

# Authoring Intelligent Tutoring Systems over the World Wide Web

Maria Moundridou and Maria Virvou

**Abstract**—WEAR is a Web-based authoring tool for the construction of Intelligent Tutoring Systems (ITSs) in Algebra-related domains, such as physics, economics, chemistry, etc. In WEAR's authoring environment instructors are able to construct problems and tests and also build adaptive electronic textbooks. In return, WEAR generates an intelligent learning environment in which students can solve problems and study the topics of the curriculum. WEAR apart from modelling the student which is a common practice in almost all ITSs and ITS authoring tools, deals also with modelling the other class of its users: the instructors. Based on the user models it maintains, WEAR adapts the interaction with both students and instructors and provides them with individualised feedback and help.

**Index Terms**—Authoring tools, Intelligent Tutoring Systems, User Modelling.

## I. INTRODUCTION

ONE of the most effective ways for students to learn is to work with an expert tutor in an individual way. Bloom in his study described in [1], showed that an individual human tutor can improve student learning by two standard deviations over classroom instruction. This means that the average individually tutored student performs better than 98% of students receiving classroom instruction. Unfortunately, the realization of such an ideal educational setting is a utopia, due to the large number of expert instructors that would be needed.

The main goal of Intelligent Tutoring Systems (ITSs), which are computer-based instructional systems, is to provide each student with a learning experience similar to the ideal one-to-one tutoring. In particular, ITSs are based on Artificial Intelligence techniques and thus have the ability to present the teaching material in a flexible way and to provide learners with tailored instruction and feedback. A number of successful evaluations of ITSs (e.g. [2], [3]) have managed to show that such systems can be effective in improving learning by increasing the students' motivation and performance in comparison with traditional instructional methods.

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 [4]. In addition, teachers and educational researchers are encountering both unprecedented opportunities and

challenges to adapt networks to their classrooms and research fields [5]. 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 ITSs and Web-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 ([6]-[8]).

However, ITSs are still seen with scepticism due to the fact that they have not been extensively used in real educational settings such as workplaces and classrooms. The main reason for this limited use is probably the fact that the task of constructing an ITS is complex, time-consuming and involves a large number of people including programmers, instructors and experts of a specific domain. Moreover, once constructed, an ITS for a specific domain can not be re-used for different domains without spending much time and effort. An approach to simplifying the ITS construction is to develop ITS authoring tools/shells. The main aim of such systems is to provide an environment that can be used by a wider range of people to easily develop cost-effective ITSs. Furthermore, it would be even more useful to students and teachers to have authoring tools for ITSs on the WWW so that new courses could be created remotely by teachers and then students could remotely use the courses.

In the last decade a lot of research energy has been put in building ITS authoring tools; for a thorough and in-depth analysis of the state of the art for ITS authoring tools/shells the reader is referred to [9]. In this paper, Murray has classified the existing authoring tools based on their capabilities and concluded that they fall into two broad categories: the pedagogy-oriented authoring tools, which focus on how to sequence and teach relatively canned content and the performance-oriented ones, which focus on providing rich learning environments in which students can learn skills by practicing them and receiving feedback.

The system that we will describe in this paper is called WEAR. An earlier version of this system embodying fewer capabilities than those of the current version was described in [10]. WEAR 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, WEAR deals with the generation of instruction, since it offers the ability of problem construction and also the ability of building adaptive electronic textbooks. In that sense it shares the same focus with RIDES [11], an

---

M. Moundridou is with the Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou str., 18534 Piraeus, Greece (telephone: +3010-414-2133, e-mail: mariam@unipi.gr).

M. Virvou is with the Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou str., 18534 Piraeus, Greece (telephone: +3010-414-2269, e-mail: mvirvou@unipi.gr).

authoring system used for the construction of tutors that teach students how to operate devices through simulations. A system which adds capabilities to RIDES is DIAG [12], 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 WEAR performs student error diagnosis by providing a mechanism that can be applied to many algebra-related domains.

However, WEAR also shares capabilities with authoring tools belonging to the pedagogy-oriented category. In particular, WEAR gives instructors the ability to control the order by which students solve problems and study the teaching material by assigning a value to each problem's attribute called "level of difficulty" and by defining prerequisite relationships between topics of the electronic textbook. Therefore, WEAR is also concerned with managing the sequence of the curriculum on top of generating it. The former is a characteristic that can likewise be met in a system called REDEEM [13], which does not generate instruction but rather focuses on the representation of instructional expertise.

