Educational Technology & Society 3(2) 2000
ISSN 1436-4522

A Web-Based Authoring Tool for Algebra-Related Intelligent Tutoring Systems

Maria Virvou
Department of Informatics
University of Piraeus
80, Karaoli and Dimitriou St.
Piraeus 185 34, Greece
mvirvou@unipi.gr

Maria Moundridou
Department of Informatics
University of Piraeus
80, Karaoli and Dimitriou St.
Piraeus 185 34, Greece
mariam@unipi.gr

ABSTRACT

This paper describes the development of a web-based authoring tool for Intelligent Tutoring Systems. The tool aims to be useful to teachers and students of domains that make use of algebraic equations. The initial input to the tool is a "description" of a specific domain given by a human teacher. In return the tool provides assistance at the construction of exercises by the human teacher and then monitors the students while they are solving the exercises and provides appropriate feedback. The tool incorporates intelligence in its diagnostic component, which performs error diagnosis to students' errors. It also handles the teaching material in a flexible and individualised way.

Keywords

Intelligent tutoring systems, Authoring tools, Problem solving, Algebra, Equations, Distance learning.

Overview

This paper describes the development of an authoring tool for Intelligent Tutoring Systems (ITSs) for the Web. The main objective of this tool is to be useful to teachers and students working in domains that make use of algebraic equations. Such domains could be chemistry, economics, medicine, physics etc. The initial input to the tool is a "description" of a specific domain given by a human teacher. In return the tool provides assistance at the construction of exercises by the human teacher. Then the authoring tool is also able to monitor the students while they are solving the exercises and provide appropriate feedback.

The authoring tool described in this paper and in (Virvou & Moundridou, 1999), incorporates knowledge about the construction of exercises and a mechanism for student error diagnosis that is applicable to many domains that make use of algebraic equations. The tool can also be regarded as an adaptive educational system since several adaptation technologies are applied in it (Brusilovsky, 1998). In particular, the system performs intelligent analysis of student solutions and provides interactive problem solving support. In addition, the system adaptively sorts and annotates the links that the students see. This is done in order to facilitate the students' choice about which problem to solve next resulting in adaptive navigation support (Brusilovsky, 1996). Adaptive navigation support is considered important especially in distance learning situations, where the learning system has to play the role of the teacher and be able to help the student navigate through the course and support him/her individually in a problem solving process (Weber & Specht, 1997; Specht & Oppermann, 1998).

Another feature of the system is its applicability to many domains relevant to algebraic equations. The tool incorporates knowledge about the solution of equations and can perform error diagnosis in this area. Therefore it can either be used as a tool for domains where algebraic equations are a prerequisite piece of knowledge or it can be used for situated learning in Algebra itself. Indeed Collins et al. (1989) point out that cognitive and metacognitive strategies and processes can best be taught through methods that employ a situated learning approach. They also claim that the most common way in which students have learned a process on the way to becoming skilled practitioners is through "apprenticeship". Apprenticeship has students learn process through active participation in the task. Environments that support apprenticeship are considered very beneficial to schools (Guzdial & Kehoe, 1998; Herrington & Oliver, 1999). In addition, a situated view of learning implies that effects on learning of using information and communication technology will depend on the context in which it is used with all the components of a learning environment (Squires, 1999); therefore collaborative learning in which peer group discussion and work is prominent is effective in helping students to learn (e.g. Watson et al., 1993). Indeed the tool can be used for peer group discussion through the Web therefore promoting collaborative learning.

The resulting ITSs of the tool can be used in several settings. They can be used in a classroom where students may be given assignments to complete. Assignments can be done by each student alone or by groups of students. Groups can either be in the same physical location or at a distance discussing through the Web. The tool can also be used by students for homework, which again could be done either by each student individually, or by groups. Finally, the tool can be used in distance learning situations as a standalone tool for working with problems or in combination with other tools which specialise in the creation and delivery of course content (e.g. Goldberg et al., 1996).

The authoring tool can be used by human tutors who may collaborate through the tool's database to create ITSs for their courses. Collaborating human tutors may be at the same physical location or at a distance. In addition their students may be in a class at a specific location or in a virtual class which may be spread over many physical locations, thus promoting distance learning.

Before proceeding to the detailed description of the system we will briefly discuss the work being done in the related fields.

