

Modelling the Instructor in a Web-Based Authoring Tool for Algebra-Related ITSs

Maria Virvou and Maria Moundridou

Department of Informatics, University of Piraeus,
80, Karaoli and Dimitriou St., Piraeus 185 34, Greece
{mvirvou, mariam}@unipi.gr

Abstract. This paper describes the development of a web-based authoring tool for Intelligent Tutoring Systems and focuses on its instructor modelling capabilities. The tool is called WEAR and aims to be useful to teachers and students of domains that make use of algebraic equations. Specifically, the tool provides assistance to human teachers while they are constructing exercises and it then monitors the students while they are solving the exercises and provides appropriate feedback. The instructor modelling mechanism renders the system adaptable to the specific needs and interests of each individual user (instructor), concerning the construction of new exercises and the retrieval of existing ones.

1 Overview

In the recent years Web-based education has attracted a lot of research energy and has resulted in significant advances concerning distance learning. There are obvious benefits in Web-based education since any student in any place having just a computer and an Internet connection can use a course or educational application developed and installed in some other place. Furthermore, there is a growing need for high quality computer based educational programs that may be used in real school environments [1; 17]. This need can be supported to a large extent by Web-based education. Therefore recently, a large number of educational applications (tutorials, course notes, etc.) have been delivered through the World Wide Web. However, most of them are just electronic books with very limited interactivity. This does not undermine the potential of Web-based education, which can be improved significantly if combined with the technology of Intelligent Tutoring Systems (ITSs). Indeed there have been a lot of successful attempts to either move existing ITSs to the WWW or build from scratch web-based ITSs [2; 4; 13].

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 it can be said that ITSs can significantly improve the learning outcomes [9; 14]. However ITSs have often been criticised that they miss the mark in terms of task reality, feasibility and effectiveness [7]. One reason for this has been the difficulty in developing an

Intelligent Tutoring System even in small domains. For example, Woolf and Cunningham [19] 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. A possible solution to these problems may be the development of authoring tools, which will help construct cost-effective and reusable ITSs in various domains.

ITSs have been described [5; 3; 18] as having four main components, namely: the domain knowledge, the student modelling component, the advice generator and the user interface. Accordingly, there are systems (authoring tools/shells) that offer the ability to their users to author one and in some cases more than one of these components [11].

The authoring tool described in this paper and in [16] is called WEAR (WEB-based authoring tool for Algebra Related domains). WEAR 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. WEAR deals with the generation of instruction, since it offers the ability of problem construction. In that sense it shares the same focus with RIDES [10], 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 [15], a tool that 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 WEAR performs student error diagnosis by providing a mechanism that can be applied to many algebra-related domains. Furthermore, WEAR gives instructors the ability to control the order by which students solve exercises, by assigning to each exercise a "level of difficulty". Therefore, WEAR 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 [8], 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.

WEAR also gives teachers the ability to search for already constructed exercises and make them available to their own class. Thus, teachers can benefit from the database of problems that have been previously created using WEAR and have been stored in its database. However, the problems stored may be relevant to a wide range of algebra-related domains, may be addressed to many different categories of student, may be associated to many several levels of difficulty and so on. These differences in the problems stored, indicate a need for WEAR to be adaptable to each particular instructor's interests so that it may help him/her in the retrieval of the appropriate information from its database. In addition, to avoid the repetition of exercises WEAR must closely monitor the instructor during the construction of a new problem and notify him/her in case a similar problem already exists in its database. For these reasons, WEAR incorporates an instructor modelling component.

This is considered a major issue for WEAR which aims at providing assistance tailored to each individual teacher's needs and interests. In an ITS there is no need of modelling the instructor since the users of such systems are students learning with them. However, the users of an authoring tool for ITSs are mainly instructors who are interacting with the system in order to build an ITS. A user modelling component for these users will be responsible of keeping and modifying their personal user models. Based on the information provided by these user models, the system will be able to adapt itself to the needs and interests of each individual instructor. This is even more important in cases of a Web-based authoring tool that is available to a diversity of instructors spread all over the range of scope of the Web site. Though the instructor modelling component is not part of the ITS that the tool generates, we consider it to be an important part of the tool since it affects the way the instructors interact with the system and consequently the effectiveness of the generated ITSs.

