# Multi-application player modelling and instruction provided by a knowledge-based authoring tool for educational games

Maria VIRVOU, Konstantinos MANOS, George KATSIONIS
*Department of Informatics, University of Piraeus,*
*80, Karaoli and Dimitriou St., Piraeus 185 34, Greece*
*mvirvou@unipi.gr; konstantinos@kman.gr; gkatsion@kman.gr*

**Abstract.** This paper is about an authoring tool for Intelligent Tutoring Systems (ITS) that generates virtual reality educational games. The ITS common practice of modelling the student has been expanded to include characteristics of players such as their level of game-playing competence irrespective of the domain of the educational game. Such characteristics are domain-independent and they may affect the overall performance of a student-player. Domain-independent player characteristics are kept in user models separately from domain-dependent characteristics such as the level of knowledge of a student in a particular domain. In this way, domain-independent features may be used for modelling an individual student across many application domains and adapt advice accordingly.

## 1. Introduction

One of the main aims of education is to engage students in learning. However, learning is a hard cognitive task that may be very tiring for students and thus requires a lot of energy from them. In this respect, presentation of the teaching material plays an important role in stimulating students and keeping them alert. Therefore educational software researchers seek new technology means to improve the appeal and effectiveness of educational applications. However, there is a whole imposing culture of computer games that has not been exploited sufficiently yet for the purposes of education, although it is widely acknowledged that children and adolescents are fascinated by computer games. Indeed, a large number of research studies show that computer game playing is by far the most popular activity on the computer for children and adolescents (e.g. [1], [2], [3] and [4]).

However, bringing the culture of computer games in class cannot be effective by itself if this is not preceded by careful design and if human instructors have not been convinced that this will help the teaching process. Combining entertainment with education can be very promising but it can also create problems. One major problem is that educational games should be addressed to all kinds of students irrespective of their skill level in playing computer games; despite the great popularity of games there are still a lot of children that do not know how to play. Another problem is that of all categories of educational software, which have to be highly interactive with the students. There must be underlying reasoning mechanisms in the educational game that may ensure individualised interaction based on the actions of the student-players, otherwise the game runs the risk of providing little educational benefit. The technology of Intelligent Tutoring Systems (ITSs) may provide such reasoning mechanisms. ITSs are computer-based learning systems, which

attempt to adapt to the needs of students. It is simple logic that response individualised to a particular student must be based on some information about that student; in ITSs this realisation led to student modelling, which became a core or even defining issue for the field [5]. Thus the combination of ITSs with the virtual reality environment of the games can render an educational application both highly adaptive to students' needs and attractive for them.

However, despite their nice features, few ITSs have seen widespread use, mainly because the development of ITSs is difficult, time-consuming and costly [6]. If this is the case for an ITS with a simple user interface then the difficulty of constructing a learning environment will be far greater if the ITS has to be combined with a demanding interface of a virtual reality game. Resolving the problem of constructing such sophisticated educational applications has been the main motivation for this research. This research concerns the development of a knowledge-based authoring tool that can provide ITSs, which operate as computer games. Knowledge-based authoring tools are meant to be used by instructors who wish to author their own ITSs on a certain domain. Murray [7] highlights the potential of ITS authoring tools in giving the instructional designer a combination of facilities to produce visually appealing, interactive screens and a deep representation of content and pedagogy. Authoring tools have to be used for multiple domains.

In this paper, a knowledge-based authoring tool is described. The authoring tool is called VR-MultiAuthor. VR-MultiAuthor provides an authoring environment to instructors who wish to create ITSs that operate through a virtual reality game. The concept of the game is used so that the ITSs may become more motivating and engaging. Moreover the ITSs are able to provide diagnostic reasoning concerning the students' answers to questions about the domain being taught.

