

# Web Services for an Intelligent Tutoring System that Operates as a Virtual Reality Game\*

**Maria Virvou**

Department of Informatics  
University of Piraeus  
Piraeus 18534, Greece  
mvirvou@unipi.gr

**George Katsionis**

Department of Informatics  
University of Piraeus  
Piraeus 18534, Greece  
gkatsion@singular.gr

**Abstract** - *This paper describes the conversion of an existing standalone, multi-agent, Intelligent Tutoring System (ITS) to one that can operate through the Internet. The particular ITS works as a Virtual Reality game and is meant to help students learn by providing them a more motivating environment and tutoring help. The system has three agents that have explicit roles. The companion that gives friendly advice, the advisor that provides help and the guard that asks the questions. In order to achieve the goal of the conversion we used a new technology, that of the Web services which has proved very satisfactory for the purposes of our research. We describe the conversion of our standalone multi-agent application to a web-based one, and we discuss the benefits of web services for the purposes of educational software.*

**Keywords:** Intelligent tutoring systems, Web services, multi agent systems, education technologies, virtual reality games.

## 1 Introduction

In recent months, the technology of “Web services” (also referred to as services-based computing or Net services) has attracted a lot of attention within the computer research and industrial community. Although “Web services” marks the entry of yet another term into the Information Technology terminology, the concept of Web services is probably the next step in the evolution of application integration.

Web services can be defined as modular programs, generally independent and self-describing, that can be discovered and invoked across the Internet or an institution intranet [21]. A Web service is a collection of functions that operate as a single entity and are published to the network for use by other programs. Web services are building blocks for creating open distributed systems [13]. There is another, somewhat broader, view of Web Services in which anything available over the network, with some associated description of its functional aspects, is considered a Web Service.

Web services provide an XML layer over the now ubiquitous Internet protocols. In simple terms, Web services technology converts the Internet into an application integration bus. It is now possible for any two applications to plug into this bus without worrying about what is in the middle, because what is in the middle is the Internet. From a pure technology standpoint, Web services represent a shift toward the broad-based adoption of standard interfaces, like SOAP (Simple Object Access Protocol) [5], UDDI (Universal Description, Discovery and Integration) [22], and WSDL (Web Service Description Language) [6], which are used to access application functionality over the Internet.

More precisely, Web services are basically designed to allow loose coupling between client and server, and they do not require clients to use a specific platform or language. In other words Web services are language neutral. Mainly for these reasons among others, these services are becoming very popular. Today, the majority of the software companies are implementing tools based on these new standards [12], [16]. Their use in many different kinds of applications is increasing rapidly.

One important application area that may benefit from the advantages of this new technology is the field of education, where there may exist very demanding applications such as Intelligent Tutoring Systems (ITSs) that have to be transferred over the Web. The benefits of Web for educational software in general are undisputable since it renders tutoring systems accessible by anyone, at any time, from any place. However, in particular for the case of ITSs there is a need for extra effort so that the transfer may be successful. The reason for this is that ITSs aim at individualising tutoring as much as possible through their student modelling component, which typically keeps a detailed record of student characteristics and behaviour. Moreover, ITSs need a very detailed representation of the domain to be taught to support operations such as student error diagnosis.

It is obvious then, that due to the complexity and the demands of these systems (ITSs) the adoption of web services will have many advantages. An intelligent tutoring system may have to be updated frequently to reflect changes on the educational syllabus. Web services can separate a big static application to accessible software components that are self-dependent. So a change to anyone of those will not affect the others. This can simplify the work that has to be done and make it more generic. Also the use of different such parts of an application makes it possible for different applications to use common parts.

The shift from standalone to networked PC computing offers the capability of modelling a large population of individuals. Information about the learner is no longer stored locally on each learner's computer but in a central repository that can be accessed by any client of the application that requests it. This is usually implemented by using client-server architectures. Examples of intelligent learning environments that are based on client-server architectures are WITS [18] and ILESA [15]. However, web services provide a step forward for these architectures.