2 System Operation

The main objective of this tool is to be useful to teachers and students of domains that make use of algebraic equations. Such domains could be chemistry, economics, physics, medicine, etc. In particular the tool takes input from a human instructor about a specific equation-related domain (e.g. physics). This input consists of knowledge about variables, units of measure, formulae and their relation.

Table 1. Examples of input to the system from the domain of physics and economics

PHYSICS		ECONOMICS	
Variables:		Variables:	
Velocity	v	Income	Y
Initial velocity	v_0	Consumption	C
Acceleration	a	Investment	I
Time	t	Autonomous consumption	a
Force	F	Marginal propensity to consume	b
Mass	m	Autonomous investment expenditure	i
Impulse	J	Interest rate	r
		Sensitivity of investment in interest rates	c
Formulae:		Formulae:	
$F=m*a$		$Y=C+I$	
$J=m*v$		$C=a+b*Y$	
$v=v_0+a*t$		$I=i-c*r$	

The instructor does not have to provide the complete list of variables and equations that describe the domain, all at once. S/he 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. Examples of input to the system that an instructor could provide from the domain of physics and economics are shown in Table 1.

2.1 Modes of Operation

WEAR can function in two different modes: the instructor's mode and the student's mode. In the instructor's mode the instructor is able to construct new exercises and/or retrieve previously created exercises. In both cases WEAR provides automatic assistance. Since this paper focuses on the system's instructor modelling capabilities, we will discuss in detail the instructor's mode of operation in the following sections and briefly describe the student's mode of operation here.

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 students' 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.

2.2 Instructors' Assistance at the Construction of Exercises

When the human instructor wishes to create exercises s/he is guided by the system through a step by step procedure. At each step of this procedure the instructor should specify values for some parameters needed to construct an exercise. Such parameters could be for example what is given and what is asked in the exercise to be constructed. After the completion of this procedure the tool constructs the full problem text and provides 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. For example, if the instructor assigns a value to both the dependent and independent variables of an equation, s/he will receive a notification by the system concerning this redundancy. 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.

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 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 the 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.

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, but rather working 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 (Fig. 1). 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 fewer variables than those needed for the problem to be solvable then 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.

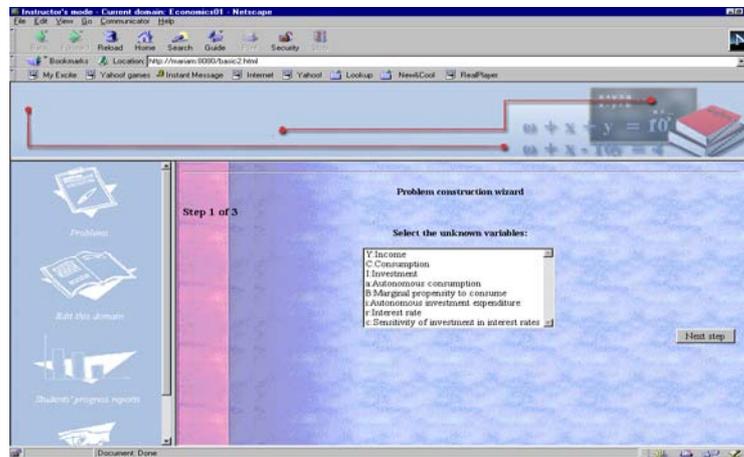


Fig. 1. Problem construction

2.3 Retrieval of Exercises

Beyond constructing an exercise by himself/herself, the instructor has the ability to explore the exercises constructed by others and choose the ones that s/he desires to be accessible by his/her class. The exercises are first categorised according to the domain they belong. At a second level the exercises of each domain are categorised according to the variables they involve and their level of difficulty. Every variable of the domain can possibly form an exercise category. For example, an exercise like: “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.” belongs to the broad category “Physics” and in the sub-categories “Impulse”, “Velocity” and “Acceleration” due to the variables involved in it. The same exercise could also belong to the sub-category “level of difficulty 1” based on the exercise’s level of difficulty (as defined by the instructor).

