Chapter 8

TELE-EDUCATION OF THE INSTRUCTION DYNAMIC SCHEDULING USING A WEB SIMULATOR

Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, Juan M. Sánchez-Pérez, J. Carlos Burgueño-Suárez and José M. Granado-Criado
Dept. Informática, Univ. Extremadura, Escuela Politécnica, Campus Universitario s/n, 10071 Cáceres, Spain, mavega@unex.es, Fax: +34-927-257-202, http://arco.unex.es

Abstract: Communication and information technologies have become a fundamental tool in education, due to their great advantages: any place and moment, saving of costs, interactivity, etc. For these reasons, we believe it is important to dedicate efforts in the development of educational proposals and prototypes via Internet. In this chapter a multimedia simulator is presented, based on Internet, which has been developed and it is applying for the teaching of Instruction Dynamic Scheduling (IDS). At present, IDS is studied in all subject about Computer Architecture because it is a fundamental aspect inside the pipelined processors, and any current computer has a pipelined processor. The platform we present here is named PDIWeb, and it has been developed thanks to a grant for Projects of Educational Innovation at the University of Extremadura (Spain). This chapter presents a general description of this web simulator, as well as the methods and tools used for its implementation. The chapter also includes the results obtained after the platform use and the realization of anonymous surveys by the students. In conclusion, it is an educational innovation that allows improving the teaching in Computer Architecture.

Key words: Internet; Multimedia; Simulator; Computer Uses in Education; Computer Architecture.

1. INTRODUCTION

Internet is already, at present, a very important educational resource, since it allows overcoming the place and time limitations, reducing costs. Also, it should not be forgotten the effect that the interactive software has in the

B. Fernández-Manjón et al. (eds.), Computers and Education: E-learning, From Theory to Practice, 89–98.

learning process. Therefore, we believe very important to dedicate efforts in the development of proposals and teaching prototypes through Internet. In this line, and from 1998, our research group has worked on diverse projects, as EDONET (Sánchez et al., 2000), TEDA (Vega et al., 2002a) or SD2I (Vega et al., 2002b).

At present, we are focused on the project PDIWeb. This project arises from the application of our research to our teaching. The global objective is the development of a system for the teaching through Internet of part of the subject Computer Architecture and Engineering (CAE). CAE is an annual compulsory subject of 90 hours that is given in fourth course of Computer Engineering (see (Vega et al., 2006b) for a more detailed description of CAE), at the University of Extremadura (UEX). In particular, this system is focused on the lessons given during the first quarter and dedicated to instruction dynamic scheduling, fundamental aspect inside the instruction pipelining.

The project PDIWeb began after obtaining a grant for Projects of Educational Innovation, being supported by the Vice-Chancellor's Office of Educational Innovation and New Technologies, and the Institute of Education Sciences (both at the University of Extremadura, Spain).

The rest of the chapter is organized as follows: next section explains the methods and tools we have followed for the implementation of our platform. In section 3 we give an overview of the platform, indicating its fundamental characteristics. Then, in section 4, we detail our experiences in the use of PDIWeb and the opinion of the students. Finally, the conclusions of this work are presented.

2. METHODS AND TOOLS

For the development and maintenance of PDIWeb we have a server that supports the educational system through Internet. The server, under Windows 2003 Server and with Internet Information Server (IIS), offers web page publication and administration services (WWW), file transfer (FTP), Gopher, electronic mail (e-mail), mail distribution lists compatible with *Majordomo*, etc.

For the implementation of PDIWeb we have used, mainly, the programming language PHP 4.1 (McCarty, 2001), (The PHP Group, 2006). PHP 4.1 is executed in web servers and it allows creating HTML pages in a dynamic way. Furthermore, it is a similar language to ASP (Active Server Pages) of Microsoft, but more potent, quick, free (of charge), multiplatform and open to improvements and extensions permanently. Due to this, PHP is prevailing over other programming languages as alternative for the development of computer applications in Internet.

