A 3-TIER DYNAMIC ADAPTIVE EDUCATIONAL ENVIRONMENT

by
Athanasios Hatzigaidas, Anastasia Papastergiou, George Tryfon, Despoina Maritsa

Abstract: The aim of this paper is to present the design, implementation and functionality description of a Web-based adaptive educational environment which employs a 3-tier architecture. The use of a dynamic 3-rd tier database is implemented to enhance the interactiveness of the educational environment in order to provide a user-adapted presentation of learning material. Equally important an alternative method for the client/server calls through http protocol is going to be explored in order to improve client/server communication and access to database. The proposed environment is going to be tested using the topic of microcontrollers as a context.

Keywords: 3-tier architecture, adaptive educational environment, dynamic database, adaptivity, web-based

INTRODUCTION

Designing and developing Web-based adaptive learning environments (ALE) and by extension Intelligent Tutoring systems (ITS) is a demanding field of research progressing to implement integrated technologies into educational systems (Brusilovski, P., 1999). Web based learning environments offer platform and classroom independence, giving the opportunity to a large number of learners to take advantage of the new technologies.

Typically web-based educational systems are based on a 2-tier client/server architecture. In order to implement an ALE and in extension an ITS there is the need for a reusable database which will be able to dynamically assembled as the user's situation evolved (Alpert, S. R et al., 1999; Martinez, M. et al., 2001). A 3-tiered architecture system can implement the increased needs of a web-based adaptive learning environment and overcome the disadvantages and limitations of 2-tiered architecture (Anil N., 2000 ; Zinn, et al.,2002).

The aim of this work is to present the architecture and describe the functionality of a Web-based adaptive educational environment which employs a 3-tier architecture.A new protocol is invented to facilitate the client/server calls through http protocol in order to improve the client/server communication and access to database. The use of a dynamic 3-rd tier database is implemented, to enhance interactiveness of the educational environment, providing a user-adapted presentation of the learning material.

The feasibility and efficiency of the proposed educational learning environment is going to be explored in the trial implementation using the subject of microcontrollers as a context in the Microcontollers Laboratory, in the Department of

Electronics. The project can be accessed by staff and students through a standard web browser without user's need for extra software or plug ins. This way the student may have an online dimension that certainly extends learning options and removes time and place restrictions on study.

DESCRIPTION

This paper describes theoretical and technical aspects concerning the design and implementation process of a 3-tier web-based adaptive educational system. The basic architecture of the proposed system was implemented and applied to the course material of Microcontrollers for the students of the Department of Electronics.

Most web sites employ a 2-tier client/server architecture which can implement a form of Computer Aided Instruction, but it falls short of an ALE or ITS (Anil N.,2000 ; Zinn, et al.,2002).

Minimum requirements of an ITS:

- transfer of a vast amount of data and contents.
- repository of permanently-stored reusable 3rd-tier database learning components which can be dynamically assembled according to established protocols by the learner, the educator, or the learning system
- the adaptive behavior conforms to the individual and changes as the learners' situation evolves.

The architecture of 3-tier adapted educational system is constituted basically from: 1) Web browser 2) Web application server and 3) a backend database. (Figure 1)

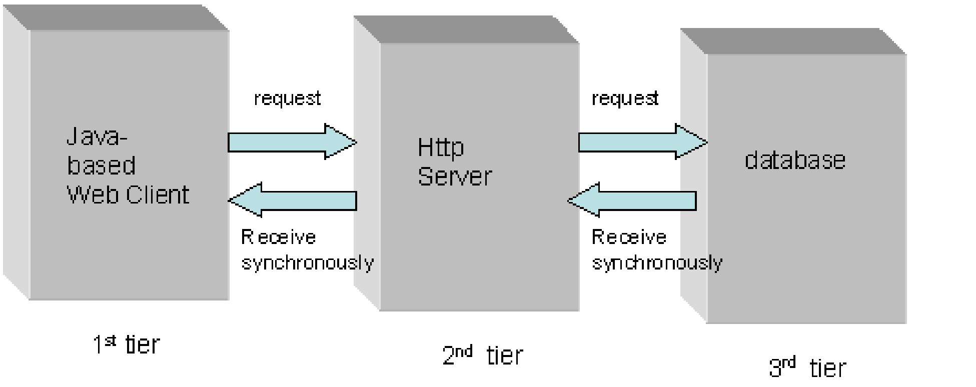


Figure 1:3-Tier Architecture for an Intelligent Tutorng System

The three tier architecture is used when an effective distributed client/server design is needed that provides (compared to the two tier) increased *performance* , flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user (Korson D., 2001 ; Trella et al., 2002).