In particular, this paper focuses on the student modelling component, which in the case of VR-MultiAuthor is extended to include characteristics of the students as players. This is done because players of educational games should be modelled in two different aspects: their knowledge of the domain being taught and their competence level in playing the particular virtual reality adventure game. A student's game playing skill may affect his/her overall performance. For example, a novice player of virtual reality adventure games may not gain a good score because s/he may be lost while navigating through the game. This has to be known by the educational application that aims primarily at educating players so that the domain knowledge of the student-player is not underestimated. Moreover, in contrast with the level of domain knowledge of a student-player, the game playing skill level of a player is domain-independent. Therefore this information about a particular player may be reusable in many application domains. Thus it is stored in a separate part of the student model and can be used by all applications of different domains that are created using VR-MultiAuthor.

## 2. Operation of the Authoring Tool

The authoring tool operates at two levels, the authoring level and the game level. At the authoring level, human instructors can provide the domain-knowledge of their courses and thus create their own ITS-games through the authoring tool. At the game level the created ITS-games may be used by students who can learn while playing. The authoring tool provides the facility for the construction of student-player modelling components for the resulting ITS-games, henceforth referred to as player modelling components.

The player modelling component observes the behaviour of student-players within the educational applications and examines the correctness of the players' answers in terms of the players' factual knowledge and reasoning that they have used. The diagnostic process is based on

previous research in error diagnosis in the context of a single ITS-game for learning geography [8]. However, VR-MultiAuthor extends significantly this research by creating an authoring tool that can generate ITS-games for multiple domains and provide an integrated environment where all the generated ITS-games can collect and exchange information among them about individual students.

The initial input to the authoring tool is given by a human tutor who is acting as an author. The initial input consists of lessons and tests accompanying these lessons. Tests may consist of questions of multiple-choice questions, True/False questions or questions where the student has to type in the answer. All tests are part of the story of the virtual reality game. In all types of questions the instructor has the ability to associate erroneous answers with particular causes and explanations of errors so that these may be used by the system to give more detailed and informative feedback to players. Moreover, these explanations are recorded in each player's profile, which is updated after each session of the player with the educational application. Player features are permanently recorded in the long-term player model [9], [10]. The long-term player model represents the player's knowledge in the domain being taught, his/her game playing skill level and other features concerning his/her behaviour (e.g. how often s/he may be absent-minded etc.).

The interaction of the student with the resulting educational applications is performed through a virtual reality game. The environment of the games generated by VR-MultiAuthor is similar to that of the popular game called "DOOM" [11] which has many virtual theme worlds with castles and dragons that the player has to navigate through and achieve the goal of reaching the exit. In the case of VR-MultiAuthor, questions to students are always asked by animated agents. Then, students may type their answer in a dialog box. In the resulting educational applications, the system communicates with the student via three types of animated agent, the virtual enemy, the virtual advisor and the virtual companion. All three types of animated agent use synthesised voice as well as written messages. The reason that the animated agents use voice is that there are studies that show that voice messages may be more effective than written ones in the way that students react to the educational applications (e.g. [12]).

The ultimate goal of a student-player, is to navigate through a virtual world and climb up the "mountain of knowledge", which is at the end of the world. To navigate through the virtual world a player has to open closed doors, which are guarded by dragons. Examples of a virtual water world and a virtual fire world are illustrated in Figure 1 and 2 respectively. A guard dragon poses a question to the player from the domain of the particular educational application that has been created via VR-MultiAuthor. If the player gives a correct answer then the dragon allows him/her to take the key to open the guarded door and the player receives full marks for his/her answer.

If the player gives an answer, which is not correct then the system performs error diagnosis so that it can find out what the cause of the error has been. The explanation of a mistake may be difficult for a system to spot. Hollnagel [13, 14] makes an important distinction between the underlying cause or genotype of an error and the observable manifestation or phenotype of the error. In addition, ambiguity may be a problem, since there may be different explanations of observed incorrect users' actions (Mitrovic [15]).

