An evaluation agent that simulates students' behaviour in Intelligent Tutoring Systems*

Maria Virvou
Department of Informatics
University of Piraeus
Piraeus, Greece
mvirvou@unipi.gr

Konstantinos Manos
Department of Informatics
University of Piraeus
Piraeus, Greece
konstantinos@kman.gr

George Katsionis
Department of Informatics
University of Piraeus
Piraeus, Greece
gkatsion@singular.gr

Abstract – *Intelligent Tutoring Systems are meant to provide individualised tutoring to students by adapting the teaching material to their specific needs and abilities. However, their development is a hard task and the end-result is difficult to evaluate. In this paper we present a novel approach for the evaluation of these systems, which relies on an agent that may be used as a simulated student-user. The evaluation agent incorporates modelling techniques of real users that are based on both cognitive and temperamental data. The cognitive model is based on cognitive psychology and simulates the memorisation and retention capabilities of a student. The temperamental data creates an image of the student concerning the way s/he behaves and the kind of personality s/he has. Developers may evaluate the tutoring systems using the Agent rather than real students. Thus better quality of the end result may be achieved at no cost of the educational process.*

Keywords: Intelligent Tutoring Systems, Authoring Tools, student modelling, educational games.

1 Introduction

Educational software may serve the aims of education very effectively since it can assist students to learn and practice new skills without necessarily the presence of a human instructor. However, to benefit from educational software to the full, this software has to be included in the educational process and has to be designed very carefully. Indeed, a major issue is how to design an educational system that is beneficial to students. Towards this end, there is a need for the incorporation of reasoning aspects into educational software technology, so that the interactivity and individualisation abilities of the tutoring software may be maximised. Such reasoning abilities may be provided by Intelligent Tutoring Systems (ITSs).

ITSs have been quite good at providing dynamic aspects to the reasoning ability of educational applications. They have been shown to be effective at increasing students' motivation and performance in comparison with traditional learning methods and thus ITSs may significantly improve the learning outcomes [7, 8]. It has been widely agreed that an ITS should consist of four components, namely the domain knowledge, the student modelling component, the tutoring component and the user interface [10, 11]. In particular, the student modelling component contributes significantly to the individualisation of the electronic tutoring to each student's needs. Indeed, the student modelling component aims at gaining an understanding of what individual students know, how they learn and what their problems are while they learn. ITSs are mainly based on Artificial Intelligence (AI) techniques. AI *in* education (AIED) can offer ways to develop and test precise theories, and also important concepts relevant to individualised learning that have largely been overlooked by Education such as that of learner modeling; however AIED can scarcely claim to be *in* Education [3]. Indeed a common criticism on ITSs is that they may miss the mark in terms of task reality, feasibility and effectiveness [8].

In this paper, we address the problem of task reality, feasibility and effectiveness of ITSs by introducing a novel approach which is based on the reasoning capabilities of ITSs themselves. We argue that the development and evaluation process of ITSs can be assisted by extending the techniques used for student modelling in ITSs. To this end we have developed an evaluation component that can be used in ITSs. The evaluation component is an agent that acts as a simulated student and is meant to be used by instructors-authors to evaluate the ITSs that they have authored before these are delivered to real students. Thus, authors are given the opportunity to fine-tune the courses of the ITSs that they have created so as to have better results for the real students.

As a test-bed for our agent, we have used Ed-Game Author [12], an authoring tool for ITSs that operate as virtual reality games. An ITS authoring tool is a generalised framework for building ITSs along with a user interface that allows non-programmers (prospective

* **0-7803-7952-7/03/$17.00 © 2003 IEEE.**

authors) to formalise and visualise their knowledge [9]. Ed-Game Author is meant to be used primarily by authors-instructors who may author their courses and then by students who are going to use the resulting courses. It offers multiple virtual reality game environments and the basic story of these games. Indeed, recently a lot of researchers are convinced that education may benefit a lot from the incorporation of computer games into it (e.g. [5], [2], [1], [6]). Ed-Game Author also incorporates a learner modelling mechanism that builds the individual profile of each player who is also a learner. Then, instructors may insert the material that they wish to teach to students.

The present research work has led to considerable enhancements of the learner modelling component of Ed-Game Author so that it can constantly make observations about the students' behaviour. These observations mainly concern the way students respond to assessment questions in terms of the quality and correctness of their answers. For example, whether a student's answer was correct, whether the answer was given with certainty or with hesitation and if the answer was wrong whether this kind of mistake is a frequent one for the particular student etc.

The evaluation agent that has been incorporated in Ed-Game Author is an application that given a learner model, starts "playing" the virtual lesson inside the ITS, simulating a real user's reactions.