Related work

Computers have been used in education for many years. Research energy in this area has been put in Computer-Based Training (CBT), Computer Aided Instruction (CAI), Intelligent Tutoring Systems (ITSs) and in recent years in Web-based education. Furthermore, recently there has been a growing need for high quality computer based educational programs that may be used in real educational settings. In some cases there have been national attempts to introduce educational software to schools (e.g. Alexandris et al., 1998) and/or higher education (e.g. Gilbert, 1999). There have also been numerous projects from Universities that concern the incorporation of educational software in their courses (e.g. Berz et al., 1999). Similarly there are attempts to create software useful to schools (e.g. Virvou & Tsiriga, 1999; Virvou & Maras, 1999).

However, there has often been a criticism that technology may be limited to research products that may not be used in school environments and that educational software that is used in schools may not be up to the latest standards of educational technology and pedagogical issues. Another problem is that there has not yet been satisfactory integration of different educational technologies such as ITSs and Web-based tutoring systems so that real environments may benefit from them in full.

In particular Intelligent Tutoring Systems have the ability to present the teaching material in a flexible way and to provide students with individualised instruction and feedback. ITSs have been shown to be effective at increasing the students' motivation and performance in comparison with traditional learning methods and thus ITSs may significantly improve the learning outcomes (Mark & Greer, 1991; Shute et al., 1989).

However, ITSs have often been criticised that they miss the mark in terms of task reality, feasibility and effectiveness (Mc Graw, 1994). One reason for this has been the difficulty in developing an ITS even in small domains. For example, Woolf and Cunningham (1987) have estimated that the development of an ITS takes more than 200 hours to produce an hour of instructional material, which in most cases cannot be reused. Furthermore the development of an Intelligent Tutoring System requires the involvement of a large number of people, including experts on the specific domain, instructors and programmers. A solution to these problems may be the development of authoring tools, which will help construct cost-effective and reusable ITSs in various domains.

It has been generally acknowledged (Hartley & Sleeman, 1973; Burton & Brown, 1976; Wenger, 1987) that the main components of an ITS are the domain knowledge, the student modelling component, the advice generator and the user interface. Accordingly, there are authoring tools that focus on various aspects concerning each of these components. Murray (1999) has classified the existing authoring tools based on their capabilities and concluded that they fall into two broad categories: those which focus on how to sequence and teach relatively canned content (pedagogy-oriented authoring tools) and those which focus on providing rich learning environments in which students can learn skills by practising them and receiving feedback (performance-oriented authoring tools).

The system we describe in this paper mainly belongs to the category of performance-oriented authoring tools, since it provides a learning environment in which students can learn how to solve problems in various algebra-related domains. In particular, this tool deals with the generation of instruction, since it offers the ability of problem construction. In that sense it shares the same focus with RIDES (Munro et al., 1997), an authoring system used for the construction of tutors that teach students how to operate devices through simulations. RIDES generates instruction by providing tools for building graphical representations of a device and for defining this device’s behaviour. A system which adds capabilities to RIDES is DIAG (Towne, 1997), a tool which simulates equipment faults and guides students through their diagnosis and repair. DIAG is concerned with the creation of domain knowledge and performs student error diagnosis by providing a mechanism that is applicable to many domains that are related to diagnosis of equipment failures. In the same way our tool performs student error diagnosis by providing a mechanism that can be applied to many algebra related domains.

However, the authoring tool described in this paper also shares capabilities with authoring tools belonging to the pedagogy-oriented category. In particular, it gives instructors the ability to control the order by which students solve exercises, by assigning to each exercise a “level of difficulty”. Therefore, this tool, beyond generating student problems is also concerned with managing their sequence. The latter is a characteristic that can likewise be met in a system called REDEEM (Major et al., 1997), which does not generate instruction but rather focuses on the representation of instructional expertise. REDEEM expects the human instructor to categorise tutorial “pages” in terms of their difficulty, their generality, and whether they are prerequisite for other pages and in that way accomplishes to sequence content and learner activities.

A question that arises is how general an ITS authoring tool should be. Trying to build an authoring tool that can be used to produce ITSs in every possible domain is not considered feasible. Thus, Murray (1999) concludes that one way to have both powerful and usable authoring tools is to limit them to particular domains or knowledge types. The tool we describe in this paper is in compliance with Murray's suggestion about the design trade-offs concerning the generality and usability of authoring tools, since it can be used to build ITSs only in domains that can be described by algebraic equations.