For example, a student may give an erroneous answer due to a typing or spelling error and may appear that the student does not know the answer in the domain being taught. If the system decides that the error has been due to a typing mistake then the error is considered superficial and the player receives some marks (although not full marks) for his/her answer. However, if a player types a totally irrelevant answer then this is considered a serious error and the player does not receive any marks at all. In cases where there is an ambiguity as to what the underlying cause of

an error has been, the system consults the player's long-term model. For example, if a player has made an error that could either be a spelling mistake or a serious error then the long-term player model will resolve the ambiguity. If this player has been consistently making a lot of spelling mistakes when s/he is typing answers to questions posed but s/he does not usually make serious errors, then the spelling mistake will be favoured as the most probable cause of the error and the player will be given the benefit of the doubt.

At times a player is given a key as a bonus, in which case s/he will not have to answer a question to get through a guarded door. In such cases the bonus-key is kept in the player's inventory list to be used by the player in a difficult situation where s/he does not know an answer posed to him/her by a dragon. As part of the adventure of the game the player may also come across certain objects where s/he may click on. These objects appear at random and give hints to students or guide them to read another part of the domain being taught. For example, Figure 2 illustrates a hint in the form of a blue ball in a virtual volcano world. However, these hints or the parts of the theory that are visited, are not immediately usable by the students, since they refer to questions that the students will have to answer at a location of the virtual world other than the one that they are currently at. Hence, the students will have to remember these hints or parts of the theory so that they may use them when the time comes. Educationally, these hints motivate students to read and memorise important parts of the theory.

## 3. Domain-Independent vs Domain-Dependent Player Modelling

The authoring tool creates an integrated learning environment where multiple human instructors can contribute the respective domain knowledge of the lesson they teach. The integrated environment aims at being used by students of the same class of a school. Such students normally attend a variety of courses of different domains such as geography, biology, mathematics etc. In this way each student may play games for more than one course that s/he is being taught at school. This means that a particular student who is going to play in all available games has to log-in to identify himself/herself. Then each game contributes both domain-dependent and domain-independent information about the particular player to his/her individual long-term player model. Then in turn, the individual long-term player model may be used by all applications to resolve ambiguities about errors and to provide personalised advice.

**Figure 1**: Virtual water world

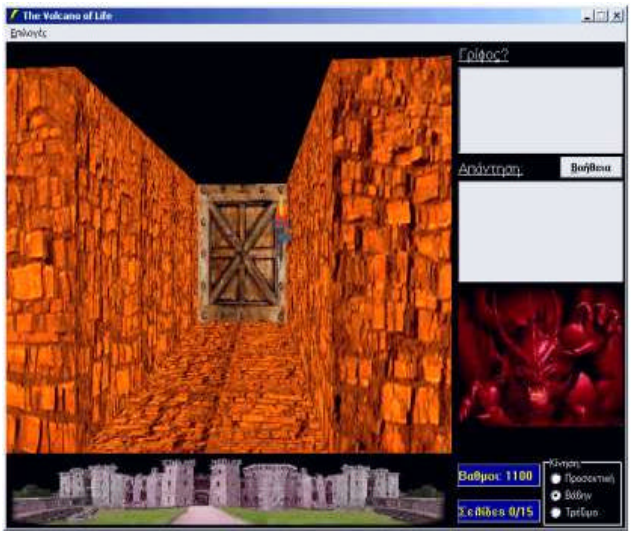


**Figure 2**. Virtual fire world

For example, a student may be consistently facing navigation problems while s/he is moving through a virtual world. This is a domain-independent feature of the player, which remains the same in all the domain-applications that have resulted from VR-MutiAuthor. This kind of feature is recorded in the long-term player model and is updated constantly in all domains. This feature is taken into account so that the particular student receives more automatic help on how to navigate through the world. On the other hand, when a student knows all the answers in a particular domain, this is a domain-dependent feature concerning his/her level of knowledge in the particular domain. For example, a student's knowledge level in history does not relate with his/her knowledge level in Mathematics.