The users of ITS authoring tools are instructors who are responsible for the authoring procedure and learners who work with the produced ITSs. While learner modelling is a common task that is performed in almost every ITS and in many ITS authoring tools, instructor modelling has not gained any attention yet. This is an observation made also by Kinshuk and Patel in [14] who mention that user modelling as carried out in an ITS context is deficient, partly because it excludes the teacher by focusing solely on the student model. However, the role of instructors as users/authors of ITS authoring tools is very important for the effectiveness of the produced ITSs. In order for authoring tools to benefit the most from the involvement of instructors, they should provide individualised feedback to them throughout the ITS's life cycle. This can be achieved by an instructor modelling component incorporated in the architecture of the authoring tool [15].

Indeed, WEAR is an ITS authoring tool for the Web that models not only its students-users but also the instructors who author the ITSs to be generated [16]. Furthermore, WEAR's user models (instructor and student model) interact with each other by exchanging information [17]. This communication mimics in some sense the interaction that takes place in a real setting of a one-to-one tutoring: both the instructor and the student build models of each other and these models affect their attitude towards the learning process.

Concerning the Web-based ITSs that WEAR generates, their main intelligent features are the *interactive problem solving support* and the *adaptive navigation support* that are provided to students. Another characteristic of the system is the animated interface agent that it embodies in its student interface and which is responsible for communicating the system's messages to students.

This paper is a review of the current version of WEAR which incorporates new features, such as the adaptive navigation support provided to students and the instructor

modelling mechanisms that are used for offering intelligent and tailored assistance to instructors. In the main body of this paper we will describe the models that WEAR maintains in order to achieve its aims. We will then present WEAR's authoring and learning environments and the conclusions drawn from this work.

## II. MODELS OF WEAR

WEAR's main aims are: (i) to provide individualised support to instructors concerning the authoring of the course, and (ii) to adapt the interaction with each student working with the generated course. To achieve these aims, WEAR is based on three models: the domain model (representing knowledge about the domain of the subject matter), the student model (representing knowledge about the individual student) and the instructor model (representing knowledge about the instructor).

### A. Domain Model

The domain model containing knowledge about the subject matter is structured as a semantic network of hierarchically organised topics (textbook sections). Links between nodes of that network represent relationships between topics. At the moment, two types of relationship are used: *is\_prerequisite\_of*, to describe a topic a learner should know before accessing the more advanced one, and *is\_related\_to*, to describe that these two topics are in some way related to each other. Each topic has an associated difficulty level ranging from 1 (very easy) to 5 (very difficult). In addition, the domain model contains the variables, units of measure and equations that describe the specific domain (e.g. Table I illustrates the variables, units of measure and equations that describe a portion of the domain of physics).

TABLE I  
VARIABLES & EQUATIONS DESCRIBING A PORTION OF THE DOMAIN OF PHYSICS

Variable's description	Variable's name	Units of measure
Velocity	$v$	Meters per second (m/s)
Initial velocity	$v_0$	Meters per second (m/s)
Acceleration	$a$	Meters per second squared (m/s <sup>2</sup> )
Time	$t$	Seconds (s)
Force	$F$	Newtons (N)
Mass	$m$	Kilograms (kg)
Impulse	$J$	Kilograms*meters per second (kg*m/s)
<b>Equations</b>		
$F=m*a; J=m*v; v=v_0+a*t$		

Each variable constitutes a domain concept that is associated with nodes of the topic network. Finally, problems and/or tests examining the knowledge that must be acquired by studying a particular topic are associated with it; these associations are also part of the domain knowledge.

### B. Student Model

The student model that WEAR maintains is a combination of a stereotype and an overlay student model, similarly with other systems such as [18]. The stereotype student model (formed either directly by the instructor or after a preliminary test that has been posed to the student) classifies initially the student according to his/her knowledge of the domain and his/her mathematical skills.

As a result of this, each student is assigned to a stereotype (novice, beginner, intermediate or expert). The stereotype model defines initial values for the overlay student model. The latter is represented by a set of pairs “concept-value”. The concepts are domain concepts and concepts concerning the equation solving process (e.g. isolating the unknown variable in an equation). As has been mentioned in the previous section, domain concepts include domain variables and topics constituting the teaching material. For example, each variable presented in Table I constitutes a domain concept for the physics domain.