On the other hand, Web-based education has numerous advantages such as the convenience of taking a course without leaving the workplace or home and the reduced cost (Berz et al., 1999). In addition, teachers and educational researchers are encountering both unprecedented opportunities and challenges to adapt networks to their classrooms and research fields (Chou, 1999). However, most of the educational applications (tutorials, course notes, etc.) that have been delivered through the World Wide Web are just electronic books with very limited interactivity and diagnostic capability. An integration of ITS and WWW-based technologies would be very beneficial for the purposes of education. Indeed, there have been successful attempts to either move existing ITSs to the WWW or build from scratch web-based ITSs (Brusilovsky et al., 1996; Eliot et al., 1997; Ritter, 1997). However, it would be even more useful to students and teachers to have authoring tools for ITSs on the Web so that new courses can be created remotely by teachers and then students can remotely use the courses.

System’s architecture

The system’s underlying architecture is shown in Figure 1. The teacher’s input is the domain description in terms of variables, equations and units of measure and all the information needed to construct a problem (known and unknown variables, level of difficulty, etc.). This information is stored and used by the Problem Solver, a component that interacts with the student while s/he is solving a problem. When the student makes a mistake, the Error Diagnoser is responsible for finding out the cause of it. According to that and the Student Model kept for each student, the Advice Generator provides the student with the appropriate feedback. The Student Model is updated at every interaction of the student with the system. Using the information kept in that model the system performs adaptive navigation support, builds progress reports and helps the instructor reconsider the level of difficulty that s/he had originally assigned to the constructed problems.

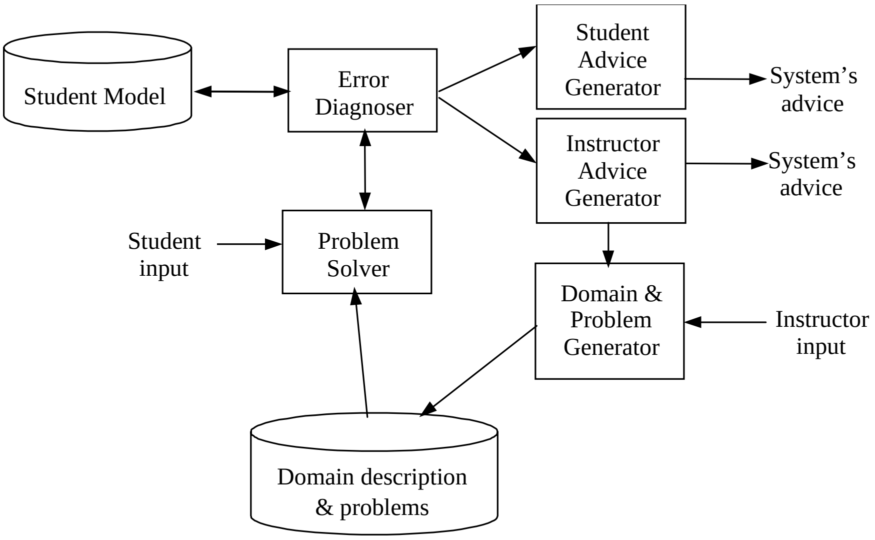


Figure 1. System's architecture

The implementation of the system is based on the client-server architecture. Both students and instructors are clients who can use the teaching and authoring services offered by the system using a conventional WWW browser. The system itself resides on a WWW server. All of the system's components excluding the Problem Solver, are completely implemented in JAVA to allow platform and browser independent access to the system. The Problem Solver is implemented in PROLOG. This was selected due to its symbolic manipulation capabilities, which are required in an equation solving process.

Description of the system's use

The tool takes input from a human instructor about a specific equation-related domain (e.g. economics). This input consists of knowledge about variables, units of measure, formulae and their relation. An example is illustrated in Figure 2. For each of these elements the instructor can associate a URL pointing to an HTML file which contains information concerning this element. The instructor does not have to provide the complete list of variables and equations that describe the domain, all at once. The instructor may only enter the ones that will be used in the problems to be constructed in the current interaction and add more in subsequent interactions. The tool accumulates domain knowledge each time that the human instructor gives new input. This means that the instructor may give information to the tool at the same rate as lessons progress in a course. An example of input to the system that an instructor could provide from the domain of physics is shown in Table 1.