Domain-independent features mainly concern the player's level of game playing skill, which should not interfere with the player's level of knowledge in the domain being taught. There are also other domain-independent player features, such as the player's proneness in making typing mistakes, spelling mistakes, his/her level of concentration while playing etc. Some important features of students that concern their game playing skill are the following:

1. Virtual Reality User Interface Acquaintance

This feature measures the player's level of acquaintance with the Virtual Reality User Interface of the application. In this feature the system measures whether the player knows the functionality of user interface features such as the "Inventory", the "Tutor" etc., and how to read/answer the various questions which are presented in the relevant section of the screen. For example, supposing a case where a player faces a question that he cannot answer (s/he tried at least twice) and the player retreats and starts wandering around in the virtual world although there is a key inside the player's inventory, then this player is probably ignorant of his inventory's usage.

2. Navigational Effort

Not all users know how to play a 3D Virtual Reality Game. This feature measures how well the user can navigate through the Virtual World. This feature is calculated taking into account the frequency of occurrence of certain actions of the player inside a virtual world. Such actions are the

character's bumping onto a wall (or other virtual items), aimless rotation around the same position, which shows that the player tries to find his/her way out of a point where s/he has been stuck, etc.

3. VR Environment Distractions

There are many cases when the Virtual Environment draws the player's attention so much that s/he may miss the main point of the educational game (which is learning a specific subject). This is the case in situations such as when a player finds a door, does not answer the riddle, goes back to the previous encountered tutor (angel), reads the hint, goes to the door, does not answer the riddle, goes again to the tutor etc. This behaviour shows that from the tutor to the door the player may have been so distracted that s/he forgot the hint.

However, naturally there are other domain-dependent errors, which may only be made in the corresponding domain game. For each domain there are pre-defined categories of error that the system knows about. These categories have been inserted into the system by the human tutor who acted as author of the particular domain application. For example, in an educational application about modern history, a student may have made 10 errors in questions concerning the period 1900-1945 and none in questions concerning the period 1850-1900. In this case, the system will record the fact that the student has a serious lack of knowledge about the particular history period and will compare this finding with the findings of previous interactions and future interactions to determine how the student is progressing.

Domain dependent information about a player is kept separately in the corresponding domain application. On the other hand, domain independent information about each student is kept in the integrated learning environment so that it may be used and updated by all available domain applications. The interaction of these parts of the player model is illustrated in Figure 3.