The value for each concept is an estimation of the student’s knowledge level of this concept and it is initialised by the stereotype student model. If, for example, the stereotype model indicates that a student is “intermediate” as to his/her mathematical skills and “beginner” as to his/her knowledge in the domain, then the concepts constituting the overlay student model are given the corresponding values: every concept that concerns the equation solving process and that has not been rated by the instructor as difficult or very difficult is considered known by the student; every domain concept rated as very easy is considered already known.

After the initialisation of each “concept-value” pair, the student model is updated taking into account the student’s performance in solving the problems associated with this concept and the reading or not of the corresponding teaching material. For example, if a student has successfully solved all problems evaluating the domain concept “acceleration –  $a$ ” and s/he has also read the corresponding topics of the electronic textbook, then in his/her student model the concept “acceleration” will hold the value 1 and thus it will be considered known.

During the process of solving a problem the student’s actions are monitored by the system and in case of an erroneous action the student’s answer is passed to the Student modeller, which is then responsible for diagnosing the cause of the error. The errors that are recognised by WEAR’s Student modeller are the following:

1. Domain errors. These include errors that are due to the student’s unfamiliarity with the domain being taught. For example, if a student enters the equation  $F=m/a$  instead of  $F=m*a$ , then the error is attributed to the category of “Domain errors” and in particular to the sub-category of “erroneous relationship between variables”.

2. Mathematical errors. These include errors that are due to the student’s lack of skills in solving mathematical equations. Such errors could be calculation errors, errors in isolating the unknown variable, etc. For example, if a student trying to isolate  $a$  in the equation  $F=m*a$  enters  $a=m/F$  instead of  $a=F/m$ , then the error is attributed to the category of “Mathematical errors” and in particular to the sub-category of “wrong isolation of the unknown variable”.

The student model finally holds information concerning the particular kinds of error each student makes. This information is used by WEAR 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 detailed progress reports of the student, which could be requested by the student

and/or the instructor.

- To provide 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.

### C. Instructor Model

The instructor modelling component monitors each instructor’s interactions with WEAR and constructs and/or updates his/her user model. In particular, the instructor aspects that are being modelled in WEAR, are the instructor’s preferences, usual activities, special interests and his/her level of expertise in teaching. These are described below.

In WEAR the instructor may give some long-term preferences as to whether s/he wishes the course to be difficult, average or easy or whether s/he wishes it to be very popular or fairly popular or whether s/he is not interested in this feature. Each of these preferences is associated with a percentage of failure in performances of class students and student interest in the course (e.g. how many times students visit the electronic textbook, and/or how many problems they have solved). The instructor may also state how important s/he considers each category of student error to be. In that way the students’ level of knowledge could be calculated according to the instructor’s preferences, assigning higher weight to those errors that the instructor has defined as more important.

Instructor’s activities that are frequent are also recorded in his/her long-term model. For example, if an instructor often constructs problems that belong to the same category then it is inferred that this particular instructor is a “major contributor” in that sort of problem and this usual activity of his/her is recorded in his/her user model. In addition, the instructor’s interests are inferred and recorded in the long-term instructor model. For example, if an instructor frequently searches for specific categories of problem then the inference made is that this instructor is “interested” in these categories of problem and this special interest of his/her is recorded in his/her user model.

Finally, in WEAR the instructor model records the teaching expertise of the instructor. This is explicitly stated by the instructor himself/herself. Each instructor may situate himself/herself in one of three categories: novice, having little experience and experienced. In the case of novice tutors and those having little experience, the authoring tool offers more detailed help concerning the teaching strategies that the tutor may select and shows him/her by default the results of the consistency checks.

The instructor model is utilised by the system in the following ways:

*To provide individualised help to the instructor.* For example, if an instructor has stated a long-term goal that s/he wishes to render the course popular within the class students then the authoring tool will examine whether the instructor’s short-term goals are consistent with his/her long-term goals. Student models provide information about how many students have attempted certain exercises and how many times they have seen certain lectures.

*To adapt the interaction with instructors.* When an

instructor wishes to find a problem 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. In addition, if new problems belonging to the categories that a particular user is interested in are added, the system informs the user when s/he logs in.

*To promote collaborative work among instructors.* Users are offered the choice of seeing what other users have done along two dimensions: the course structure and the constructed problems. Concerning the former, the information that is presented to the instructor is the structure of a similar course created by another instructor. In that way, instructors who may be novice as course designers could be assisted by more experienced peers who have previously used WEAR. When selecting to see problems constructed by others, the instructor is presented with a list of problems constructed by instructors who are considered by the system as "major contributors" in the categories that this specific instructor is considered "interested".