In view of the above, this paper describes and discusses the conversion of a standalone ITS to one that can operate through the Internet. To maximize the students' engagement and motivation the ITS operates as a virtual reality game similar to the commercial virtual reality games. This version of the game is the latest version of the standalone VR-ENGAGE [24], which operates through the web by using web services. The VR-game renders the ITS more likeable and competitive. As Collins and his colleagues [7] point out, as people take more control of their learning they will choose environments that actively engage them in interesting activities.

## **2 Transferring a standalone ITS-game to the Internet**

Many researchers and educators agree that one-to-one tutoring is a very effective way of teaching. To achieve this, the technology of ITSs is very appropriate since ITSs aim at providing individualised assistance, guidance, error diagnosis and navigation support. This means that beside the human teacher of the classroom every student may have a personal instructor too.

What we have created is an ITS for teaching English orthography and grammatical rules, which operates as a virtual reality game. This ITS-game is called VIRGE (Virtual Reality Game for English). In particular, VIRGE integrates the VR-game with the reasoning of an ITS. Students have the opportunity to play a 3D game, similar

to the commercial ones, which enables them to learn while playing.

VIRGE takes into account the history of answers of students and constructs a student-model for each one of them. VIRGE also monitors closely the actions of students, and it updates the individual student model for every student-player while they play the game. Additionally it generates individualised instruction and advice for students based on their student models. The student characteristics that are being modelled concern the knowledge level of students (answers' results - errors) as well as their behaviour while learning (user actions/ characteristics), which can be connected to their emotions.

VIRGE invites the culture of computer games for creating a language tutoring system that can be very engaging, motivating and cross-cultural. Indeed, many researchers agree that children and adolescents spend most of their time with computers playing games because they are fascinated by them (e.g. [17], [11], and [10]). Many researchers as well point out the potential benefits of education from computer games (e.g. [20], [14], [2]). In the case of language tutoring systems the use of computer games may additionally provide a cultural internationalisation and wide acceptance of these systems.

The environment of a game plays a very important role for its popularity. Griffiths [9] found that the machine's "aura" typified by characteristics such as music, lights, colours and noise was perceived as one of the machine's most exciting features for a large part of the population questioned.

The environment of VIRGE is similar to that of most popular virtual reality games, 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. The main similarity of this ITS with computer games lies in their use of a 3D-engine. However, VIRGE unlike commercial computer games of this kind is not violent at all and is connected to an educational application. In VIRGE 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 English spelling. These virtual worlds look like the one in Figure 1.

However, an application like VIRGE has many resource demands. The 3D environment in which the students can play and learn renders the application very attractive but at the same time it requires a lot of memory and CPU usage. This is certainly expected if one takes into consideration the requirements of most of the commercial games.

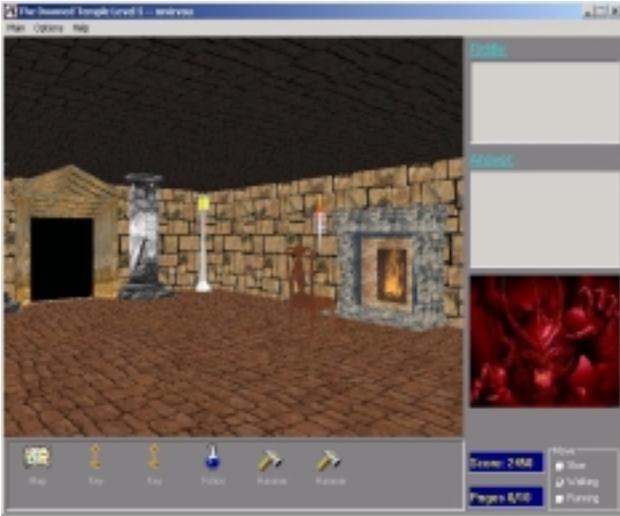


Figure 1: A screenshot of the game.