The main advantage of 3-tier architecture is:

- Scalability because extra resources can be added to any tier to improve performance.
- Easy deployment in development
- Clear separation of user-interface-control and data presentation from application-logic. Through this separation more clients are able to have access to a wide variety of server applications
- Re-definition of the data won't influence the clients
- Less software to the client, better security

The proposed adaptive educational system implies a 3-tier architecture having the following structure (Figure 2):

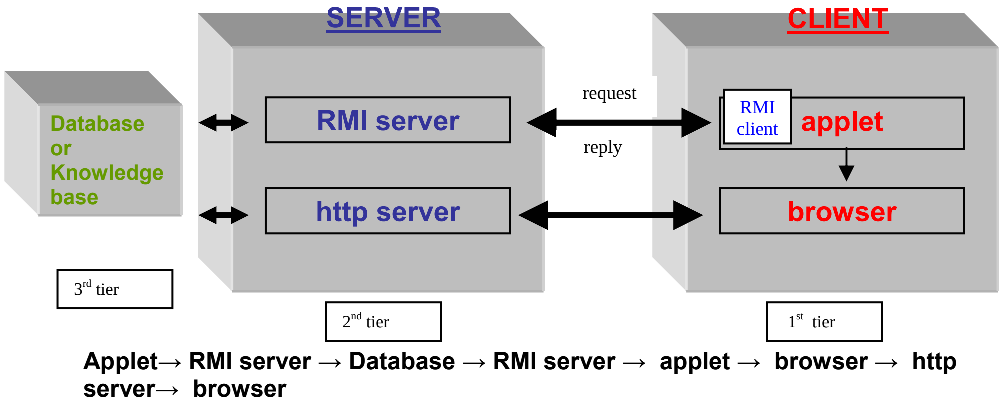

SERVER
CLIENT
Database or Knowledge base
RMI server
http server
request
reply
RMI client
applet
browser
3rd tier
2nd tier
1st tier
Applet→ RMI server → Database → RMI server → applet → browser → http server→ browser

Figure 2: The overall structure of the system using RMI communication protocol

The system is Java-based and it is implemented via a java applet in an internet browser. Java technology provides a reliable solution for highly interactive Web-based adaptive intelligent tutoring systems (Brusilovski, P., 1999). In the client side an applet is working in a Java enabled standard browser (**Applet**: Java program incorporated in the html page, which is transferred with the page in the PC where is executed). Web client via user interface communicates with the server.

The management and maintenance of the system relies on the server side (Alpert et al., 1999, Korson D., 2001). Software modifications and updates takes place in the server side without the need of any redistribution. The management of the database becomes in the server, without any client relation.

The database components are written in XML (eXtended Markup Language), based on its facility to exchange complex document on the Internet (Brusilovsky et al., 1996; Gaul et al., 2002).The approach of xml registrations in the database aiming to the following :

1. The supervision of the registrations becomes easier as they are text type
2. Easy modification of the registrations with a simple text processing program
3. Flexible management of registrations of database.

Furthermore, it must be noticed that the use of XML to provide adaptable courses on the Internet has several advantages. Among these, it would be mentioned its simplicity, the small data transfer between the learner and the course server –because only profile related data are transferred- as well as the negligible processing to adapt a course to each learner.

The database contains students characteristics and history, pedagogical model and learning components (tutorials and tests) stored in xml files which can be accessed, and matched efficiently. The components of the database must be able to dynamically assembled and adapted as the student's situation evolves.

Java supports numerous client-server communications ie - HTTP/CGI, sockets, RMI, CORBA (Brusilovski, P., 1999). In the early stage client/server communication accomplished via RMI (remote method invocation) protocol. The RMI client part is implemented in the applet, and the RMI server part is installed in the server. RMI protocol, based on the TCP/IP, does occupy predetermined port, through which works, but it gets a parameter each time, depending on the case.

The system had the forecasted output, but under the condition that there was an open port available from the existing system of protection (firewall), for communication via RMI stream both on server and on client side. This particularity had the disadvantage that in these cases, required prior and advanced computer architecture knowledge from the user (open port).