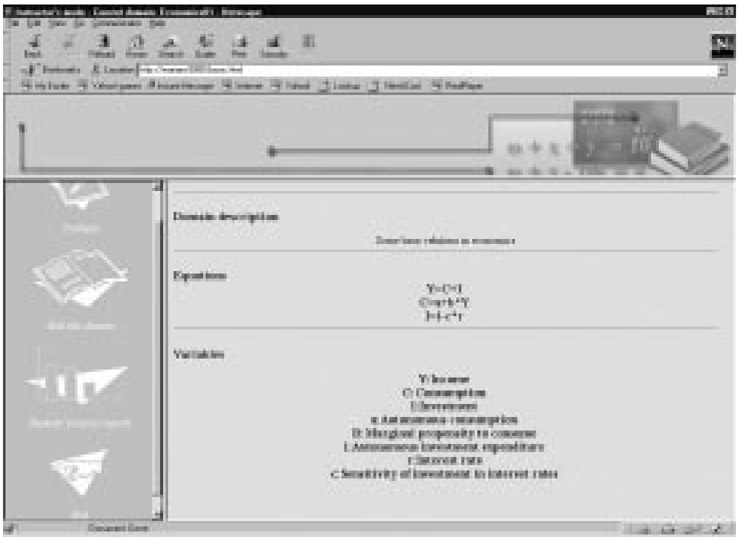
Variables:	
Velocity	v
Initial velocity	v_0
Acceleration	a
Time	t
Force	F
Mass	m
Impulse	J
Formulae:	
F=m*a	
J=m*v	
$v=v_0+a*t$	

Table 1. Example of input to the system from the domain of physics

A second example from the domain of economics which is also equation related can be seen in Table 2.

Variables:	
Income	Y
Consumption	C
Investment	I
Autonomous consumption	a
Marginal propensity to consume	b
Autonomous investment expenditure	i
Interest rate	r
Sensitivity of investment in interest rates	c
Formulae:	
Y=C+I	
C=a+b*Y	
I=i-c*R	

Table 2. Example of input to the system from the domain of economics



Domain description

Some basic relations in economics

Equations

Y=C+I
C=a+b*Y
I=i-c*r

Variables

Y: Income
C: Consumption
I: Investment
a: Autonomous consumption
b: Marginal propensity to consume
i: Autonomous investment expenditure
r: Interest rate
c: Sensitivity of investment in interest rates

Figure 2. A sample screen (instructor’s mode)

Instructor’s mode

When the human instructor wishes to create exercises s/he can type in what is given and what is asked and the tool can either construct the full problem text or provide consistency checks that help the instructor verify its completeness and correctness. In case of redundancies in the given data the tool lets the instructor know. After the construction of a problem the tool lets the instructor preview the problem text and the solution of the exercise as formulated by the system. At this point, the instructor is asked to assign to the problem the appropriate “level of difficulty”. The system uses this measure in order to suggest to each student (while in student’s mode) what problem to try next.

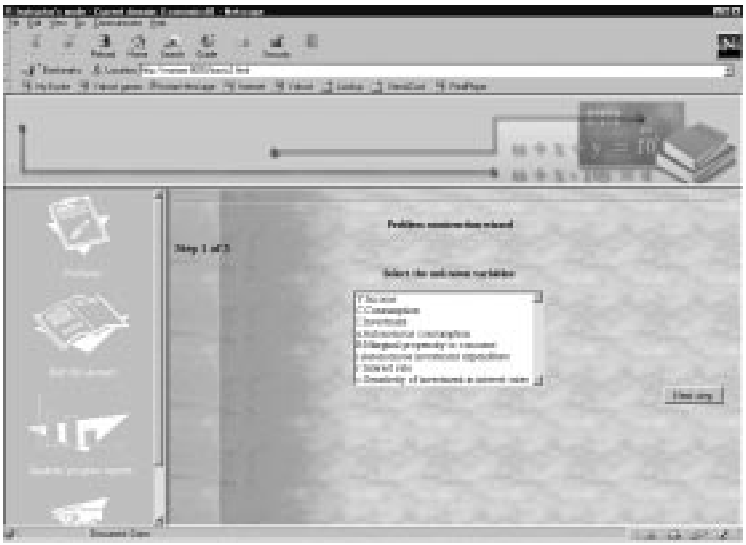
However, while students are tackling the given problems the system collects evidence about the level of difficulty so that it can provide feedback to the instructor. For example, if the majority of the students of a certain level have failed in solving a particular problem, which has been assigned to this level, then the instructor is being informed. In a case like this, perhaps the instructor may wish to reconsider the level of difficulty since there is evidence that the problem may be of a higher level of difficulty. On the other hand, if many students have managed to solve a problem of a higher level of difficulty than the one proposed by their instructor, the level of difficulty may have been overestimated by the instructor. In this case too, the system informs the instructor. In both cases, the tool does not take the initiative to alter the level of difficulty by itself: it suggests the instructor to increase or decrease this measure according to the observed students’ performance in a specific problem. In this way an instructor is being assisted by the system in the classification of problems.