The categorisation of each exercise (according to the variables involved) is achieved by the system through the following algorithm:

1. Assign the exercise to the categories of every unknown variable in this exercise (In the above example this step results in the category “Impulse” which is the unknown variable).
2. Search the equations that must be solved in order to define the unknown variables’ values (in the example, that is the equation $J=m \cdot v$) for the independent variables that are not “given” in this exercise and assign the exercise to these categories too (in our case that is the variable-category “Velocity”). Consider these variables that must be given a value by solving an equation as “unknown” variables and repeat this step (this will end in assigning the exercise to the category “Acceleration”).

Instructors are allowed either to browse the collection of exercises by selecting the categories and sub-categories that match their needs and interests, or to search the

entire collection using some keywords. A user modelling mechanism incorporated in the system is responsible for tailoring the interaction of the instructors with the system to the instructors' needs.

3 Instructor Modelling

The instructor modelling component monitors the instructors' interactions with WEAR and constructs and/or updates a long-term implicit user model [12] for each user - instructor. As we have already mentioned, an instructor either searches for an already constructed exercise or constructs an exercise by himself/herself. WEAR infers from these two actions the users' interest or expertise in something, respectively. This is similar to a system called InfoVine [6], which infers that users are interested in something if they are repeatedly asking the system about it whereas they are expert in something if they are repeatedly telling the system about it. In WEAR, when a user frequently searches for specific categories of exercise then it is inferred that this particular user is highly "interested" in these categories of exercise; whereas when a user often constructs exercises that belong to the same category the inference made is that this user is a "major contributor" in that sort of exercise.

This user model is utilised by the system in order to:

- *Adapt the interaction with its users.* When a user wishes to find an exercise and decides to browse the available categories, s/he will see that in the categories' list the ones that s/he frequently explores are pre-selected for him/her by the system. Of course the instructor is free to select some other categories as well, or even ignore the already selected ones. In addition, if new exercises belonging to the categories that a particular user is interested in are added, the system informs the user when s/he logs in. When a user searches the collection of exercises using some keywords instead of selecting categories, his/her search is saved and the next time s/he logs in and wishes to search the collection s/he is presented with the option to run again the last saved search.
- *Promote co-operative or collaborative work.* Users are offered the choice of seeing what other users have done. When selecting this option, a user is presented with a list of exercises constructed by users who are considered by the system as "major contributors" in the categories that this specific user is considered "interested". A user is considered a "major contributor" in a particular area of s/he has created and/or updated many exercises in this area and these have been solved by a number of students. In addition, when an instructor constructs an exercise by himself/herself and before completing all the steps needed, the system checks if there is any similar exercise already constructed by another instructor who is considered "major contributor". If this is the case, the instructor is offered the choice of seeing the similar exercises and use them instead of constructing his/her own. In that way, the system avoids the repetition of exercises, facilitates the instructors' work and advances the co-operation and collaboration among them.

informs the “Instructor Model” about categories of exercises and provides problems to the “Problem Selector”.

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.

5 Conclusions and Future Work

In this paper we described a web based authoring tool for Intelligent Tutoring Systems in Algebra related domains. We showed that there is a need for an instructor modelling component in order to render the system more flexible and adaptable to particular instructors’ interests. The instructor modelling component interacts with other components of the system in order to form an implicit long-term user model of the instructor which is used for tailoring advice to the individual instructors’ interests. In this way instructors can easily benefit from the acquisition of domain problems of the system through various users of the web application.