2 Reasoning of the evaluation agent

The evaluation agent is constructed for a sole purpose: To be able to simulate a user in any aspect inside the ITS. To accomplish this the Agent should have adequate information so as to be able to mimic the student's actions inside the system. This information is stored in the learner's individual model within the ITS. Obviously, to have a solid learner profile, that is stable and has adequate information for the Agent, a student must have interacted and used the system for at least a couple of sessions, during which time the digital image of the student is being composed (and stored in the learner model).

The kind of information that is needed for the creation of the digital image of a student, is determined by the way a student's image is perceived by the ITS. This means that a student model is determined by the way that the system actually understands the term "student". From the system's point of view the "student-user" is nothing more than digital-binary input, either from the mouse (clicking on the screen) or from the keyboard. To be more exact this is a very low-level image of a "student-user". If we see it from the ITS's layer then the "student-user" represents input of very specific type, for example: students' answers to questions, movement inside the Virtual Environment, responses to the system's interaction etc. For the composition of the "student-user's" digital image the ITS needs to keep data for each category of input that it may "understand" and may use. Such kind of data may be classified into two major categories: The "Temperamental" data and the "Cognitive" data. The "Temperamental" data include all the information that is needed to mimic the student's reaction inside the Virtual Environment of an ITS game. The "Cognitive" data have to do with the student's mental capabilities and his/her knowledge level of the domain.

The reason why both categories of information are needed, is the fact that the evaluation agent needs to simulate both the knowledge level of a student and the way s/he learns. From the way that a student learns depending on the type of person that s/he is and his/her cognitive abilities, the instructor may understand how motivating and educationally effective a course is.

3 Temperamental model

Temperamental data is connected to the way that a student behaves and responds to the system. In the case of Ed-Game Author the students' behaviour is related to the way that the student plays the educational game in the virtual world.

The features of virtual reality games include dungeons, dragons, castles, keys etc. In these games the student-player tries to reach the "land of knowledge" and find the treasure, which is hidden there. The difference of these games with other commercial games is that in these games one must fight one's way through by using one's knowledge. However, to achieve this, the player has to obtain a good score, which is accumulated while the player navigates through the virtual world and answers questions concerning the domain being taught.

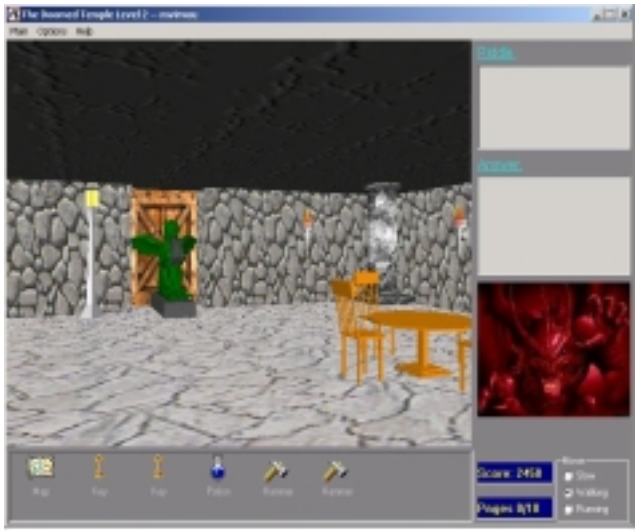

Figure 1: A screenshot of the game.

In the game worlds there are animated agents that communicate with the players. There are three types of animated agent, the advisor, the guard of a passage and the student's companion. Animated agents, who act as advisors, lead the student to lessons that s/he has to read. Animated agents, who act as companions are responsible for showing empathy to the students and help them in managing their emotions while playing and answering questions. On the other hand, animated agents who act as guards of passages ask questions to players. These questions have to be answered correctly by the students so that they are allowed to continue their way into the passage and receive more points for their total score. An example of an animated agent who acts as a guard is the dragon, which is illustrated in Figure 1.

Temperamental data include:

- The way that the user walks around the virtual world. Is s/he lost easily inside the labyrinth? Does s/he keep walking around the same places (maybe looking for something or just checking around)? Does s/he take a lot of time at specific locations (maybe staring the surroundings and thus being distracted from the system)?

- How familiar s/he is with the use of a computer. While moving around the Virtual Environment does s/he keep bumping on physical obstacles? Does s/he take a lot of time to navigate through the user interface?

Time plays an important role in temperamental measurements. There are many inferences that can be drawn for the students' feelings and reactions depending on the time they spend before and after they make some actions. Some examples of inferences based on observations on time spent for various activities are the following:

- The time that a student takes to answer a question. This measures the degree of speed of the student.