### III. WEAR'S OPERATION

WEAR functions in two different modes: the instructor's mode and the student's mode. The instructor's mode provides the environment of the authoring tool itself while the student's mode provides the environment for the ITS that WEAR produces. In the student's mode, students are presented with a number of problems to work on and are provided with individualised feedback while they are solving them. They also have at their disposal an electronic textbook and are offered navigation support adapted to their individual knowledge. In the instructor's mode the instructor is able to construct new problems, retrieve previously created ones and author the adaptive electronic textbook. In all cases, WEAR provides automatic assistance, as will be discussed in the subsequent sections.

#### A. The Authoring Environment

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.

When an instructor wishes to create problems 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 a problem. In particular, the procedure of constructing a problem is the following: The system displays every variable that the human instructor has entered when describing the domain and requests the unknown. 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 a simple problem text describing the given and asked data, which the instructor may change to make it more realistic

and comprehensible. The information concerning the known and unknown variables is used by WEAR to examine the domain equations and isolate the ones that are needed for the problem to be solved (Fig. 1). In some domains more than one equation exists that defines the same variable. In such cases, WEAR finds all the relevant equations to the variables selected. Then it presents them to the instructor to choose by himself/herself the appropriate ones for the problem s/he is currently constructing.

After the construction of a problem the tool lets the instructor preview the problem text and the solution of the problem 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.

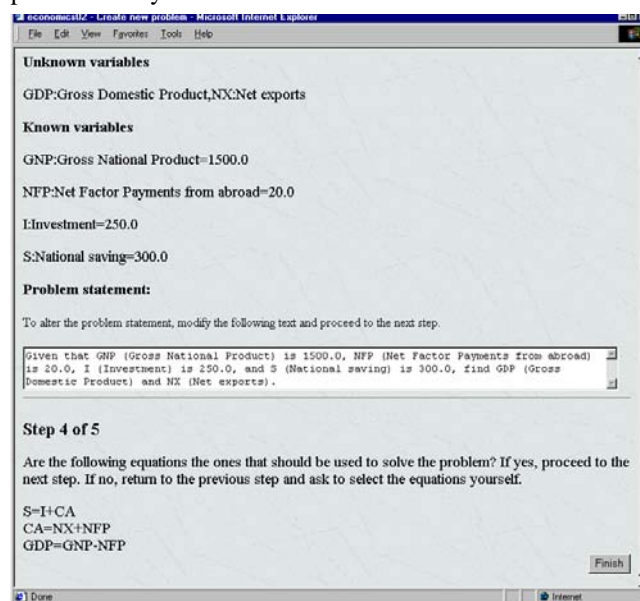


Fig. 1. Problem construction (economics domain)

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.

Beyond this kind of problem in which 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, WEAR also offers instructors the ability to create multiple-choice tests. Since there are topics of the curriculum that can not be

assessed by problems such as the above mentioned, by using multiple-choice tests instructors can be aware of their students' performance and understanding in these topics as well.

Beyond constructing a problem by himself/herself, the instructor has the ability to explore the problems constructed by others and choose the ones that s/he desires to be accessible by his/her class. Since new problems (belonging to different domains, involving different variables, etc.) can be continuously added to the system, there is no way for the system to have fixed categories of problem. Every time an instructor constructs a new problem the system performs this problem's categorisation based on some parameters. The problems are first categorised according to the domain to which they belong. At a second level the problems of each domain are categorised according to the variables they involve and their level of difficulty. Every variable of the domain can possibly form a problem category. For example, a problem like: "A force of 100 Newtons is acting on a 25 kg object, which is initially stable. After 10 seconds how much is 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 problem could also belong to the sub-category "level of difficulty 1" based on the problem's level of difficulty as this has been defined by the instructor.

Instructors are allowed either to browse the collection of problems by selecting the categories and sub-categories that match their needs and interests, or to search the entire collection using some keywords. As has been already mentioned, an instructor modelling mechanism incorporated in the system is responsible for tailoring the interaction of the instructors with the system to the instructors' needs.

