoyed learning with the system (Q3, Table 3). Most of them (80%) reported that they needed less than five minutes to start using the system (Q1). Also, they found that the user interface is easy to use (Q5, Table 3).

On the other hand, the students (75%) agreed that the system helped them in learning about AI search algorithms. Also, due to this fact they suggest the system to the next year students (80%) (Q4, Table 3).

Table 3. Questionnaire Results

Q	Question	Answers (%)				
		1	2	3	4	5
2	How much did the system help you to learn AI search algorithms?	0	5	30	40	25
3	Did you enjoy learning with the system?	0	0	20	30	40
4	Will you suggest the system to next year students?	0	0	20	40	40
5	Did you find the interface easy to use?	0	0	25	40	35

In general, students felt that the system was helpful and easy to learn. Moreover, they indicated that the system helped them better understand the AI search algorithms and learn, especially through the errors they made.

6. CONCLUSION AND FUTURE WORK

In this paper, we present an adaptive and intelligent web-based educational system, called AITS, which is used for teaching AI search algorithms. The system offers theory descriptions, but most importantly interactive examples and exercises. A student can study the theory and the interactive examples, which use visualized animation to present AI search algorithms in a step-by-step way, to make them more attractive. Also, some explanations are given during interactive examples sessions. Then, the student can try practice exercises, which are interactive exercises and where immediate feedback is given when the student makes an error, but further help is optional. So, the student can try by him/herself to correct the error or ask for help from the system. At a second level, the system provides the solution to each step, when asked by the student. Finally, the student can take a test consisting of assessment exercises, which are interactive, but no help is provided. The result of the test is an assessment of the student's knowledge level on corresponding concepts.

We evaluated the system through a pre-test/post-test and experimental/control group method. Results are very promising. The experimental group made better than the control group. Also, results of a questionnaire given to the experimental group show that the majority of the students liked the system very much. So, it seems that visualized animation and interactivity are two crucial factors that contribute in better learning, at least for subjects like AI search algorithms.

A point for further improvement of the system is the following. At the moment, the system offers a rather limited number of interactive examples and exercises of both types. Implementation of such learning materials is quite time consuming, because it requires programming. So, a way for semi-automatic or automatic generation of such learning elements (actually programs) is a quite interesting direction for further research. This is one of our next research efforts.

ACKNOWLEDGEMENTS

This work was supported by the Research Committee of the University of Patras, Greece, Program “KARATHEODORI”, project No C901.

REFERENCES

Brusilovsky, P., and Paylo, C., 2003. Adaptive and Intelligent Web-Based Educational Systems. *International Journal of AI in Education* 13, pp. 156-169.

Brusilovsky P. et al, 1998. *Adaptive hypertext and hypermedia*. Kluwer Academic Publishers, Dordrecht: The Netherlands.

Hatzilygeroudis, I. et al, 2006. Knowledge-Based Adaptive Assessment in a Web-Based Intelligent Educational System, *Proceedings of the 6th IEEE International Conference on Advanced Learning Technology (ICALT-2006)*. Kerkrade,Netherlands, pp. 651-655.

Hatzilygeroudis, I. et al, 2005. Combining Expert Systems and Adaptive Hypermedia Technologies in a Web Based Educational System. *Proceedings of the 5th IEEE International Conference on Advanced Learning Technology (ICALT-2005),* Kaohsiung, Taiwan, pp. 249-253.

Kordaki, M. et al, 2007. Starting with Algorithmic Structures: A multi-representational, Real-life-activity learning environment for beginners. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications* Vol. 2007, No. 1, pp. 3584-3589.

Malmi, L. et al, 2004. Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. In *Informatics in Education*, Vol. 3, No 2, pp. 267-288.

Naser Abu, S. S., 2008. Developing Vizualization Tool for Teaching AI Search Algorithms. *Information Technology Journal* Vol. 7, No 2, pp. 350-355.

Sanchez-Torrubia, M. G. et al, 2009. PathFinder: a visualization eMathTeacher for actively learning Dijkstra’s algorithm, *Electronic Notes in Theoretical Computer Science* Vol. 224, pp. 151–158.

Wade, V.P. and Ashman, H., 2007. Evolving the infrastructure for technologyenhanced distance learning. *IEEE Internet Comput*ing; Vol. 11, No 3, pp. 16-18.

Woolf, B. P., 2009. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Burlington MA: Morgan Kaufman Publishers.