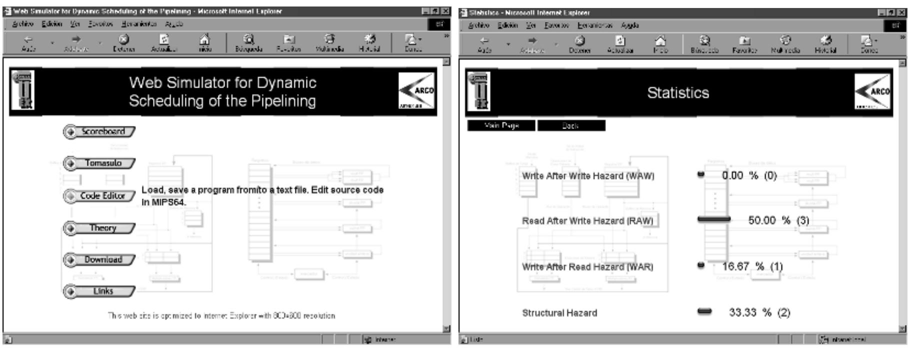
For some specific functions, it has been necessary to include scripts coded in JavaScript (Goodman and Morrison, 2004), a language designed so that the applications support the remote execution. HTML code fragments also exist (Morrison, 2002), since, after all, the simulator is formed by a set of web pages. In spite of this, 80% of the code is implemented in PHP.

The editing of all this code in PHP, JavaScript and HTML has been performed by means of the tool Macromedia Dreamweaver MX (Lowery, 2003), a powerful application for the creation and maintenance of websites. Furthermore, Macromedia Fireworks MX (Cohen, 2004) has also been used for the graphic design of diverse web elements: images, animations, flash texts, etc.

3. PDIWEB CHARACTERISTICS

PDIWeb (Vega et al., 2006a) is a web simulator, based on the architecture of a pipelined MIPS processor with instruction set of 64 bits, which simulates the instruction dynamic scheduling using two different techniques: one centralized by means of the *Scoreboard method*, and another distributed according to the *Tomasulo algorithm* (Hennessy and Patterson, 2003).

The simulator does not require any installation in the PC by the student, since it is executed through Internet in a remote server. Therefore, in order to execute the simulator, a student only needs a PC with access to Internet and a browser that can visualize web pages designed with HTML frames (Morrison, 2002) and multimedia elements of flash type. In short, the simulator has been tested with success in the last versions of Internet Explorer and Netscape.



(a) (b)

Figure 8-1. (a) Main page. (b) Web page with the statistics about hazards after a simulation

Figure 8-1(a) shows the main page of the simulator. This web page contains the main menu of the platform. The student can see a help text moving the mouse over each option. The following subsections describe the PDIWeb interface briefly.

3.1 Scoreboard

Figure 8-2 presents the web page for the Scoreboard method. This web page contains a menu, followed by the options that control the simulation progress: *Partial Cycle*, *Current Cycle*, *Previous* and *Next*. *Partial Cycle* allows the use of breakpoints or complete simulations (without stops). The rest of the page is occupied by the tables used by the Scoreboard method: Instruction Status, Register Result Status, Functional Unit Status, and Stage Status.

By means of the menu options the student can: go back to the main page (option *Main Page*, figure 8-1(a)), start a simulation (option *Go Simulation!!!*), go to the code editor (option *Code Editor*, subsection 3.3) in order to change the program to simulate, print a simulation (option *Print Simulation*, figure 8-3(a)), configure the functional units in the processor (option *F.U. Configuration*, figure 8-3(b)), and show the statistics about hazards (in graphic, percentual and numerical format) for the current simulation (option *Statistics*, figure 8-1(b)).

When the student prints a simulation the report includes: the tables of the Scoreboard with the current simulation results, the current simulation cycle, the hazards, and the configuration of the functional units. Figure 8-3(a) shows an example, with the HTML page generated by the simulator before the student prints it.

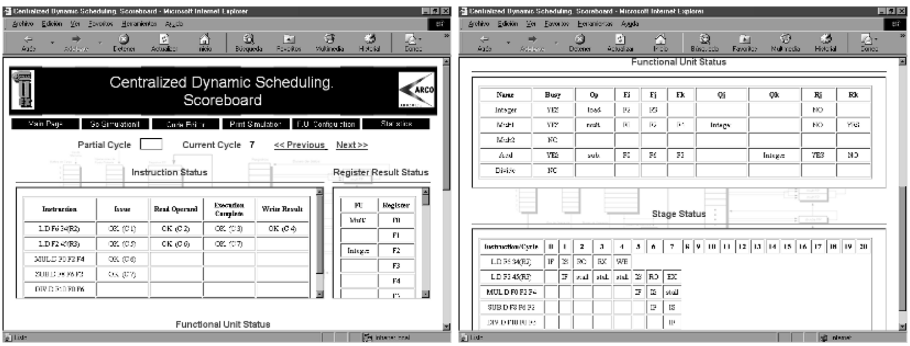
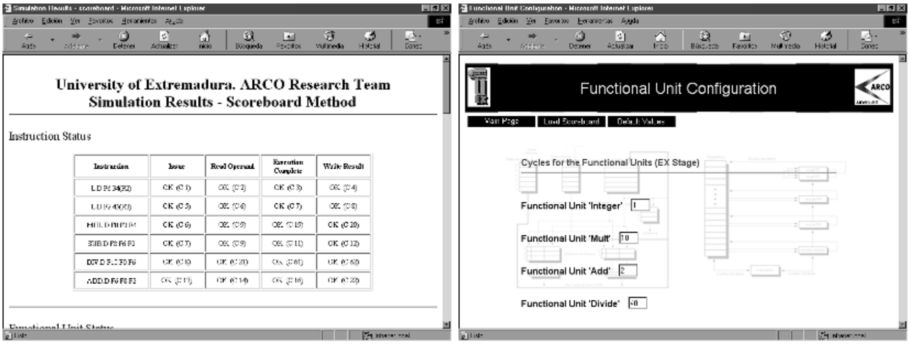