The tool incorporates also a user modelling mechanism that focuses on diagnostic reasoning about students' errors. This mechanism is used by the system when a student tackles the problems that the instructor has created and is explained in more detail in the next section.

There are two types of problem that the system can assist the instructor to construct:

Problems without numbers. In problems without numbers the system displays every variable that the human instructor has entered. The human instructor should specify which variable is the unknown, which one is given and the type of change. For example in the domain of economics the instructor could select as unknown the variable "income", and as given an "increase" at the level of "interest rates". The system would then produce the following problem text: "How will the increase of interest rates affect the level of income?". This kind of problem evaluates the students' knowledge of the equations involved in each of these problems. In addition it evaluates the students' ability to decide about the influence of each variable over the others. In cases like this students are not requested to solve a particular system of equations. Instead, they are requested to work with analogies. In this way, such problems might measure the students' overall understanding in the domain being taught.

Problems with numbers. In problems with numbers the system displays again every variable that the human instructor has entered and requests the unknown (Figure 3). The system considers automatically all the variables, which depend on the "unknown" (according to the equations), as possible given data. These variables are shown to the instructor who should now enter their values. The system follows the instructor's actions and reports any inconsistencies. For example, if the instructor enters values for both the dependent and independent variables of an equation the system points out the error. Finally, the system produces the problem text. An example of problem text is the following: "If the force is 100 Newtons, the mass is 25 kg, the initial velocity is 0 m/sec and the time is 5 secs, then find the impulse." The instructor may change the problem text to make it more comprehensible; for example: "A force of 100 Newtons is acting on a 25 kg object which is initially stable. After 5 secs how much is the impulse?". In such problems, the students are tested over their ability to solve a system of linear equations (mathematical skills) and their knowledge of the equations describing the particular domain.



Problem construction wizard

Step 1 of 5

Select the unknown variable

Figure 3. Problem construction

Student's mode

The system recognises each student by his/her user name and his/her password. Each student is assigned a level of knowledge by the system according to his/her past performance in solving problems with the tool. When a student interacts with the tool for the first time s/he is asked to fill in a questionnaire concerning his/her familiarity with the specific domain, his/her ability to solve equations and his/her competence as a computer user. Based on the student's answers the tool assigns to each student an initial level of knowledge, which will then be modified according to the student's progress. The students' "level of knowledge" and the "level of difficulty" that is assigned to each problem are both in the same range. The tool suggests each student to try the problems corresponding to his/her level of knowledge. For example, if a student at a specific session of interaction with the system is considered to be at the third level of knowledge, the tool will suggest to the student problems of the third level of difficulty to be solved next.

When a student attempts to solve an exercise the system provides an environment where the student gives the solution step by step. The system compares the student's solution to its own. The system's solution is generated by the domain knowledge about algebraic equations and about the specific domain in which the exercise belongs (e.g. economics). While the student is in the process of solving the exercise the system monitors his/her actions. If the student makes a mistake, the diagnostic component of the system will attempt to diagnose the cause of it. The diagnosis of the underlying cause of a mistake is a difficult task for an ITS because the observable symptoms need to be analysed further. As Hollnagel (1993) pointed out, there is an important distinction between the underlying cause or genotype of an error and the observable manifestation or phenotype of the error. There are three categories of cause that can be recognised by the system:

Typographic errors. In this case, errors are ignored as far as the student knowledge is concerned.

Mathematical errors. In this case, errors are due to solving equations incorrectly.

Domain errors. In this case, errors are due to lack of knowledge or misconceptions concerning the domain taught. For example, the student may not remember correctly one formula.

Each student error is attributed to one of the three categories listed above. For example, if a student types in the equation: x+5= = 8, where there has been a repetition of the equal sign, the system recognises this as a typographic error and ignores it. However, there may be cases where a certain error may be attributed to more than one category. For example, if a student types in the equation x+5=9 where 9 should have been 8, then this error could either be a typographic error or a mathematical error. In cases like this the system uses a history mechanism about a particular user in order to resolve ambiguities. For example, if the particular user has been recorded to have frequently made typographic errors but no mathematical errors at all then the system may favour the hypothesis of a typographic error in this case too.