VIRGE was a standalone application but we wanted to make it accessible to all students. That meant that there should be a client-server program that would allow the students to interact with the application. Each student would play through the client on his or her system, and all of the information results would be stored in the server, located at the teacher's computer. In this way the teacher would always have access to all the information about the students' answers and their user-models.

In the past years there were some approaches to make client-server applications work through the Internet. For example, a method used for the deployment of an ITS over the web [1] was to take all the parts of the client program and provide them through the Internet. In that way, the client-server program worked by having the client as web pages on the Internet and the server located at the Web server where the web pages existed. The server was receiving information from the web pages.

For VIRGE this solution was out of the question. The virtual game was too heavy to work through web pages and would take over 15 minutes just to load each of the web pages. So what was best was to have the client installed at each student's computer, and by some way to communicate with the server at the teacher's computer through the Internet. For this goal we used the new trend in web communications, the Web services, which turned very successful and could be used by other ITSs as well.

### 3 Multi - Agent System

When users interact with a computer, they provide a great deal of information about themselves. This information can be collected and stored in a user model in order to be further processed and used to adapt the system's interaction to each individual user. Agents have been quite successful at observing users' behaviour and,

therefore, they have been widely used in learning environments in order to capture the users' characteristics and perform user-modelling tasks [19], [4] and [23]. Software agents play an important role in human-computer interaction and in the coordination of the internal processes of the system [3].

In the case of VIRGE, animated agents always ask questions to students. Then, students may type their answer in a dialog box. 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. [25]). There is also one other agent, which is not visible to the user and is responsible for updating the student-model, the student profile agent (SPR agent). The architecture of the multi agent part of the system is illustrated in Figure 2.

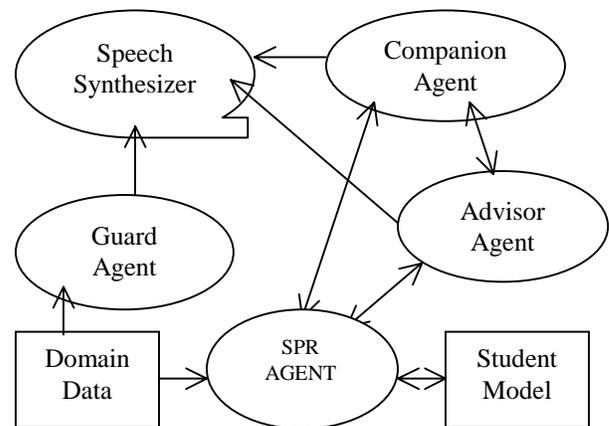


Figure 2: Multi agent system.

The animated agent who is acting as a virtual enemy is responsible for asking questions to students. These questions constitute tests relating to the factual knowledge of the domain and have been constructed during the authoring process. All tests are part of the story of the virtual reality game. Tests may consist of questions of the following types:

1. Spelling questions.
2. Plural form of nouns.
3. Comparative and superlative form of adjectives.

The virtual enemy is a dragon guard outside every door in the virtual worlds of the game, and uses a speech synthesiser. This animated dragon agent asks a question and the student thinks and answers it.

The SPR agent manipulates the results of the answer and any other user actions/ characteristics that might have had place before or after giving the answer. The SPR agent updates the user model and provides the advisor and the companion agent with the equivalent information.

The virtual advisor agent, which has the form of a female angel, appears in situations where the student has to read new parts of the theory or has to repeat parts that s/he appears not to know well. In addition, virtual advisors are responsible for showing empathy to the students and help them in managing their emotions while playing and answering questions. The game itself may motivate students but it may also cause disappointment and frustration each time a student does not perform as well as s/he would like or expect. Moreover, the testing process where students have to answer questions may cause them anxiety (as exams always do) and thus they may perform worse than they could, if they let their anxiety take over them. You can see the virtual advisor agent illustrated in Figure 3.