Figure 8-2. Web page for the Scoreboard method: from top to bottom



(a) (b)

Figure 8-3. (a) Printing the results of a simulation. (b) Configuring the functional units

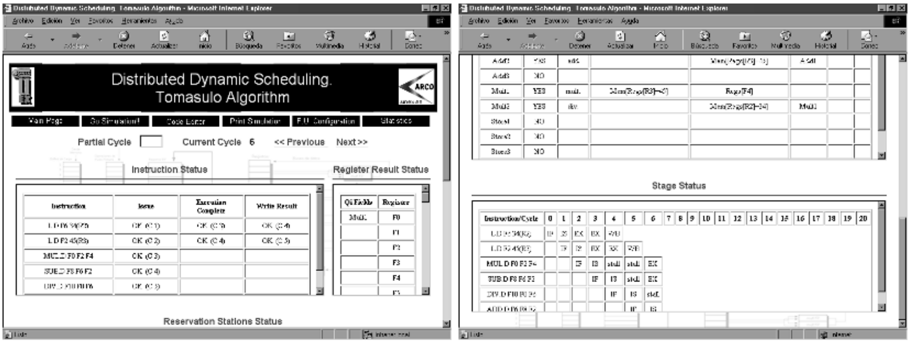


Figure 8-4. Web page for the Tomasulo algorithm: from top to bottom

Figure 8-3(b) displays the web page for configuring the functional units of the processor. The page has a different appearance depending on whether the Scoreboard method or the Tomasulo algorithm is being used. Note that, in this web page, the user can set default values for all the configurations (option *Default Values*).

3.2 Tomasulo

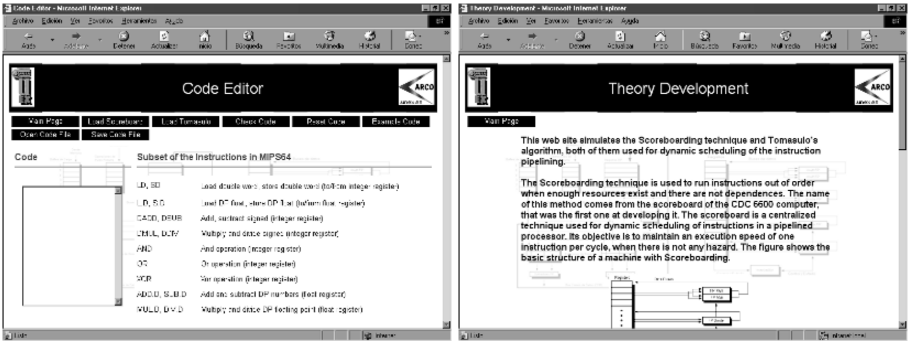
Figure 8-4 shows the web page for the Tomasulo algorithm. As you can observe, the top of the page is occupied by a menu with the possible operations. These options are very similar to those explained for the Scoreboard method in the previous subsection. The options that control the simulation progress are below this menu: *Partial Cycle*, *Current Cycle*, *Previous* and *Next*; whose functionality is identical to the one explained for the Scoreboard method previously.

The rest of the web page is occupied by the tables used by the Tomasulo algorithm: Instruction Status, Register Result Status, Reservation Stations Status, and Stage Status.