Student model

The "history mechanism" records certain student features that have been inferred during past interactions such as persistence of a certain type of error (e.g. mathematical error). These features form the long-term student model (Rich, 1979; 1983) which represents the student's knowledge both in the domain being taught and in solving linear equations. This student model is a combination of a stereotype and an overlay student model (de Rosis et al., 1993). The stereotype student model (formed after the short interview with a new student) classifies initially the student according to the student's knowledge of the domain and the student's mathematical skills. As a result of this it assigns the student to a stereotype (beginner, or intermediate, or expert). For example, a student may be assigned to the stereotype "expert" for his/her mathematical skills and to "beginner" for his/her knowledge of the domain taught. The stereotype model also defines initial values for the overlay student model. The latter is represented by a set of pairs "concept-value" which are explained below.

The concepts are domain concepts and concepts concerning the process of solving equations (e.g. separating known from the unknown). Domain concepts include domain variables. For example, in the domain of Economics the variables "Consumption", "Income", etc. are seen as concepts. It is assumed by the system that a student knows a concept if in a given problem that this variable-concept is needed, s/he enters the correct equation that defines this variable. For example, concerning the variable-concept "Income" the correct equation should be: Y=C+I. In cases when a variable can be defined by more than one equation, the corresponding concept is considered "mastered" by the student if s/he has used successfully all the different equations –in different problems- at least once.

The value for each concept is an estimation of the student's knowledge level of this concept (poor, or average or good). As we have already mentioned, these values are initialised from the stereotype student model. If for example the stereotype model indicates that a student is "expert" as for his/her mathematical skills and "beginner" as for his/her knowledge in the domain, then all the concepts comprising the overlay student model are given the corresponding values: The initial value for every concept concerning the process of solving equations will be "good" while the initial value for every domain concept is "poor".

The student model is used by the system in order:

- To resolve ambiguities that arise from errors for which more than one hypothesis can be generated as to what the cause of the error has been.

- To form individualised progress reports of the student, which could be requested by the student and/or the instructor.
- To adjust the level of knowledge of each student.
- To offer advice to the instructor concerning a modification at the level of difficulty s/he initially assigned to each problem.

The system provides the appropriate feedback depending on the cause of the error that the student has made. This feedback could be for example an error message followed by an explanation about the student's underlying misconception. In cases when the error is attributed to erroneous or incomplete knowledge of the domain taught, the feedback provided by the system includes links to corresponding teaching material (URLs to HTML documents provided by the instructor during the authoring process).

Future work

The system we described in this paper is in an early state of its evolution and so the first task is to make it fully functional. The next step is to evaluate it. This is a common stage of evolution for most authoring tools at the moment. For example, Murray in a review of existing ITS authoring tools (1999), found that 5 out of 18 authoring tools were early prototypes. This is probably due to their complex nature and the fact that they have not been used long enough to claim their "existence proof". However, it is within our future plans to evaluate our authoring tool as soon as we complete it.

A good approach for evaluating it would be to combine computer logging techniques with self-reporting methods in a similar way as (Chou, 1999). For example, the authoring tool can be evaluated by keeping a record of human tutors' and students' actions using the system. These records can then be given to human tutors who would not have used the system to comment them. In addition, all kinds of users of the system could be given questionnaires over the Web to comment on the system's usability.

Plans for future work also include the investigation of the integration of our authoring tool with another one dealing with the representation of instructional expertise. The World Wide Web is a distributed environment and an important educational medium. Therefore, there have been considerations concerning how many different web-based educational systems might interact with one another or even be integrated. For example, Brusilovsky et al. (1997) have successfully integrated two separate web-based adaptive tutoring systems. The integrated system incorporated conceptual instruction through the first system (Interbook) and problem solving interaction through the second one (PAT Online). Similarly, our system which is an authoring tool dealing with problem construction and solving might be combined with existing authoring tools dealing with the representation of instructional expertise.

Another direction for future work concerns the integration of our tool within a communication framework. Since the users of this tool are students and instructors who are interacting with the system through the Web, it might be helpful if both kinds of user could communicate and collaborate with each other using tools to chat, to send and receive e-mails and so on. These facilities are quite common in tools which deal with the creation and delivery of web-based courses (Goldberg, 1997) but are not, in our knowledge, popular in ITS authoring tools.

