Computers & Education 54 (2010) 562–577

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

A new *Moodle* module supporting automatic verification of VHDL-based assignments

Eladio Gutiérrez *, María A. Trenas, Julián Ramos, Francisco Corbera, Sergio Romero

Department of Computer Architecture, University of Málaga, 29071 Málaga, Spain

ARTICLE INFO

Article history:
Received 5 May 2009
Received in revised form 31 August 2009
Accepted 3 September 2009

Keywords:
Automatic assessment
Applications in subject areas
Distance education and telelearning
Learning management systems
Simulations

ABSTRACT

This work describes a new *Moodle* module developed to give support to the practical content of a basic computer organization course. This module goes beyond the mere hosting of resources and assignments. It makes use of an automatic checking and verification engine that works on the VHDL designs submitted by the students. The module automatically keeps up to date information about their state, and significantly reduces the overload that a continuous assessment demands to the teacher. Additionally, this new module is oriented to promote a collaborative teamwork allowing to define student teams in a more operative way than built-in Moodle groups. The module has been designed according to the *Moodle* philosophy and its application can be extended to other similar subjects.

1. Introduction

At the present time, university education is strongly influenced by new technologies. The impact of learning management systems (LMS) is particularly significant for science and engineering studies where they are but a particular case of the set of software tools applied in these areas, which emerge from the field of Computer Science. But LMS systems are more than just a tool. Their use has a significant influence on various aspects of the teaching/learning process. Their use encourages a more interactive learning and increases the possibilities of virtual teaching. They make easier to implement pedagogical approaches of constructivist nature, that are characterized by the construction of knowledge through the completion of different activities (Jonassen, 1999; Khalifa & Lam, 2002; Knuth & Cunnigham, 1993). This encourages self-learning with an special emphasis in real world problem solving. And this is essential as refers to engineering studies, as much of the work of an engineer will be to provide solutions in order to solve problems effectively. Collaborative learning (Tan, Lin, Yang, & Zhao, 2008) is a perfect complement to constructivist teaching. It will encourage teamwork, another important skill in real world, but also important as students will help each other, thus strengthening the learning process.

In general, LMSs provide web-based interfaces that support a wide range of activities. These include forums, content resources, questionnaires, chats, assignment and so on, which are, in general, sufficient for setting up standard courses. LMS may also integrate other tools of great interest when teaching an engineering course, as it is the case of the automatic checking and verification of the student's lab work. This is a very time consuming task if manually done by the teacher (Rice, Beg, Waters, Bokreta, & Santiago-Avilés, 1999), becoming an important overload. Its automation would greatly benefit the constructivist approach. But new technologies also bring negative effects, such as the increase of the chances of plagiarism among students (Lathrop & Foss, 2000).

Focusing on Computer Science and Engineering studies, and related to the teaching of both high and low-level programming, a number of approaches have been developed whose characteristics include automatic assessment (Bennedsen, Caspersen, & Kölling, 2008; Douce, Livingstone, & Orwell, 2005; Jackson & Usher, 1997; Malmi, Korhonen, & Saikkonen, 2002) and the detection or avoidance of plagiarism (Butakov & Scherbinin, 2009; Rosales et al., 2008). For example, a framework to teaching low-level input/output in mass courses by means of a specific platform can be found (Rodríguez, Zamorano, Rosales, Dopico, & Pedraza, 2007). However this is not so extended in the area of Computer Architecture, where the teaching tools mainly concentrate on the simulation, at the organizational level, of computer architectures and their assembler language (Djordjevic, Nikolic, & Milenkovic, 2005; Moreno, González, Castilla, González, & Sigut, 2007), with few developments on the simulation of the digital circuits at the low level (Kurmas, 2008; Poplawski & Kurmas, 2008). This is the case when

* Corresponding author. Tel.: +34 952 13 2821; fax: +34 952 13 2790.
E-mail addresses: eladio@uma.es (E. Gutiérrez), matrenas@uma.es (M.A. Trenas), julian@uma.es (J. Ramos), corbera@uma.es (F. Corbera), sromero@uma.es (S. Romero).

0360-1315/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2009.09.006

teaching a basic course on Computer Architecture, an essential topic in the Computer Science & Engineering curricula, which generally involves practical assignments related to the design and simulation of digital circuits and elementary processors. Such practical component is usually performed by means of real CAD tools making use of schematic or VHDL (Very high speed integrated circuit Hardware Description Language) design entries (Calazans & Moraes, 2001).

With the purpose of providing a better support for such a kind of practicals we have developed a specific *Moodle* module that provides automatic assessment features, allowing an advanced management of assignments and submissions together with a enhanced feedback information mechanisms for both teachers and students. All of this based on the automatic checking and verification of the student's lab work. Also it includes some organizational aspects oriented to promote student teamwork.