3.3 Code Editor

Figure 8-5(a) presents the code editor, where the student can indicate the program to simulate. In the central area, there is a text editing window (where the user will introduce the code, the instructions, of the program to simulate) and information about the instruction subset of MIPS64 that the simulator accepts. At the top, there is a menu with the different editor functions:

- *Main Page*. Link to the main page (figure 8-1(a)) of the web platform.
- *Load Scoreboard*. It loads the code edited in the simulator, performing the simulation by means of the Scoreboard method. The code should be validated before selecting this option, otherwise an error message will appear.
- *Load Tomasulo*. It loads the code edited in the simulator, performing the simulation following the Tomasulo algorithm. Before loading the code, this should be validated.
- *Check Code*. It performs a verification of the code written in the text editing window. At the top of the web page a table will be shown with the results of this validation, indicating the possible errors in each instruction.
- *Reset Code*. It deletes the code that has been written in the text editing window.
- *Example Code*. It displays at the top of the page several examples of programs written in a correct way for the simulator.
- *Open Code File*. It allows the user to open a disk file to load the code stored in it. After selecting this option a window will appear, allowing the user to examine the hard disk (and other devices of massive storage) in order to set the path of the file to open. The file to open should be of text type (*.txt).
- *Save Code File.* It allows the student to save the edited code in a disk file. When choosing this option, the student will be able to examine the hard disk (and other storage devices) to set the path of the file to save. The file to save should be of text type (*.txt).



(a) (b)

Figure 8-5. (a) Code editor. (b) Page with the theory within the web platform

3.4 Other Options

The Theory web page includes an introduction to the theoretical foundations of the Scoreboard method and the Tomasulo algorithm, as well as the instruction dynamic scheduling in pipelined processors (see figure 8-5(b)).

In the Downloads page the student can download didactic material related to the simulator. At present, for example, it is possible to download a local application that simulates the Scoreboard method. This local application is an application for Windows, with an user-friendly and full graphic interface. This application has been developed in Delphi, using object-oriented programming. The page presents a brief description of each application that can be downloaded, explaining its main characteristics, showing images of that application, and including its technical specifications (name of the installation file, size, language, etc.).

Finally, the Links web page includes a list of links to websites that, due to their contents, have some relation with PDIWeb.

4. STUDENTS OPINION

After the system construction (the results can be observed in (Vega et al., 2006a)), the platform has been evaluated by the students. That is, once it has been concluded, we have asked the students to test the developed platform. In this way, the defects and virtues of the system, in comparison to the traditional teaching method, can be detected. This evaluation has been carried out by means of the system use by the students in a massive way, and the realization of surveys to them.

Furthermore, the experience allowed checking the correct system operation, as well as the acceptance degree by the student body. A total of 50 students tested the system and answered to an anonymous survey. All of them attend to the subject CAE. Students had enough knowledge about instruction dynamic scheduling and experience on the conventional teaching of this subject, so they were able to evaluate the PDIWeb platform for comparing it with the traditional classes (professor explanations). We did not give any explanation about the platform use to the students. The student age ranged from 21 to 29 years. Due to space reasons we only detail the most significant results of the survey.

As for the evaluation to the PDIWeb platform contents (*How do you value, in general, the contents of the PDIWeb platform? Very good, Good, Average, Bad, Very bad*), the 95.8% of the students believes that the contents are good or very good. The other 4.2% thinks that the contents are average.

The results about the platform pedagogical utility (*Value the pedagogical utility of the PDIWeb platform. Very good, Good, Average, Deficient, Very Deficient*) also are encouraging. Again, around the 90% of the students (concretely, the 89.6%) thinks that the pedagogical utility is good or very good. It must be observed that, from the pedagogical point of view, none of the students evaluates the platform negatively (the other 10.4% thinks that the pedagogical utility is average).

Regarding the utility degree given by the students to the platform for its use within the subject CAE (*Do you consider that the application of PDIWeb is useful within the subject? Very useful, Useful, Moderately useful, Little useful, Not useful*), again the 90% is exceeded, since the 93.8% considers that the platform is useful or very useful for this subject (the other 6.2% considers that the platform is only moderately useful).

We also asked for the decision of adopting the platform in other subjects about Computer Architecture (*Would you advise the use of PDIWeb in other subjects about Computer Architecture? Yes, No*). In this occasion, the 95.8% of the students, after evaluating the PDIWeb system, says that it should be used in other subjects about Computer Architecture. This question corroborates, and even it improves, the results obtained in the previous question where the platform was described by more than 93% of the students as useful.

Now, we focus on the results obtained when students are asked about if they would advise the implementation of this type of applications for other subjects (*Would you advise the implementation of applications of this kind for other subjects? Yes, No*). In this case, the answer is almost unanimous, the 97.9% supports this type of initiatives, and they see the development of educational contents (theory, exercises, etc.) through Internet as a need in order to improve the teaching quality of the subjects, independently of their topic.