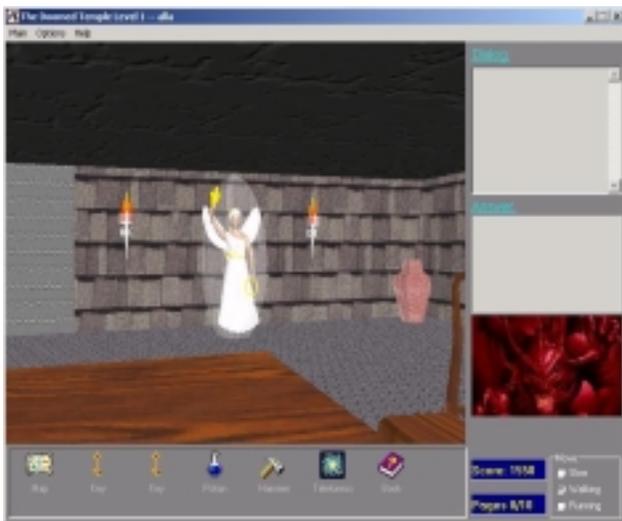


Figure 3: The virtual advisor agent.

The virtual companion, who has the form of an elf, appears in cases where the student has given an answer, either correct or wrong, and the student has declined much from his usual actions or has made a repeated mistake. Then the virtual companion appears, asks the student about this declination and makes some notes. The existence of the virtual companion has been considered quite important by many researchers for the purpose of improving the educational benefit of tutoring systems and promoting the student's sense of collaboration. In Figure 4 you can see a screenshot of the game where a window with the virtual companion (elf) agent has opened to give advice to the student.

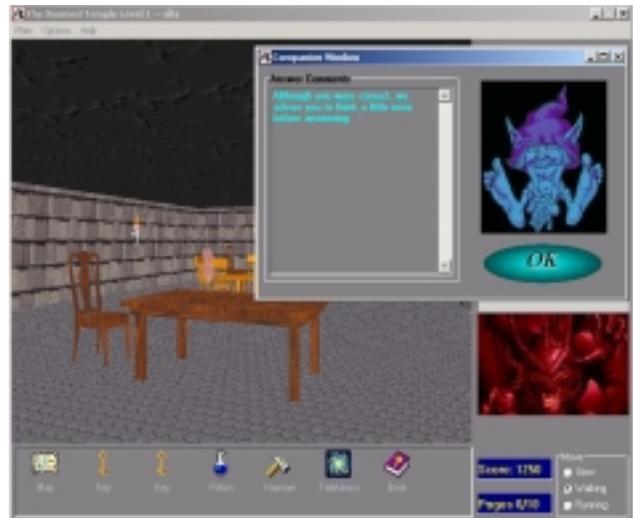


Figure 4: The virtual companion agent.

## 4 The Architecture of the system

Web services are interfaces that describe a collection of operations that are network-accessible through standardized XML messaging. Additionally Web services perform a specific task or a set of tasks. A Web service is described using a standard, formal XML notation, called its *service description*, which provides all of the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location [8].

The basic function of Web services is that they allow a program to pass anything through the Internet by using interfaces. The client and the server may even be written in different programming languages. What is needed is to have some services to be mentioned in the interface part of a Web server (located on the server), and be implemented on the server. So a client calls a Web service form the Web server, which automatically calls that service from the server.

In a similar way we have developed a client, which calls some web services from the web server for specific tasks. One such task is the addition of a student's answer to the database, which is located at the server. Along with his answer there are stored its' characteristics (was it correct, what kind of mistake), along with user actions (haste, second thoughts, etc). Other tasks are any questions regarding the information that exists in the database that keeps the students model. For example the companion agent frequently asks about the category of knowledge being taught, or the specific rule of this category, that the student makes most mistakes, as to provide help for the student.