Finally, WEAR allows the authoring of electronic textbooks by instructors and delivers them over the WWW to learners [19]. These textbooks offer navigation support to students, adapted to their individual needs and knowledge. The authoring procedure to create an adaptive electronic textbook with WEAR is quite simple. In particular, the instructor should prepare HTML files for the topics that would be contained in the electronic textbook. The next step is to use WEAR's facilities for uploading these files to the WEAR server. For each uploaded file the instructor must specify a title, a difficulty level and the position that it should have in the topics hierarchy. S/he should also relate topics to the domain variables. Finally, the instructor must edit the *is\_prerequisite\_of* and *is\_related\_to* relationships between topics. This information is used to form WEAR's domain model. The domain and student models are used by WEAR to generate a table of contents for each student. This table of contents consists of links to each topic of the textbook. These links are annotated in order to inform students about the educational appropriateness of the topic behind them.

When building an electronic textbook, instructors are provided with tools that verify the consistency of the course and report possible problems or errors, such as the case when the prerequisite relationships imply that a topic indirectly requires the knowledge of itself. To offer more

intelligent and individualised help WEAR relies on the information provided by the instructor modelling component that it embodies.

### B. The Learning Environment

Information obtained from student models, as well as the knowledge of the domain being taught, are exploited by WEAR to provide *adaptive navigation support* to students [20]. To achieve this, WEAR makes use of the adaptive link annotation technique: students interacting with the system see visual cues (different icons next to each link) that inform them about the current state both of the available problems and of the topics constituting the teaching material. This is done in order to facilitate the student's choice about which problem to solve next and which topic to study, as well as to provide them with information concerning the already mastered topics and concepts.



Fig. 2. Solving a problem while in student's mode

When a student attempts to solve a problem, the system provides an environment where the student gives the solution step by step. At first the student is presented with a problem statement like the one shown in Fig. 2. The student is requested to write down the equations that are needed to solve the problem and then s/he is requested to mathematically solve the problem. To detect the erroneous answers the system compares the student's solution to its own at every step. The system's solution is generated by WEAR's Problem Solver, which is implemented in PROLOG. The Problem Solver incorporates knowledge about how to solve systems of linear equations correctly and may generate the solution to a problem using information about the specific domain to which the problem belongs (e.g. physics). During the process of solving a problem the student's actions are monitored by the system. In case of an erroneous action, the Problem Solver passes the student's answer to the Student modeller, which is then responsible for diagnosing the cause of the error. Based on this diagnosis, the system provides the student with the appropriate feedback message.

The student interface includes an animated speaking character (Fig. 2) which is responsible for communicating the instructions and any feedback messages to the students [21].

#### IV. CONCLUSIONS

In this paper we described WEAR which is a Web-based authoring tool for the construction of Intelligent Tutoring Systems in Algebra-related domains. In WEAR's authoring environment instructors are able to construct problems and tests and also build adaptive electronic textbooks. In return, WEAR generates a learning environment in which students can solve problems and study the topics of the curriculum.

An important aspect of the system is its user modelling capabilities. In WEAR both classes of user (students and instructors) are being modelled, unlike what is happening in most ITS authoring tools that only model the students. Based on the user models it maintains, WEAR can tailor the interaction with each user.

WEAR's features both as a learning and as an authoring environment can be considered quite important, since there is still a shortage of intelligent tutoring systems functioning on the Web and a more serious shortage of Web-based authoring tools which can generate such systems.