To overcome this problem, a new client/server protocol was designed and implemented. The new protocol (let's name it 80MI), is supported in http and thus the calls and answers pass from port 80. Figure 3 shows the relation between the protocols.

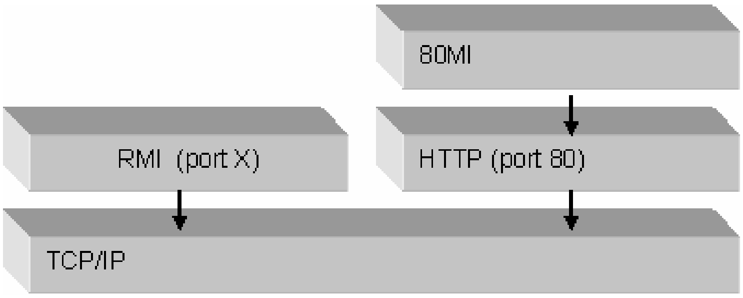
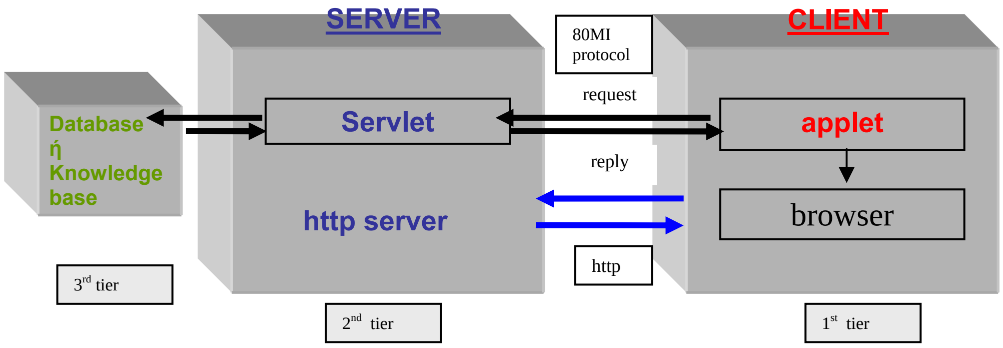


Figure 3: Representation of RMI and 80MI protocols

Consequently, the proposed system with the use of the new protocol for client/server communication is shaped as shown Figure 4.



Applet→ Servlet → Database → Servlet → applet → browser → http server→ browser

Figure 4 The overall structure of the system using 80MI communication protocol

The server part of the new protocol 80MI constitutes part of a new library and was installed in the Servlet replacing RMI. **Servlet** is a Java program that is installed in the server and answers in http calls.

The main problems in the design and implementation of the new communication protocol 80MI were:

- The transport of description of call and parameters from Applet to Servlet and the transport of results of implementation back to Applet.
- Servlet should recognize dynamically which call the Applet demands and execute this call by passing the necessary parameters.

INNOVATION

To overcome the above problems, a class named Walker has been build. This class transports objects from the Applet to the Servlet and backwards, as well as the name of the call which the client requires to take place on the server. Objects of the Walker type have the ability to be serialized, and pass through channels transferring all the contents.

Two parts of the protocol were developed and were placed respectively to the client and server part of the application manipulating the Walker Objects.

The client part accepts the user request, constructs a Walker Object and serializes it to the server. The server part from the other side accepts the serialize Walker, unserializes it and derives from it the request call and the parameter objects. Then executes the call, get the results from it and constructs a Walker object which is then serializes and pass back to the client. Finally it is the client part of protocol, to unserialize the returned Walker Object and extract the results of the remote call.

In order the server to be flexible and execute every requested call, the calls are not hardcoded in the protocol, but a **Class loader** is used with the ability to locate and execute functions based on their name and their parameter types.

The advantages of the proposed 80MI client/server protocol for the achievement of client/server communication are summarized as follows:

- All the required functionality for the client/server communication with no use of heavy libraries that require installation or slowing down the application.
- Simplicity as there is no the need to maintain static lists with processes
- Flexibility in software development and software maintenance
- No need for advanced software knowledge from the user/student in order to overcome the problems that occurred by the use of RMI protocol

FUNCTIONALITY

In this section three modes of operation are described. The procedures that the system executes are quite a few, such as:

1. Login of the user/student into the system.
2. Choice of the tutorial that the student will attend
3. Procedure of test construction.