- Pausing time after a system's response. The time the computer is left idle after a response to the student is used to measure the degree of surprise that the response may have caused to the student. At the first five times that the student answers questions the system measures this pausing time and calculates an average. After that, the system has a measure about whether a student's pausing time is out of ordinary and may have been caused by surprise.

In addition, certain patterns of actions are used to show aspects of the students' cognitive and emotional state. Some examples of students' actions that are used as evidence are the following:

- The number of times that a student presses the "backspace" and "delete" button while forming an answer. This evidence is used to measure the degree of certainty of the student concerning a particular answer; the more times the student presses "backspace" and "delete" the less certain s/he is about the answer. If the student consistently hesitates and does not seem certain about his/her answers irrespective of their correctness, then this may reflect a personality attribute of lack of self-confidence. On the other hand, if lack of certainty is occasional then this probably means that the student does not probably know the particular piece of the domain that is related to the exam question that the student has answered.

- Mouse movements without any obvious intent in the Virtual Reality space of the game. This kind of evidence is mainly connected to the degree of concentration or frustration or intimidation of the student; the more mouse movements without any obvious intent, the less concentrated or the more frustrated or intimidated the student is.

In some cases, inferences are drawn from the combination of two different categories of evidence. For example, the degree of determination is calculated as the means of the degree of speed and the degree of certainty of a student.

4 Cognitive model

Cognitive data include:

- What the level of the student's knowledge is. This is measured by keeping track of the right and wrong answers s/he gives and the time that s/he needs to give a correct one. Moreover, since the ITS provides help through the use of virtual tutors, a metric is also provided from the number of times the student uses a tutor to help him/her answer a question.

- The student's retention capabilities. The agent incorporates a cognitive model, which is based on cognitive psychology. This model calculates and simulates the retention and memorization capabilities of a student and gives the teacher an insight on the proportion of the information that is actually learnt by a student-player, during the Virtual Lesson.

- The ITS also keeps track of specific statistics that have to do with the causes underlying an error, for example whether the student made a typographic error, or a syntactical or a spelling one, etc These metrics are also stored in the student model.

4.1 Retention and memorization capabilities

The simulated student-player incorporates a cognitive model that keeps track of the students' memory of facts that have been taught to them. For this reason, principles of cognitive psychology have been adapted and incorporated into the system. As a result, the educational application takes into account the time that has passed since the learning of a fact has occurred and combines this information with evidence from each individual student's actions. Such evidence includes how easily a student can memorise new facts and how well s/he can answer questions concerning the material being taught. In this way the system may know when each individual student may need to revise each part of the theory being taught.

The cognitive model is based on a classical approach about how people forget that has been introduced by Ebbinghaus [4]. Ebbinghaus' empirical research led him to the creation of a mathematical formula which calculates an approximation of how much may be remembered by an individual in relation to the time from the end of learning (Formula 1).

$$b = \frac{100 * k}{(\log t)^c + k} \quad (1)$$

Where:

- t: is the time in minutes counting from one minute before the end of the learning
- b: the equivalent of the amount remembered from the first learning.
- c and k : two constants with the following calculated values: k = 1.84 and c = 1.25

In the cognitive model of the simulated student-player the Ebbinghaus calculations have been the basis for finding out how much is remembered by an average student. In particular, there is a database that simulates the mental library of the student. Each fact a student encounters during the game-lesson is stored in this database as a record. At each time a Retention Factor (RF) may be calculated for each fact. The RF represents the student's memory state at that time.

A further enhancement of the student model is the addition of a new factor, the Student's Retention Factor (SRF). The initial Retention Factor (RF) is used to measure the portion of a fact that is actually remembered by the student after a specific time interval. Although the actual calculation of the RF has been modified to take into account a portion of the existing student model, the factor continues to be based mostly on the Ebbinghaus' mathematical formula, which is very general and does not take account the particular circumstances of an individual student model. Experiments have shown that this is not enough, and that each student tends to have a very personal way of reacting inside the system. More specifically, the level of retention in previous studies was considered to be static and equal to 75%. If a fact had an RF value equal or higher than this level of retention, it was considered to have been successfully memorised by a student. The result of the research conducted was that each student's base level of retention varies, so with the use of questionnaires, after the end of the lesson, we checked the facts that were actually learnt and compared these results with the ones provided by the cognitive model. From this procedure the SRF was created, which represented a personalised level of retention for each student.