#### REFERENCES

- [1] B. Bloom, "The 2 sigma problem: The search for methods of instruction as effective as one-to-one tutoring," *Educational Researcher*, vol. 13(6), pp. 4-16, 1984.
- [2] K.R. Koedinger, J.R. Anderson, W. Hadley, and M. Mark, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30-43, 1997.
- [3] V. Shute, and R. Glaser, "A large-scale evaluation of an intelligent discovery world: Smithtown," *Interactive Learning Environments*, vol. 1(1), pp. 51-77, 1990.
- [4] M. Berz, B. Erdelyi, and J. Hoefkens, "Experiences with interactive remote graduate instruction in beam physics," *Journal of Interactive Learning Research*, vol. 10(1), pp. 49-58, 1999.
- [5] C. Chou, "Developing CLUE: A formative evaluation system for computer network learning courseware," *Journal of Interactive Learning Research*, vol. 10(2), pp. 179-193, 1999.
- [6] P. Brusilovsky, E. Schwarz, and G. Weber, "ELM-ART: An intelligent tutoring system on World Wide Web," in *3rd International Conference on Intelligent Tutoring Systems*, Lecture Notes in Computer Science, vol. 1086, C. Frasson, G. Gauthier, and A. Lesgold, Eds. Berlin: Springer Verlag, 1996, pp. 261-269.
- [7] C. Eliot, D. Neiman, and M. Lamar, "Medtec: A Web-based intelligent tutor for basic anatomy," in *Proceedings of WebNet '97, World Conference of the WWW, Internet and Intranet*, S. Lobodzinski and I. Tomek Eds. Charlottesville, VA: AACE, 1997, pp. 161-165.
- [8] S. Ritter, "PAT Online: A model-tracing tutor on the world-wide web," in *Proceedings of Workshop Intelligent Educational Systems on the World Wide Web at 8th World Conference on Artificial Intelligence in Education*, P. Brusilovsky, K. Nakabayashi, and S. Ritter, Eds. Kobe, Japan: ISIR, 1997, pp. 11-17.
- [9] T. Murray, "Authoring Intelligent Tutoring Systems: An analysis of the state of the art," *International Journal of Artificial Intelligence in Education*, vol. 10, pp. 98-129, 1999.
- [10] M. Virvou, and M. Moundridou, "A web-based authoring tool for algebra-related Intelligent Tutoring Systems," *Educational Technology and Society*, vol. 3(2), pp. 61-70, 2000.
- [11] A. Munro, M. Johnson, Q. Pizzini, D. Surmon, D. Towne, and J. Wogulis, "Authoring simulation-centered tutors with RIDES," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 284-316, 1997.
- [12] D. Towne, "Approximate reasoning techniques for intelligent diagnostic instruction," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 262-283, 1997.
- [13] N. Major, S. Ainsworth, and D. Wood, "REDEEM: exploiting symbiosis between psychology and authoring environments," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 317-340, 1997.
- [14] Kinshuk, and A. Patel, "Intelligent tutoring tools: Redesigning ITSs for adequate knowledge transfer emphasis," in *Proceedings of 1996 International Conference on Intelligent and Cognitive Systems*, C. Lucas, Ed. Tehran: IPM, 1996, pp. 221-226.
- [15] M. Virvou, and M. Moundridou, "Adding an instructor modelling component to the architecture of ITS authoring tools," *International Journal of Artificial Intelligence in Education*, vol. 12, pp. 185-211, 2001.
- [16] M. Virvou, and M. Moundridou, "Modelling the instructor in a web-based authoring tool for algebra-related ITSs," in *Intelligent Tutoring Systems: Proceedings of the 5th International Conference*, Lecture Notes in Computer Science, vol. 1839, G. Gauthier, C. Frasson, and K. VanLehn, Eds. Berlin: Springer, 2000, pp. 635-644.
- [17] M. Virvou, and M. Moundridou, "Student and instructor models: two kinds of user model and their interaction in an ITS authoring tool," in *User Modeling 2001: Proceedings of the 8th International Conference*, Lecture Notes in Artificial Intelligence, vol. 2109, M. Bauer, P. Gmytrasiewicz, and J. Vassileva, Eds. Berlin: Springer, 2001, pp. 158-167.
- [18] H. Hohl, H. Böcker, and R. Gunzenhäuser, "Hypadapter: An adaptive hypertext system for exploratory learning and programming," *User Modeling and User Adapted Interaction*, vol. 6(2-3), pp. 131-155, 1996.
- [19] M. Moundridou, and M. Virvou, "Authoring and delivering adaptive Web-based textbooks using WEAR," in *IEEE International Conference on Advanced Learning Technologies; Issues, Achievements, and Challenges - ICALT 2001*, T. Okamoto, R. Hartley, Kinshuk, and J.P. Klus, Eds. Los Alamitos, California: IEEE Computer Society, 2001, pp. 185-188.
- [20] P. Brusilovsky, "Methods and techniques of adaptive hypermedia," *User Modeling and User-Adapted Interaction*, vol. 6(2-3), pp. 87-129, 1996.
- [21] M. Moundridou, and M. Virvou, "Evaluating the impact of interface agents in an Intelligent Tutoring Systems authoring tool," in *Advances in Human-Computer Interaction I: Proceedings of the Panhellenic Conference with International participation in Human-Computer interaction - PC-HCI 2001*, N. Avouris, and N. Fakotakis, Eds. Patras, Greece: Typorama Publications, 2001, pp. 371-376.