PROJECT-BASED LEARNING ENVIRONMENT FOR SPECIAL PURPOSE DSP ARCHITECTURES

Ivan Milentijevic, Vladimir Ciric
Faculty of Electronic Engineering, University of Nis, Serbia
{milentijevic, vciric }@elfak.ni.ac.yu

ABSTRACT

Project-based learning is de-facto a great challenge for both students and instructors, and it can bring the number of advantages, or even disadvantages if suitable environment is not set. In DSP courses where special purpose architectures are studied, project-based learning should be carefully employed. On contrary to the approach in which student gets a project and instructor evaluates results of design efforts, this paper deals with development of environment for project-based learning, where instructors can see the process used to produce those results. In this paper we propose a low-cost, CVS-based environment for hardware oriented project-based courses. It enables good and fast startup for students' projects, and supports critical stages in learning by allowing students to study other students' work. System requirements are given, and the model is developed and described using a UML sequence diagram. Evaluation of the system has been done in the period of one year on the test group of 21 students.

1. INTRODUCTION

Coding is an important part of development processes However, professional and academic computer development communities agree on the fact that overall design process is of the greatest importance [1], [2]. Project-based learning is suitable for learning an overall design process by giving the students appropriate projects, or in other words assignments. However, if course, which is conceived as project-based, is not supported with the proper environment, the goal of "overall development process monitoring" can be easily omitted. Thus, project-based becomes coding-based learning.

The problem in project-based teaching environments is that, due to their nature, they can be quite chaotic. Thus, full interaction between students and instructors is hard to be achieved. Nevertheless, the role of collaboration among students during the development process is often neglected, or even forbidden, especially in education process [2], [3]. Furthermore, in such environments, it is important to be able to clearly see the process of development itself, rather then the result of the development.

At the heart of most modern commercial and open source software projects lays a version control system, as a system that allows multiple developers to share files in a safe, controlled environment, while automatically creating a history of the project's evolution [4]. Version control (also known as revision control or source control) is the management of multiple revisions of the same unit of information. It is most commonly used to manage ongoing development of digital documents like source code, art resources or electronic models and other critical information that may be worked on by a team of people. Changes to these documents are identified by incrementing an associated number or letter code, termed the "revision number" or simply "revision", and associated historically with the person making the change. Version Control System (VCS) is the system that provides version control [4], [5].

In this paper we will demonstrate an idea of using software version control methods in project-based learning of special purpose DSP architectures. We implemented version control as project-based learning environment in senior-level course. The course, which follows Parhi's book [6], is devoted to VLSI digital signal processing systems with emphasis on design and implementation. Having in mind that this course is an advanced one, and that prerequisite is knowledge in computer architecture area, VLSI design, and basic DSP algorithms, we employed project-based learning as the most suitable method that brings together algorithms and circuit designs for special purpose DSP applications. Stages, which each candidate has to pass working on the project, are shown in Fig. 1.

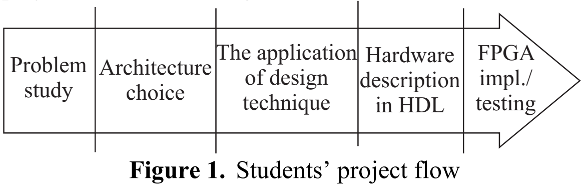
Problem study
Architecture choice
The application of design technique
Hardware description in HDL
FPGA impl./ testing

Figure 1. Students' project flow

Starting with problem related to choosing of project assignment by students, project-based learning without suitable support introduces monitoring problems in almost each stage from Fig. 1. In this paper we propose a low-cost, CVS-based environment for hardware oriented project-based courses.

The paper is organized as follows: Section 2 introduces version control terminology; Section 3 is devoted to system requirements development; Section 4 is the main section and presents the role of version control system in course organization; Section 5 is devoted to implementation results, while in Section 6 concluding remarks are given.

2. VERSION CONTROL SYSTEMS TERMINOLOGY

With the aim to clarify the role of VCS as a potential support to project-based courses, we give the basics of version control terminology.

The electronic documents within CVS are stored in, so called, *modules*. A term module stands for a single project (set of related electronic documents) managed by Version Control (VC) server. The VCS is established by adding a VC server to a local area network. Modules stored and managed by VC server are called *repositories*. In VCS terminology, acquiring a copy of a module is called *checking out* (the checked out files serve as a local working copy). The VCS server is usually equipped with security management software. Depending on privileges LAN users are either able or not to check out specific modules. Changes on the working copy are reflected in the repository by *committing*. Acquiring the latest changes from the repository in the working copy is called *updating* [4], [5].

As long as the VC server has both access and security set up, modules could be accessed from the same local network, or from a computer connected through Internet or another wide-area network, providing a good core for development of project-based learning environment.

3. VERSION CONTROL SYSTEM REQUIREMENTS FOR PROJECT-BASED LEARNING ENVIRONMENT

In order to establish project-based learning environment, we set the following system requirements:

R.1. Instructor defines assignments and creates the modules,
R.2. One-student - one-assignment,
R.3. Class and off-class access,
R.4. Monitoring and controlling the process of students' learning, instead of just being able to evaluate final results of assignment,
R.5. Student chooses assignment,
R.6. Security level enables collaborative work.