When the student visits the applications page, a part of it (application) is transferred into the students PC. This is the "client" part of the application. The application still does not know who (student) called the application, and displays a button that the student should press in order to start the recognition process to (login). By click on the "log-in" button (Figure 5), a dialog box requests from the student the user-name and password. The Applet itself can not confirm for the validity of the data and sends them to the servlet for confirmation.

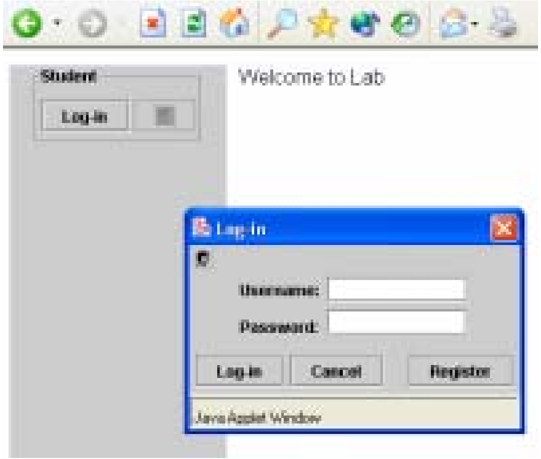


Figure 5: Snapshot of the overall system

As it has been described, the Applet constructs a Walker Object, passes the name of the call to be executed (into the server), “login” for this case and also passes the data (username, password). Then forwards the Walker object to the Servlet. The Applet will wait for the reply before being able to continue his work. For security reasons there is a time-out period that releases the Applet (waiting period). If no reply is transmitted from the other side a relative error message is displayed.

Let’s follow the Walker object as it travels through the wire.

The Walker object is reconstructed inside the Servlet and the Servlet extracts the name for the call and the parameters and uses a “Class loader” to fire the call. Finally takes the results from the call, constructs a new Walker object and sends it back to the Applet.

APPLICATIONS

If there is an interest let us examine more closely what happened with this specific call “login”, which is part of a library that manages students. The “login” checks if there is a registration with this particular “username” and “password”. It will return true if the above are true, otherwise will return false. In the case of “true”, in the history file for this particular student a new entry with the date and the time of the log-in will be added. After sending the reply back to the Applet, the Servlet remains idle waiting calls from the same or different Applet with the next request. If the reply from the Servlet is positive, the Applet will allow for the student to proceed, otherwise will display an error message and prompt him to retry.

After a successful login, the student has the right to requests several functions. A classical one is the request to follow a particular tutorial. Let’s follow this process

Inside the Applet there is a tree-like structure where all the tutorials are visible. The selection is just a click on the name of the wanted tutorial.

Let us say that a tutorial corresponds to some html pages on the server.

By clicking the tutorials name on the Applet, the Applet tries to find the specific URL (Universal Resource Location) path to the html page which is the wanted tutorial, and calls the Servlet with this question. The Servlet searches in the database and extracts the answer and sends it back to the Applet. The Applet gets the path and forces the browser to load the specific path. The browser will load the page normally.

Finally, it is interesting to follow the test construction. Accepting that for each tutorial the student follows, there is a relative test. By right clicking on the tutorials name in the above mentioned tree, a pop up menu appears from which the student may select the “test” option. Each test is dynamically constructed and it is not the same even if it is for the same tutorial and for the same student.

Our approach is the following. The Applet constructs a description of an HTTP call, not to a static page, but to the Servlet itself and inside the call embeds the username and the tutorials id.

The call is of the form:

http://<server name>/path/servlets/labwiz.server.servlet.Serv?action=test;userName=<user name>;tutID=<tutorial ID>

Then the browser is forced to proceed with this HTTP call. The browser proceeds and expects an HTTP page as a result, but it is the Servlet that is called and it is the Servlet that constructs dynamically a test for the specific student, taking various questions from an item pool containing questions following the pedagogical procedures which the tutoring system implies. Finally appends the "submit test" button which should be pressed from the student upon the completion of the test. When the "submit test" is pressed, the Applet is trigered, summarizes students answers, and send the answers to the Servlet. The Servlet scores the student and adapts the profile of the student.

EVALUATION AND CONCLUSION

An approach to implement a web-based adaptive educational environment to be used it in the training field of microcontrollers. Currently, the basic architecture of the proposed system was implemented and applied to the course material of Microcontrollers of Department of Electronics.