Motivations for choosing *Moodle* (Modular Object-Oriented Dynamic Learning Environment) include its modular organization based on free software whose sources are freely available (Main Moodle site: http://moodle.org). This enables the developer to add new modules using well-known interfaces. Additionally, *Moodle* has become a very popular platform with more than 50,000 registered sites at the beginning of 2009, and at present it is widely used in higher and university education (Martín-Blas & Serrano-Fernández, 2009). Besides, it is the platform currently used by the Virtual Campus of our university.

This paper is organized as follows. Next section introduces the context of the course the new module was developed for. Then, general Moodle development guidelines are presented, prior to a general overview of our module. Sections 5 and 6 focus on the user interactions with the system for both students and teachers, respectively. Due to its relevance, next section is devoted to the implementation of the automatic verification support, followed by sections presenting site configuration issues and other features. The paper ends with a section discussing our experience and some final conclusions.

2. Context of the course

Computer Technology is a basic course on computer architecture delivered in the Computer Science academic plan of the University of Málaga (Spain). It includes 3.6 practical European credits from a total of 7.2. The objective of the course is to introduce how the CPU of a computer system works at the register transfer level, covering the gap between the machine level and the assembler language. To this end, the laboratory assignments (Corbera et al., 2006) consist of the design and implementation of various selected elements of the central processing unit and memory system, as well as the whole control unit (both hardwired and microprogrammed) for a simplified MIPS based architecture (Patterson & Hennessy, 2008). The whole work is developed as a hierarchical schematic using Xilinx Foundation as CAD tool (Xilinx: http://www.xilinx.com).

In order to better guide the students, teachers should perform a functional evaluation using a simulation tool to verify whether the students designs work or not according to the specifications. This is a very time consuming task, specially when there is a high number of enrolled students (Rice et al., 1999). Therefore it commonly produces bottleneck when looking for a continuous assessment of students and even provide them feedback about their evaluation. And this is an issue to solve, if we are to implement the recently proposed European Credit Transfer System (ECTS).

The European Credit Transfer System is affecting all the higher education system with a special focus on the teaching quality (Clausen, 2005; Reichert & Tauch, 2003). Its successful implementation involves changing the teaching practice carried out traditionally. The goal is to exchange the exclusiveness of conventional master classes for a more partaking model, where the student's work plays a significant role. For this reason, teachers are required to evaluate the student effort (amount of effective work) and not only the amount of knowledge. Currently the University of Málaga is enrolled in a ECTS experience in order to address successfully the imminent implementation of the ECTS system.

In this context, a specific LMS was built for the Computer Technology Laboratory that has been in use from the academic year 2004 to 2007 which has been a great help in order to achieve most of the goals pursued by the ECTS experience (Gutiérrez, Ramos, Romero, & Trenas, 2007). This prototype was conceived to receive lab works that students must upload after their verification. Students know that the system will check their work against a battery of tests unknown for them. The prototype provided support for several aspects concerning the laboratory work: material repository, planning and deadlines, assessment assistance, working at home, improved feedback, students workload estimation, etc.

Nevertheless such a prototype suffered from various drawbacks. From the developer viewpoint, the system had a monolithic design based on a great number of *cgi*-style programs making it difficult to modify or improve. From the teacher point of view the system was initially configured for a fixed number of projects making difficult to propose new projects or make a deeper verification. Teachers could not modify, without the assistance of the system managers, neither the project definition nor the testing bench. From the student viewpoint, the system was isolated from the LMS platform used by our University, based on *Moodle*. This fact limited the interaction between students in the system because activities like forums and messages were outside the prototype.

All these reasons led us to design and develop a specific *Moodle* module supporting computer technology practicals, that is being used since the academic year 2007/2008. It inherits all features provided by the prototype system, but porting them to the *Moodle* philosophy. On integrating the activity in *Moodle*, teachers and students can benefit from all the advantages provided by this platform. Hereafter we will refer to this module as *CTPracticals*, after "Computer Technology Practicals".

3. Moodle development

Moodle is a course management system able of handling a large number of courses and users, as it occurs with a university center. It is freely provided as Open Source software under the GNU Public License, and it can be installed on computers running PHP, with SQL database support, as MySQL. A typical Moodle installation is made up of three elements: a directory for the PHP files constituting the source code of the application, another one with files containing data about courses and users, and a database which defines the different objects that integrate the system.

Moodle basic organizational unit is the *course*, which is acceded through a web page such as the one shown in Fig. 1. A course is organized into *sections* that may correspond to topics or weeks, appearing in the middle column of the page. In each section, it is possible to