R.1 provides only the instructor with all the authority related to repository management, leaving the students with check-out, commit and update options only. This approach significantly reduces security management problems that appear in [2] and [3].

R.2 is set in order to make an evaluation of learning process straightforward. Giving one assignment per student does not mean that team working benefits are not exploited. In contrary, larger projects are decomposed by instructor in the set of suitable assignments, making all students to be part of a (relatively) large team.

Requirements R.3 and R.4 lay in the features of chosen VC server. The features of VC server will determine the way of implementation and the degree of achievement of requirement R.4. However, in order to provide R.5 and R.6, the following system parameters should be determined: (a) students' access method, and (b) security. Concerning students' access method, checking out, committing and updating files are straightforward basic operations [4], [5]. Furthermore, concerning the system access, it is expected that students are able to access a list of assignments/modules with aim to select an assignment (R.5). The solution is the choice of VC server that can provide a web interface with read-only anonymous access to its repositories. Web interface is a suitable solution because the modules can be visually grouped into different categories giving the students the role of a team member.

Security and students collaboration are tightly coupled. In order to allow collaboration, a system with as fewer forbidden or limited VC operations as possible, should be developed. More precisely, such a system should allow all students access to all available features within their own repository, and read-only access for other server modules, just in order to prevent accidental damages. This is straightforward because of the R.2 that well defines the 1-1 relation between students and repositories. In such a system, other students' modules represent an open source "knowledge base", so there is no need to protect access to the already done assignments, as suggested in [2] and [3]. Prerequisite is that assignments are set in the way that original students' work could be recognized. The mentioned approach gives the students' possibility to mine the knowledge from other students' assignments, to compare different problem solving approaches, and to find solution for a particular case, or in a word: to collaborate (R.6).

4. THE ROLE OF VERSION CONTROL SYSTEM IN COURSE ORGANIZATION

The role of VCS, along with associated web interface, during the assignment lifetime is shown by UML sequence diagram (SD) in Fig. 2. The SD consists of four timelines (vertical dashed lines). Two timelines stand for actors (teaching staff and student), while the other two are for system components. The diagram in Fig. 2 is divided by horizontal dashed lines into three sections. The first section delineates a process that begins with the creation of repository by teaching staff (TS) and that lasts until student checks out the repository; the second section is the main section devoted to the process where student works on assignment, while the third section stands for final report approval.

The first section within SD relates to repositories creation and checking out. The section begins with defining project topics by TS and creation of one repository per assignment. Project decomposition into assignments, suitable for students' tasks, is done by TS within this phase.

The central point of the first section in SD (Fig. 2) is student's choosing the assignment. It is already mentioned that VC server should be equipped with web interface that dynamically creates HTML response upon student's request (Fig. 2). The response should involve all repositories related to students' assignments which in turn are clearly marked as "done", "work in progress", or "available" [7]. Student chooses one of available assignments from the assignment list (Fig. 2). Upon choosing the assignment, student requests access to the specific project repository. TS creates an account for that student and gives account specific details (server address, protocol, username, password, etc) to the student. Using the received account parameters, student is able to access and check out files from the permitted repository. Security issue is to enable read-only anonymous access to all modules (web-http or any of VC protocols), and full access to authorized students for particular repositories (only through VC protocols). While checking out the files for the first time, student is provided with background material, starter code, and templates for reporting documents.

The second SD section is devoted to development. Once the project files are checked out, student is able to begin work on the assignment. In order to monitor project work as a process, all work is done in small "one-week" increments (Fig. 2). Incremental approach involves milestones once per week. In each milestone student has to write a short report that contains a list of performed actions, short plan for the next week, and a list of problems that have occurred during the previous increment. Changed files should be committed along with report delivery (Fig. 2). Group of activities, marked as Section 2 in SD (Fig. 2), is a repetitive procedure that ends with the evaluation of previous work.

A crucial decision within section 2 is the duration of the iteration. If iteration lasts too short, then students do not have enough time to work on assignment and to point out their skills. Thus, it can cause difficulties for both students and TS. For students it is difficult to work on assignment keeping in mind milestones and reports that should be written. For TS it is hard to evaluate the effort because of the shortness of working interval. On the other hand, using a too long iteration period, the idea of monitoring the process of student's work on project could be lost. The duration of iteration that we choose is one week (Fig. 2). Knowing the average complexity of student's assignments in our case, one week period gives