A quantitative study was conducted using a sample of 50 students of the Department. Students have access to the proposed system through a standard web. The evaluation was made in three stages:

1. Students were given the opportunity to interact with the system. Specifically they navigate through the system and explored the functionalities given. Firstly explored the system using RMI communication protocol, but most of the them faces difficulties n order to establish the communication with the server.
2. The students navigated through the proposed web-based tutoring system and explored the functionalities given.
3. Finally, the students were interviewed in an informal manner in order to collect their comments and impressions about the system. They were asked to evaluate the system, corresponding to specific questions concerning system's flexibility, functionality and adaptivity to the student.

From the analysis of the students response and from observations concerning the functionality and efficiency of the proposed educational environment we came to the conclusions that

1. the employment of 3-tier architecture in the proposed web-based educational system,
2. the use of the 3rd database containing all the knowledge base in a form of xml-files
3. and the introduction of the new 80MI protocol for the achievement of the client-server communication

contributed to achieve a web-based educational system with increased *performance* , flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user.

Within the future of this work is the employment of artificial intellligent teqniques in order to improve the adaptation of the system to the student's need and to enhance its effectivness.

BIBLIOGRAPHY

Alpert, S. R., Singley, M. K., Fairweather, P. G (1999), Deploying Intelligent Tutors on the Web: An Architecture and an Example. Journal of Artificial Intelligence in Education 1999; 10:183-197.

Anil Nori (2000): Application Architecture: 2Tier or 3Tier? What is DBMS's Role? (Panel Abstract). SIGMOD Conference , 575

Brusilovski, P., (1999), Adaptive and Intelligent Technologies for Web-based Education, In C. Rollinger and C. Peylo (eds) Special Issue on Intelligent Systems and Teleteaching, Kunstliche Intelligenz, Vol. 4, pp 19-25

Brusilovsky, P., Schwarz, E. and Weber, G. (1996), ELM-ART: an Intelligent Tutoring system on World Wide Web, In C. Frasson, G. Gauthier, and A. Lesgold (Eds.) ITS Lecture Notes in Computer Science, Springer Verlag, Berlin, Vol. 1086, pp.261-269.

Gaul, T. , Lowe, W. and Noga (2002). Foundations of Fast Communication via XML. Annals of Software Engineering— Special Volume on Object-Oriented Web-Based Software Engineering, in revision.

Mandl H. & Lesgold A. (1988). Learning Issues for Intelligent Tutoring Systems. Springer-Verlag,London.

Mark, M. A. and Greer, J. E (1993). Evaluation Methodologies for Intelligent Tutoring Systems. Journal of Artificial Intelligence in Education 4(2/3),p. 129-153

Martinez, M., & Bunderson, C. (2001). Building Interactive World Wide Web (Web) Learning Environments to Match and Support Individual Learning Differences. Journal of Interactive Learning Research, 11 (2)

Murali Sitaraman, Timothy J. Long, Bruce W. Weide, E. James Harner, Liqing Wang, (2001), A Formal Approach to Component-Based Software Engineering: Education and Evaluation. ICSE 2001: 601-609

Korson Timothy D., (2001), From UML to Java, Building a 3-Tier Architecture: Case Study. ICSE 2001: 735-736

Polson, Martha C., and Richardson, J.Jeffry (1998) Foundations of Intelligent Tutoring Systems, New Jersey.

Trella, M., Conejo, R., Bueno, D., Guzmán, E.: (2002). An autonomous component architecture to develop WWW-ITS. In Brusilovsky, P., Henze, N., Millán, E. (eds.) Proceedings of the Workshops on Adaptive Systems for Web-Based Education. Malaga

Wenger (1987). Artificial Intelligence and Tutoring Systems. Morgan Kaufmann Publishers, Inc.

Zinn, C., Moore. D.J., and Core G.M.,(2002) A 3-tier Planning Architecture for managing Tutorial Dialogue, 6th Int. Conf. Intelligent Tutoring Systems, S 574-584

Authors:

Athanasios Hatzigaidas, sakis@el.teithe.gr.
Anastasia Papastergiou, natp@el.teithe.gr.
George Tryfon, georgetryfon@freemail.
Despoina Maritsa, gr, maritsa@hotmail.com.

Department of Electronics, Technological Educational Institution of Thessaloniki, Thessaloniki, Greece