4.2 Diagnostics

The educational games perform error diagnosis and record all errors that the student may have made. Thus the system records a detailed report for every student in the student model. While the educational game examines the knowledge of students it can distinguish between spelling mistakes, typing/keyboard mistakes and errors that are due to lack of domain knowledge. For example, if the student types an answer, which contains an extra letter, in comparison with the correct one then it has probably been a typing error. If the student types an answer that contains a letter substituting the correct one, which is near the correct one on the keyboard then it has probably been a keyboard error. If the student types an erroneous answer that is pronounced in a similar way as the correct one then s/he has probably made a spelling error. The results of these answers are then kept in the statistical part of the system. In particular, the students' errors are kept and classified in different categories depending on their underlying cause.

5 The overall architecture

In our approach and implementation, we use the functionality provided by the new technology of the Web Services. From the perspective of a system's architecture, Web services are a collection of procedures and/or functions that have the ability to be called remotely by any external system. What the system may gain from this new technology is greater scalability and flexibility.

By using web services, the system can be cut down to relatively small independent pieces, and then distributed along the network. For the simulated student-player we

have created a web service from all the input procedures and functions of an ITS, making the system's core independent of the source of input.

The cognitive and temperamental models have also been implemented as two Web Services. Using the flexibility that this new technology provides, the models are expandable while at the same time they can be used with almost no effort from various other modules. Web services also provide an important advantage, the fact that they can run over the Internet. In that way, the models and the ITS can actually be in different machines across the web. This is extremely useful for the authors who may try their courses on simulated student-players using real students' profiles which may reside in different PCs.

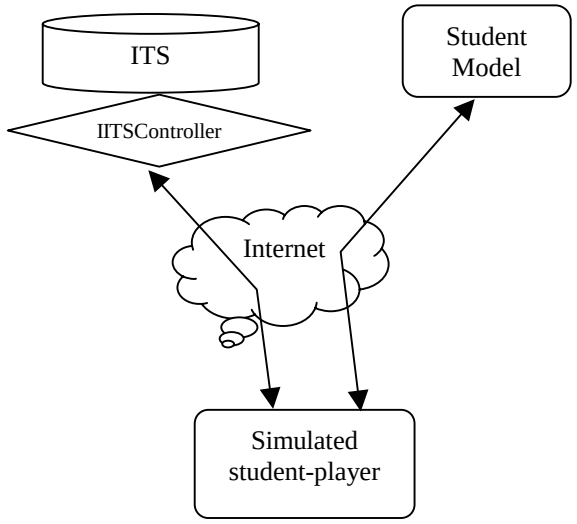

Figure 2: The overall architecture

The implementation of the simulated student-player requires computer-computer interaction rather than human-computer interaction that is needed for the user-interface of a standard ITS. Thus the input from the mouse or keyboard, which is suitable for real students, is not suitable for the simulated student-agent. For this reason, another interface for the ITSs was created which was called IITSController (Interface Intelligent Tutoring System Controller). IITSController is a web service provided by the ITS, that gives the caller full control over the virtual lesson's interface. This means that one can implement a program, which can reproduce any input for the system that would otherwise be generated from the mouse or the keyboard. Moreover since IITSController is a web service, it may be called by any computer in the web, so an application (in our case the agent) may invoke remotely the ITS system.

By using this architecture the system's core no longer "knows" who/what is providing the input, because this is not of any interest to it. As a result, it still works, calculating user profiles and statistic results but irrespective of who/what is interacting with it in the local computer. Both the simulated student-player and a real student-player are regarded as valid users of the ITS and are treated in the same way.

The simulated student-player consists of a core module and two proxy classes, one that communicates with the ITS and one that accesses the cognitive model. At this point, we should point out that the agent itself is implemented as a web service, thus it can be invoked and used through the Internet.

The agent takes as input a web service for the cognitive model, an IITSController, and an ASCII file containing the student's profile as it has been compiled through the last couple of virtual sessions. A graphical representation of the system's architecture can be seen in Figure 2. The agent uses the information stored inside the student's profile to simulate the student's actions.

6 Iterations of the authoring process

During the authoring process, after the teacher finishes describing a new virtual world s/he may ask the simulated student-player, which acts as an evaluation agent, to "play" the virtual game using different student profiles. These student profiles contain long-term characteristics of real student-players that have played in other parts of the game-course, which had been previously authored by the instructor and used by students. These profiles may have been stored in the students' PCs which may be different from the author's PC. However, they can be collected through the Internet.

In the end the teacher views the results and may choose to modify the virtual world's content, so as to emphasise some parts of the theory more than others, or s/he may find a mistake in the flow of the lesson which s/he may wish to correct. With that tool, the teacher may actually have a measure of the virtual lesson's efficiency before taking it to class. This allows an iteration of the authoring process of the ITS and thus ensures better quality of the resulting educational application. Thus, the life-cycle of the educational games that are created by instructors may contain several iterations as illustrated in Figure 3. Each iteration improves the previous version of the system and leads to better quality at no cost for the educational process.