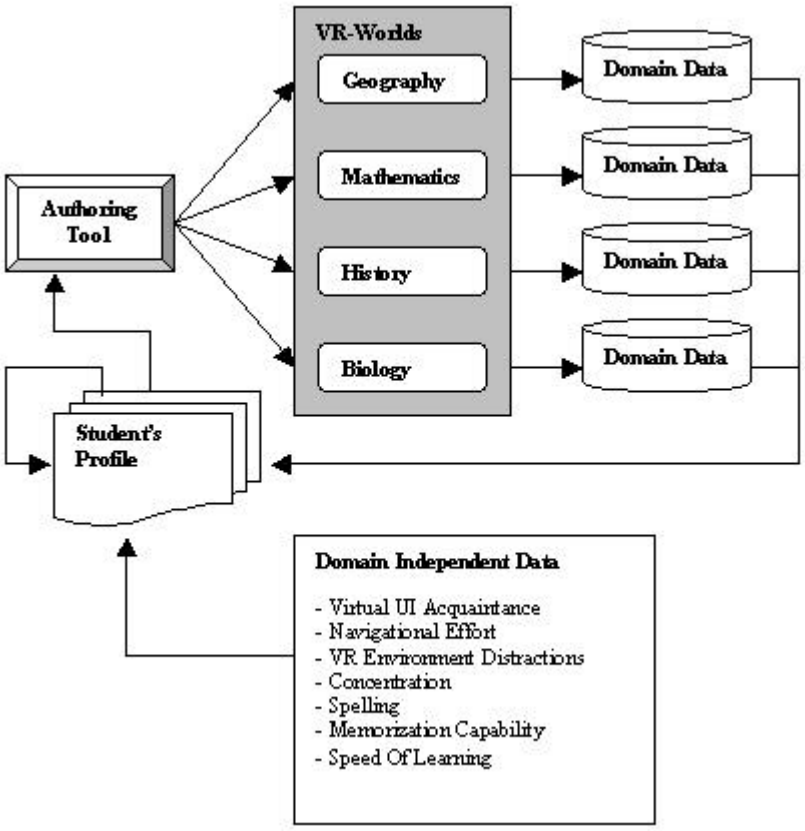

**Figure 3**. Interaction of domain-dependent and domain-independent parts of player models

## 4. Conclusions

In this paper we presented VR-MultiAuthor, a knowledge-based authoring tool for Intelligent Tutoring Systems that operate as virtual reality computer games, and focused on its player modelling capabilities. VR-MultiAuthor models two broad categories of user characteristics: the domain dependent and the domain independent. Domain dependent features mainly concern the particular domain being taught. On the other hand, domain independent features concern the player's behaviour in the game irrespective of the content of questions being asked to him or her. We argued that a computer game, which aims to be used for educational purposes, has to have the ability to distinguish between a player's ability to play the game itself and a player's level of knowledge in the particular domain being taught. Players who are not familiar with the user interfaces of games should be given extra help in this respect and their level of domain knowledge should not be underestimated. Moreover, we discussed an issue that we consider important: the construction of an integrated learning environment where several game applications could co-exist and pass domain independent information about each player to each other.

## References


[1] Kirkman, C.: Computer experience and attitudes of 12-year-old students: implications for the U.K. national curriculum. *Journal of Computer Assisted Learning*, 9, (1993) 51-62.

[2] Cunningham, H.: Gender and computer games. *Media Education Journal*, 17, (1994) 13-14.

[3] Downes, T.: Playing with computer technologies in the home. *Education and Information Technologies*, 4, (1999) 65-79.

[4] Mumtaz, S.: Children's enjoyment and perception of computer use in the home and the school. *Computers and Education*, 36 *(*2001) 347-362.

[5] Cumming G. & McDougall A.: Mainstreaming AIED into Education? *International Journal of Artificial Intelligence in Education*, Vol. 11, (2000), 197-207

[6] El-Sheikh E. & Sticklen, J.: Generating Intelligent Tutoring Systems from Reusable Components and Knowledge-Based Systems. *In S.A. Cerri, G. Gouarderes and F. Paraguacu (Eds.*): ITS 2002, LNCS 2363, (2002), 199-207.

[7] Murray, T.: Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*. 10 (1999) 98-129

[8] Virvou, M., Manos, C., Katsionis, G. & Tourtoglou, K.: VR-ENGAGE: A Virtual Reality Educational Game that Incorporates *Intelligence In IEEE International Conference on Advanced Learning Technologies* (2002), Kazan, Russia, September 16-19, 2002

[9] Rich, E.: User Modelling via Stereotypes. *Cognitive Science*. 3 (4) (1979) 329-354

[10] Rich, E.: Users as Individuals: Individualizing User Models. *International Journal of Man-Machine Studies*. 18 (1983) 199-214

[11] Id Software (1993), Doom - Virtual Reality Computer Game, Id Software Company Texas.

[12] Walker, J.H., Sproull, L. & Subramani, R. (1994) Using a Human Face in an Interface. In B. Adelson, S. Dumais & J. Olson (eds.) *Proceedings of ACM Conference on Human Factors in Computing Systems* CHI'94, pp. 85-91.

[13] Hollnagel, E. (1991). The Phenotype of Erroneous Actions: Implications for HCI Design. *Human-ComputerInteraction and Complex Systems*, G. R. S. Weir and J.L. Alty (eds.), London Academic Press Ltd.

[14] Hollnagel, E. (1993). The Phenotype of Erroneous Actions. *International Journal of Man-Machine Studies*, 39, 1-32.

[15] Mitrovic, A., Djordjevic-Kajan, S., Stoimenov, L. (1996). INSTRUCT: Modeling students by asking questions. *User Modeling and User Adapted Interaction*, 6 (4), 273-302