It is within the future plans of this research to evaluate the instructor modelling function of WEAR using a variety of instructors in Algebra-related domains.

Acknowledgements

This work was partially supported by the Greek R&D project MITOS (EPET-II, Greek General Secretariat of Research & Technology). MITOS exploits the technologies of user modelling, information filtering, information extraction and data mining in order to provide users information from financial data, according to their interests. MITOS will be completed by March 2001.

References

1. Alexandris, N., Virvou, M., Moundridou, M.: 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. AACE, Charlottesville VA (1998) 1595-1597.
2. Brusilovsky, P., Schwarz, E., Weber, G.: ELM-ART: An intelligent tutoring system on World Wide Web. In: Frasson, C., Gauthier, G., Lesgold, A. (eds.): Proceedings of the 3rd International Conference on Intelligent Tutoring Systems, ITS-96. Lecture Notes in Computer Science, Vol. 1086. Springer-Verlag, Berlin (1996) 261-369
3. Burton, R.R., Brown, J.S.: A tutoring and student modelling paradigm for gaming environments. In: Colman, R., Lorton, P.Jr. (eds.): Computer Science and Education. ACM SIGCSE Bulletin, Vol. 8(1) (1976) 236-246

4. Eliot, C., Neiman, D., Lamar, M.: 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. AACE, Charlottesville VA (1997) 161-165
5. Hartley, J.R., Sleeman D.H.: Towards intelligent teaching systems. *Int. J. of Man-Machine Studies* 5 (1973) 215-236
6. Harvey, C.F., Smith, P., Lund, P.: Providing a networked future for interpersonal information retrieval: InfoVine and user modelling. *Interacting with Computers* 10 (1998) 195-212
7. McGraw, K. L.: Performance Support Systems: Integrating AI, Hypermedia and CBT to Enhance User Performance. *J. of Artificial Intelligence in Education* 5(1) (1994) 3-26
8. Major, N., Ainsworth, S., Wood, D.: REDEEM: Exploiting Symbiosis Between Psychology and Authoring Environments. *Int. J. of Artificial Intelligence in Education* 8 (1997) 317-340
9. Mark, M.A., Greer, J.E.: The VCR tutor: Evaluating instructional effectiveness. In: Proceedings of 13th Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates, Hillsdale, NJ (1991) 564-569
10. Munro, A., Johnson, M., Pizzini, Q., Surmon, D., Towne, D., Wogulis, J.: Authoring Simulation-centered tutors with RIDES. *Int. J. of Artificial Intelligence in Education* 8 (1997) 284-316
11. Murray, T.: Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *Int. J. of Artificial Intelligence in Education* 10 (1999) 98-129
12. Rich, E.: Users as Individuals: Individualizing User Models. *Int. J. of Man-Machine Studies* 18 (1983) 199-214
13. Ritter, S.: 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. ISIR (1997) 11-17
14. Shute, V., Glaser, R., Raghaven, K.: Inference and Discovery in an Exploratory Laboratory. In: Ackerman, P., Glaser, R. (eds.): *Learning and Individual Differences*. Freeman, San Francisco (1989) 279-326
15. Towne, D.: Approximate reasoning techniques for intelligent diagnostic instruction. *Int. J. of Artificial Intelligence in Education* 8 (1997) 262-283
16. Virvou, M., Moundridou, M.: 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. Lawrence Erlbaum Associates, Mahwah NJ (1999) 647-651
17. Virvou, M., Tsiriga, V.: 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. AACE, Charlottesville VA (1999) 933-938
18. Wenger, E.: *Artificial Intelligence and Tutoring Systems*. Morgan Kaufman, Los Altos CA (1987)
19. Woolf, B.P., Cunningham, P.A.: Multiple knowledge sources in intelligent teaching systems. *IEEE Expert* 2(2) (1987) 41-54