The client is aware of the operations that are implemented on the server. This means that there are some common units that include information about the interfaces of those operations and are compiled both for the server and the client. When the client wants to perform a task, which requires communication with the server, it calls a service of those interfaces from the web server through the Internet. Then the web server executes this operation, which is implemented on the server. The server can be another standalone application (.exe) or even a dynamic library (.dll) as in our example.

The resulting architecture of the system is illustrated in figure 5.

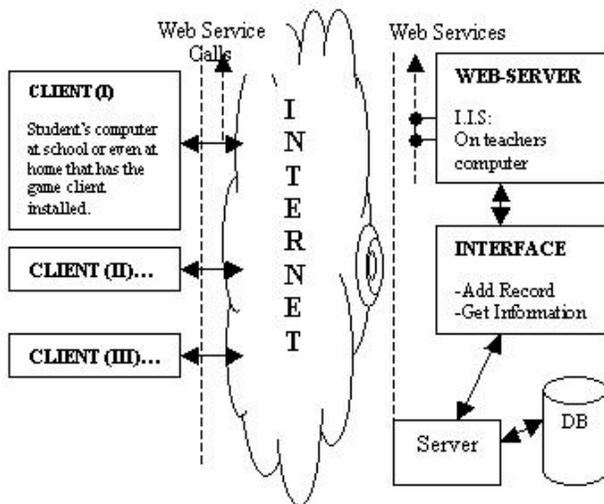


Figure 5. The architecture.

An example of the operation of the system is the following. Every client, located in each student's computer can add an answer of that student to the database located at the teacher's computer. So the client calls a Web service using an interface, through the Internet, from the Web server (Internet Information Server in our example (IIS)) located at the server. Every call provides the Web service with some variables, which contain the information about the student's answer. The Web server (IIS) then calls the service from the server (.DLL in our example) that implements this interface. Then the server executes the service and stores the answer to the database. In this way, from a user's point of view the game works in the same way as the standalone version.

Until now the results we are getting from our experiments show that the transfer to a web-based ITS will help us a lot. We have not converted the whole program to work by using web-services. It is certain that this is going to take us some time. Most of its' operations are executed directly from the client. We have implemented though some parts of it that are general for all the students and should be located at the server on the teacher's computer.

For example the database that keeps the user models and it is updated instantly from every client. The main usability difference that we have gained is that through web services we are managing to make this demanding application to work through the Internet. That was our main purpose.

## 5 Conclusions

This paper has shown how a standalone, multi-agent ITS that operates as a Virtual Reality game has been transferred over the web using the technology of Web services. The advantage of using Web services is that it may render the application usable in the same way as the standalone version despite its very high demands on computer resources. As a result, all students through the Internet may access the ITS-game and their human tutors may have access to these students' progress reports and user models without any problem.

## References

- [1] Alpert S. R., Singley M.K. & Fairweather P.G. (1999) "Deploying Intelligent Tutors on the Web: An Architecture and an Example", *International Journal of Artificial Intelligence in Education*, 10, 183-197
- [2] Amory A., Naicker, K., Vincent, J. & Claudia, A. "Computer Games as a Learning Resource". *Proceedings of ED-MEDIA, ED-TELECOM 1998. World Conference on Educational Multimedia and Educational Telecommunications*, Vol.1, 1998, pp. 50-55.
- [3] Aroyo L. and P. Kommers, "Preface-Intelligent Agents for Educational Computer-Aided Systems", *Journal of Interactive Learning Research, Special Issue on Intelligent Agents for Educational Computer-Aided Systems*, 1999 Vol. 10, No 3/4, pp. 235-242.
- [4] Belkada S., A.I. Cristea & T. Okamoto, "Measuring Knowledge Transfer Skills by Using Constrained-Student Modeler Autonomous Agent", *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001)*, IEEE Computer Society, 2001, pp. 375-378.
- [5] Box D., D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, D. Winer, "Simple Object Access Protocol (SOAP) 1.1", *W3C Note*, May 2000, <http://www.w3.org/TR/SOAP>.
- [6] Christensen E., F. Curbera, G. Meredith, S. Weerawarana (eds.) "Web Services Description Language (WSDL) 1.1", *W3C Note*, March 2001, <http://www.w3.org/2001/NOTE-wsdl-20010315>.