Conclusions

In this paper we described a web based authoring tool for Intelligent Tutoring Systems (ITSs). The initial input to the tool is a "description" of a specific domain given by a human instructor. In return the tool assists the instructor at the construction of exercises and it then monitors the students while they are solving the exercises. Finally it provides individualised feedback. The components that constitute the system are namely the Domain & Problem Generator, the Problem Solver, the Error Diagnoser and the Advice Generator. What we consider as the most important characteristics of this system are:

a) The fact that the system is accessible through the WWW. This means that students from any place can practice with the same exercises. In addition human instructors from any place can contribute to the system by adding exercises and by improving the description of the domain taught.
b) The fact that this authoring tool can be used in creating teaching material (exercises) for many different domains. The only requirement is for the domain to be "equation-related". Even if a small portion of the domain can be described by equations, the authoring tool can be used for this portion of the domain. This is possible due to the system's easiness of use: instructors do not need to invest much time learning how to "describe" a domain and construct exercises with the tool; all they need to do is enter variables and

equations. Then with very little effort they can produce problems which are made automatically available to students.

c) The tool's usefulness to students' learning. While students are solving problems they are being monitored by the system. This means that individualised feedback (concerning the students' mistakes and performance in solving the problems) is given by the system to both students and instructors. While most authoring tools deal solely with the creation of teaching material, the system described here provides also reports containing information about the students' performance in that teaching material. Furthermore it assists students' learning in both algebra-related domains and algebra itself. In particular, for algebra the tool provides a "situated learning" environment which is considered very beneficial for students' learning of concepts and procedures.

Acknowledgements

The authors would like to express their appreciation to Dr N. Diamantidis and the anonymous reviewers for their useful comments.

References

- Alexandris, N., Virvou, M. & Moundridou, M. (1998). A Multimedia Tool for Teaching Geometry at Schools. In Ottmann, T. & Tomek, I. (Eds.) *Proceedings of ED/MEDIA 98, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Vol. 2, Charlottesville, VA: AACE, 1595-1597.
- Berz, M., Erdelyi, B. & Hoefkens, J. (1999). Experiences with interactive remote graduate instruction in beam physics. *Journal of Interactive Learning Research*, 10 (1), 49-58.
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), 87-129.
- Brusilovsky, P., Schwarz, E. & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In Frasson, C., Gauthier, G. & Lesgold, A. (Eds.), *3rd International Conference on Intelligent Tutoring Systems, ITS-96* (Lecture Notes in Computer Science, Vol. 1086), Berlin: Springer Verlag, 261-269.
- Brusilovsky, P., Ritter, S. & Schwarz, E. (1997). Distributed intelligent tutoring on the Web. In du Boulay, B. & Mizoguchi, R. (Eds.) *Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Amsterdam: IOS, 482-489.
- Brusilovsky, P. (1998). Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies. In *Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring systems - ITS '98*, San Antonio, TX, http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/brusilovsky.html
- Burton, R.R. & Brown, J.S. (1976). A tutoring and student modelling paradigm for gaming environments. In Colman, R. & Lorton, P.Jr. (Eds.) *Computer Science and Education*, ACM SIGCSE Bulletin, 8 (1), 236-246.
- Chou, C. (1999). Developing CLUE: A Formative Evaluation System for Computer Network Learning Courseware. *Journal of Interactive Learning Research*, 10 (2), 179-193.
- Collins, A., Brown, J.S. & Newman, S.E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing and mathematics. In Resnick L. B. (Ed.) *Knowing, learning and instruction: Essays in honour of Robert Glaser*. Hillsdale, NJ: Lawrence Erlbaum Associates, 453-494.
- de Rosis, F., de Carolis, B. & Pizzutilo, S. (1993). User tailored hypermedia explanations. In Ashlund, S., Mullet, K., Henderson, A., Hollnagel, E. & White, T. (Eds.) *Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems*, New York, NY: ACM, 169-170.
- Eliot, C., Neiman, D. & Lamar, M. (1997). Medtec: A Web-based intelligent tutor for basic anatomy. In Lobodzinski, S. & Tomek, I. (Eds.) *Proceedings of WebNet '97, World Conference of the WWW, Internet and Intranet*, Charlottesville, VA: AACE, 161-165.
- Gilbert, L. (1999). Some Valuable Lessons from the Teaching and Learning Technology Programme in the U.K. *Journal of Interactive Learning Research*, 10 (1), 67-85.
- Goldberg, M.W., Salari, S. & Swoboda, P. (1996). World Wide Web - Course Tool: An environment for building www-based courses. *Computer Networks and ISDN Systems*, 28, 1219-1231.
- Goldberg, M.W. (1997). Communication and Collaboration Tools in World Wide Web Course Tools (WebCT). In *Proceedings of the Conference Enabling Network-Based Learning*, ENABLE'97, Espoo, Finland, http://about.webct.com/library/comm.html