Another question of interest is the following: *After having examined PDIWeb, what would your recommendation be for learning instruction dynamic scheduling? Only professor explanations, Prof. explanations + PDIWeb, Only PDIWeb*. The 97.9% thinks that the best alternative is to hold the conventional classes (professor's explanations) using the platform as didactic reinforcement. These results indicate that students evaluate positively the presence of the professor in the class. In conclusion, they think that the personal contact is important.

Not only the results obtained in the previous questions are good, but also the later comments of the students in a free-answer question were notably positive and encouraging in order to continue with this line.

5. CONCLUSIONS

In this chapter we have presented the PDIWeb system, which has been developed and it is applying for the teaching via Internet of instruction dynamic scheduling, topic given in the subject Computer Architecture and Engineering (CAE), in fourth course of Computer Engineering, at UEX. As it can be seen (section 4), the results of the survey about PDIWeb are very good, and they are exportable to other subjects.

As for the simulator advantages, it is important to highlight that, previously, this part of the subject was explained by means of problems in class. The carrying out of the problems on the blackboard limited the versatility and quantity of the problems. The development of this simulator allows the student to continue practicing the learned concepts at his/her own home, and at his/her own learning pace. Also, the quantity of practical exercises to carry out is only limited by the quantity of tests that the student wants to perform with the simulator. In fact, the simulators are the best way of understanding many of the practical concepts related with the design of processors, in which a great quantity of elements interact, making difficult the carrying out of problems. In conclusion, the simulator strengthens the concepts introduced in class, it gives the students a better appreciation of the internal work of a processor, and it motivates the students to carry out more practical exercises, redounding to an improvement of the teaching.

Finally, it is important to indicate that PDIWeb is not only applicable in the subject CAE at UEX, but also in other many subjects about Computer Architecture that are given at most of Universities, and in which the concepts about instruction dynamic scheduling are explained. In particular, it is possible to access to the simulator through the URL (Vega et al., 2006a).

ACKNOWLEDGMENTS

This work has been supported by *Vicerrectorado de Innovación Educativa y Nuevas Tecnologías* and *Instituto de Ciencias de la Educación* (University of Extremadura), under the frame of Projects of Educational Innovation.

REFERENCES

Cohen, S., 2004, *Macromedia Fireworks MX 2004 for Windows and Macintosh: Visual QuickStart Guide*, Peachpit Press.

Goodman, D., Morrison, M., 2004, *JavaScript Bible*, 5th edition, John Wiley & Sons.

Hennessy, J.L., Patterson, D.A., 2003, *Computer Architecture: A Quantitative Appproach*, 3rd edition, Morgan Kaufmann.

Lowery, J.W., 2003, *Dreamweaver MX 2004 Bible*, John Wiley & Sons.

McCarty, W., 2001, *PHP 4: A Beginner's Guide*, McGraw-Hill.

Morrison, M., 2002, *Faster Smarter HTML & XML*, Microsoft Press.

Sánchez, J.C., Sánchez, J.M., Gómez, J.A., September 2000, EDONET: Sistema Piloto para la Docencia a través de Internet, *VIII Congreso de Innovación Educativa en Enseñanzas Técnicas / I International Congress in Quality and in Technical Education Innovation*, Donostia-San Sebastián, Spain, vol. 2, pp. 25-33 (in Spanish).

The PHP Group, 2006, website for PHP (December 1, 2006); http://www.php.net.

Vega, M.A., Nieto, I., Sánchez, J.M., Chávez, F., Gómez, J.A., November 2002a, System of Tele-Education for People with Hearing Disability, *International Conference on Information and Communication Technologies in Education, ICTE'2002*, Badajoz, Spain, vol. 3, pp. 1261-1266.

Vega, M.A., Sánchez, J.M., Rubio, M., Gómez, J.A., November 2002b, Application of Information and Communication Technologies to the Teaching of Digital Systems, *Informatics Education 2002, VI Latin-American Congress on Computers & Education, IE'2002*, Vigo, Spain, pp. 1-6.

Vega, M.A., Burgueño, J.C., Sánchez, J.M., Gómez, J.A., 2006a, website for the PDIWeb Simulator (December 1, 2006); http://arco.unex.es/pdiweb.

Vega, M.A., Sánchez, J.M., Ballesteros, J., 2006b, website for the Subject CAE (December 1, 2006); http://arco.unex.es/mavega/AIC.htm (in Spanish).