- [7] Collins A., Neville P. & Bieloiczyc K. (2000), "The Role of Different Media in Designing Learning Enviroments", *International Journal of Artificial Intelligence in Education*, 11, 144-162.
- [8] Gottschalk K., Graham S., Kreger H. & Snell J. "Introduction to Web services architecture". *IBM SYSTEMS JOURNAL*, VOL 41, NO 2, 2002
- [9] Griffiths M.D. "Technological addictions". *Clinical Psychology Forum* , 76, 1995, pp. 14-19.
- [10] Griffiths M.D., & Hunt, N. (1995): "Computer game playing in adolescence- prevalence and demographic indicators". In *Journal of Community and Applied Social Psychology*, 5(3), pp. 189-193.
- [11] Harris S. (1999): "Secondary school students' use of computers at home" In *British Journal of Educational Technology*, 30(4), pp. 331-339.
- [12] IBM White Paper (2000) "The IBM WebSphere software platform and patterns for e-business invaluable tools for IT architects of the new economy".
- [13] IBM: "The Web services (r)evolution, Part 1", <http://www106.ibm.com/developerworks/webservices/library/ws-peer1.html>
- [14] Inkpen K., Upitis, R., Klawe, M., Lawry, J., Anderson, A., Mutindi, N., Sedighian, K., Leroux, S. & Hsu, D. "We Have Never-Forgetful Flowers In Our Garden: Girl's Responses to Electronic Games". *Journal of Computers in Math and Science Teaching*, 13(4), 1994, pp. 383-403.
- [15] Lopez J.M., E. Millan, J.L. Perez-de-la-Cruz, F. Triguero, "ILESA: a Web-based Intelligent Learning Environment for the Simplex Algorithm", in Alvegard C. (ed.): *Proceedings of CALISCE'98, 4th International Conference on Computer Aided Learning and Instruction in Science and Engineering*, 1998, pp. 399-406.
- [16] Microsoft White Paper (2000) "A Blueprint for Building Web Sites Using the Microsoft Windows DNA Platform".
- [17] Mumtaz S. (2001): "Children's enjoyment and perception of computer use in the home and the school" In *Computers & Education*, 36, pp. 347-362.
- [18] Okazaki Y., K. Watanabe, H. Kondo, "An Implementation of an intelligent tutoring system on the World-Wide Web", *Educational Technology Research*, 1996 Vol. 19, No. 1, pp. 35-44.
- [19] O'Riordan and J. Griffith, "A Multi-Agent System for Intelligent Online Education", *Journal of Interactive Learning Research*, Special Issue on Intelligent Agents for Educational Computer-Aided Systems, 1999, Vol. 10, No 3/4, pp. 263-274.
- [20] Papert S. "The Children's Machine: Rethinking School in the Age of the Computers". Basic Books, New York, 1993.
- [21] Pires P.F., Benevides M. & Mattoso M. (2002) "Building Reliable Web Services Compositions", *Net.Object Days - WS-RSD'02*, 551-562.
- [22] Universal Description, Discovery and Integration (UDDI) Version 2.0 Specification, June 2001, "<http://uddi.org/specification.html>".
- [23] Virvou M. & Kabassi, K. (2002). A Multi-Agent System for Intelligent Assistance in a GUI. In *Proceedings of the 3rd International NAISO Symposium on Engineering of Intelligent Systems*, NAISO Academic Press, Canada/The Netherlands.
- [24] Virvou M., Manos, C., Katsionis, G. & Tourtoglou, K. (2002) "VR-ENGAGE: A Virtual Reality Educational Game that Incorporates Intelligence". In *IEEE International Conference on Advanced Learning Technologies (ICALT' 2002)*, Kazan, Russia.
- [25] 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.