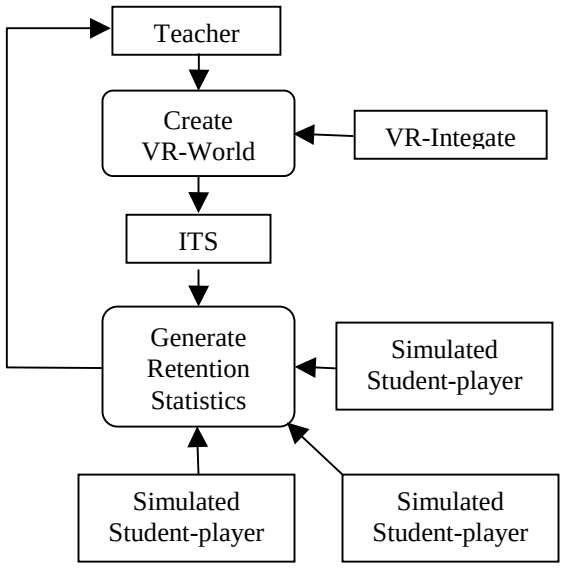

Figure 3: Iterations of the authoring process

7 Conclusions

In this paper, we have presented and discussed an evaluation agent that can be used in the context of an ITS authoring tool. This component can be particularly useful to instructors-authors who can evaluate the courses that they have created using the evaluation agent rather than real students. In this way, instructors may easily identify possible deficiencies of the courses that they have constructed and thus they may make the necessary amendments before the courses are delivered to real students. This process may allow instructors to produce courses of very high quality to the benefit of education.

The evaluation agent uses data from real students in order to imitate their behaviour while they learn. Moreover it uses theories from cognitive psychology to gain an understanding of how much new knowledge a student may learn and remember after each lesson. For these purposes, the evaluation agent described uses both temperamental data and cognitive data concerning students in order to create a simulation of them.

References

[1] Amory, A., Naicker, K., Vincent, J. & Claudia, A. (1998). Computer Games as a Learning Resource. Proceedings of ED-MEDIA, ED-TELECOM 98, World Conference on Education Multimedia and Educational Telecommunications, vol. 1, pp. 50-55.

[2] Boyle T. (1997): "Design for Multimedia Learning". Prentice Hall.

[3] Cumming, G. & McDougall, A. (2000) "Mainstreaming AIED into Education?" International Journal of Artificial Intelligence in Education, vol. 11, pp. 197-207.

[4] Ebbinghaus, H. (1998) "Classics in Psychology, 1885: Vol. 20, Memory", R.H. Wozniak (Ed.), Thoemmes Press, 1998

[5] Inkpen, K., Upitis, R., Klawe, M., Lawry, J., Anderson, A., Mutindi, N., Sedighian, K., Leroux, S. & Hsu, D (1994): "We Have Never-Forgetful Flowers In Our Garden: Girl's Responses to Electronic Games". Journal of Computers in Math and Science Teaching, 13(4), pp. 383-403.

[6] Jayakanthan R. (2002): "Application of computer games in the field of education" Electronic Library vol. 20(2), pp. 98-102.

[7] Mark, M.A. & Greer, J. E. (1991). "The VCR Tutor: Evaluating instructional effectiveness" In Hammond K.J. & Gentner D. Q. (Eds.) Proceedings of the 13th Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 564-569.

[8] McGraw, K.L. (1994) "Performance Support Systems: Integrating AI, Hypermedia and CBT to enhance user performance" International Journal of Artificial Intelligence in Education, vol 5(1), pp. 3-26.

[9] Murray, T. (1999). "Authoring intelligent tutoring systems: an analysis of the state of the art", International Journal of Artificial Intelligence in Education, vol. 10, pp. 98-129.

[10] Self J. (1999) 'The Defining Characteristics of Intelligent Tutoring Systems Research: ITSs Care, Precisely', International Journal of Artificial Intelligence in Education, vol. 10, pp. 350-364.

[11] 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, pp. 279-326.

[12] Virvou, M., Manos, K. & Katsionis, G. (2002) "Incorporating the culture of Virtual Reality Games into educational software via an authoring tool" Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics (IEEE Catalog number: 02CH37349C, ISBN: 0-7803-7438-X).

[13] Wenger E. (1987) 'Artificial Intelligence and Tutoring Systems', Morgan Kaufmann.