- Guzdial, M. & Kehoe, C. (1998). Apprenticeship-Based Learning Environments: A principled approach to providing Software-realized scaffolding through hypermedia. *Journal of Interactive Learning Research*, 9 (3-4), 289-336.
- Hartley, J.R. & Sleeman D.H. (1973). Towards intelligent teaching systems. *International Journal of Man-Machine Studies*, 5, 215-236.
- Herrington, J. & Oliver, R. (1999). Using situated learning and multimedia to investigate higher-order thinking. *Journal of Interactive Learning Research,* 10 (1), 3-24.
- Hollnagel, E. (1993). The Phenotype of Erroneous Actions. *International Journal of Man-Machine Studies*, 39, 1-32.
- McGraw, K. L. (1994). Performance Support Systems: Integrating AI, Hypermedia and CBT to Enhance User Performance. *Journal of Artificial Intelligence in Education*, 5 (1), 3-26.
- Major, N., Ainsworth, S. & Wood, D. (1997). REDEEM: Exploiting Symbiosis Between Psychology and Authoring Environments. *International Journal of Artificial Intelligence in Education*, 8, 317-340.
- Mark, M.A. & Greer, J.E. (1991). The VCR tutor: Evaluating instructional effectiveness. In Hammond, K. J. & Gentner, D. Q. (Eds.) *Proceedings of 13th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates, 564-569.
- Munro, A., Johnson, M., Pizzini, Q., Surmon, D., Towne, D. & Wogulis, J. (1997). Authoring Simulation-centered tutors with RIDES. *International Journal of Artificial Intelligence in Education*, 8, 284-316.
- Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10, 98-129.
- Rich, E. (1979). User Modelling via Stereotypes. *Cognitive Science*, 3 (4), 329-354.
- Rich, E. (1983). Users as Individuals: Individualizing User Models. *International Journal of Man-Machine Studies*, 18, 199-214.
- Ritter, S. (1997). PAT Online: A Model-tracing tutor on the World-wide Web. In Brusilovsky, P., Nakabayashi, K. & Ritter, S. (Eds.) *Proceedings of Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan: ISIR, 11-17.
- Shute, V., Glaser, R. & Raghaven, K. (1989). Inference and Discovery in an Exploratory Laboratory. In Ackerman, P.L., Sternberg, R.J. & Glaser, R. (Eds.) *Learning and Individual Differences*, San Francisco: Freeman, 279-326.
- Specht, M. & Oppermann, R. (1998). ATS-Adaptive Teaching System a WWW-based ITS. In Timm, U. (Ed.) *Proceedings of Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen: ABIS 98*,
  http://129.26.167.17:8080/pub_html/pdf/ABIS98.pdf
- Squires, D. (1999). Usability and Educational Software Design: Special Issue of Interacting with Computers. *Interacting with Computers*, 11 (5), 463-466.
- Towne, D. (1997). Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*, 8, 262-283.
- Virvou, M. & Maras, D. (1999). An intelligent multimedia tutor for English as a second language. In Collis, B. & Oliver, R. (Eds.) *Proceedings of ED-MEDIA 99, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Vol. 2, Charlottesville, VA: AACE, 928-932.
- Virvou, M. & Moundridou, M. (1999). An authoring tool for Algebra-related domains. In Bullinger, H.-J. & Ziegler, J. (Eds.) *Human-Computer Interaction: Communication, Cooperation, and Application Design, Proceedings of the 8th International Conference on Human-Computer Interaction - HCI International '99,* Vol. 2, Mahwah, NJ: Lawrence Erlbaum Associates, 647-651.
- Virvou, M. & Tsiriga, V. (1999). EasyMath: A Multimedia Tutoring System for Algebra. In Collis, B. & Oliver, R. (Eds.) *Proceedings of ED-MEDIA 99, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Vol. 2, Charlottesville, VA: AACE, 933-938.
- Watson, D., Moore, D.A. & Rhodes, V. (1993). Case studies. In Watson, D. (Ed.) *The Impact Report*, King's College London, 61-99.
- Weber, G. & Specht, M. (1997). User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems. In Jameson, A., Paris, C. & Tasso, C. (Eds.) *User Modeling: Proceedings of the 6th International Conference, UM97,* Vienna, New York: Springer, 289-300.
- Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.
- Woolf, B.P. & Cunningham, P.A. (1987). Multiple knowledge sources in intelligent teaching systems. *IEEE Expert*, 2 (2), 41-54.