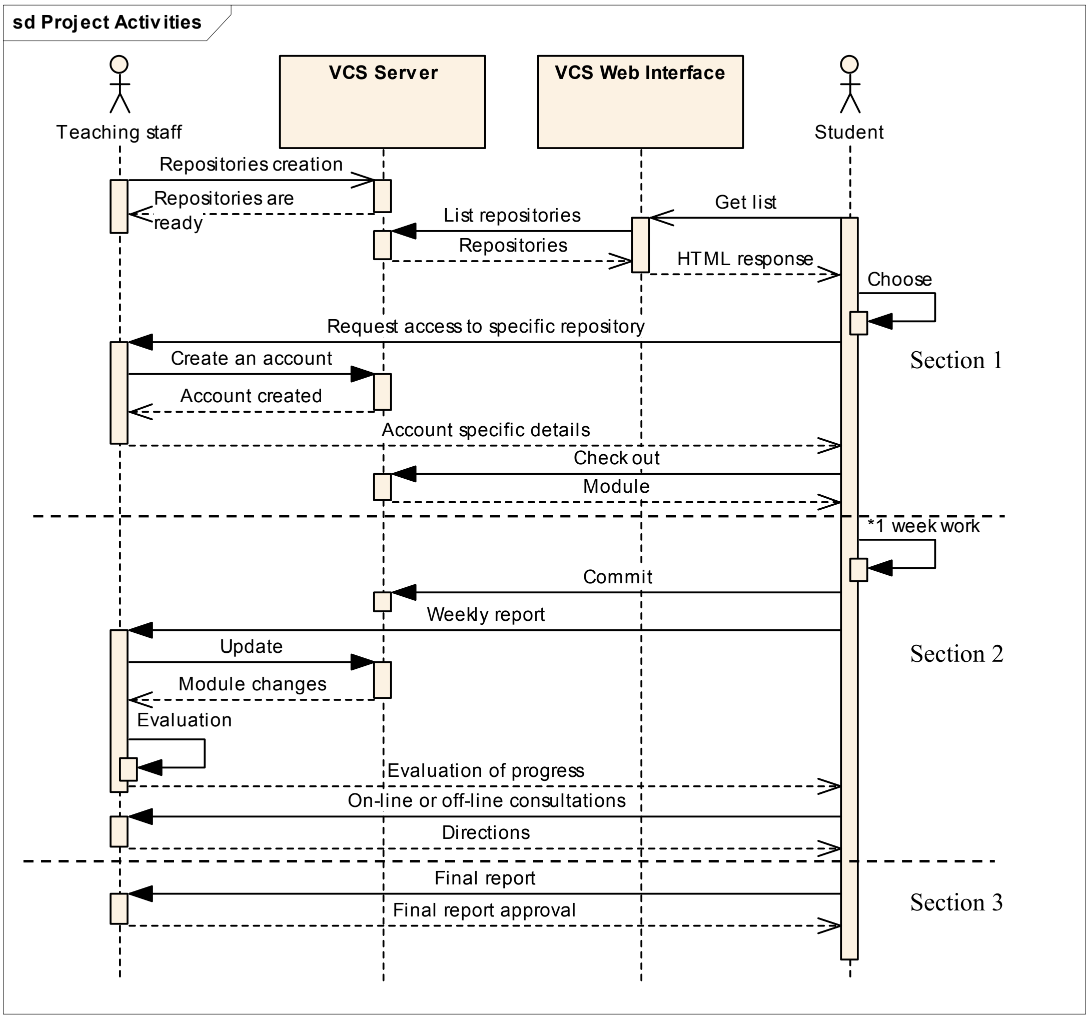

Figure 2. The role of version control system in course organization

the students opportunity to express themselves, while still allowing monitoring process to be performed.

Depending on the involved students' effort in assignment realisation, work on projects lasts no less than four weeks, and no more than eight weeks, including the first week that is devoted to background readings.

The proposed system security settings enable students to check out other students' repositories (currently under development, or done assignments), and to make their own project decisions based on other students' work. This is a point where students usually have questions for each other, rather than for the instructor. Therefore, with the aim to give students full support in their collaboration we provided web-forum as well.

5. IMPLEMENTATION RESULTS

In order to implement support for the students' project, with design flow shown in Fig. 1, we decided to use CVS [4], in the manner described in Fig. 2, on Linux platform, because it is well documented, open source VC software, with several web interface packages such as cvsview [4]. Subversion [5] is also a possible solution.

When CVS is used as a project-based learning environment for monitoring the students' learning process, as shown in Fig. 2, there is no need to consider security issues further from preventing a student modifying or accidentally destroying other student's work. Furthermore, the problem of cheating preclusion, which relates to modification of time-stamps on committed files by students, with aim to show better time performance [3], is avoided by involving written weekly reports (Fig. 2). The reports are useful not only to TS to get an overall picture of the previous work and possible problems, but also to students as they force them think about and summarize the results, thus giving a clue for the better start of the next incremental period.

For the test group of 21 students, we made a record of raised problems posted by weekly reports and by forum. Table 1 gives the number of students that faced typical problems (rows in table 1) in respect to assignment phases shown in Fig. 1 (columns). The table values are written as y/x, where x is the total number of students that faced the problem, while y stands for students that asked the instructor for help.

Table 1 Typical problems in respect to time spent on assignment

Week / *Problem*	**Probl. study**	**Arch. choice**	**Design technique**		**Hardware description**		**FPGA**
	1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**
Background	10/12	2/3	/	1/1	/	/	/
CVS access	3/6	2/2	/	/	/	/	/
Design decis.	/	7/9	8/10	9/18	1/12	/	/
Code-related	/	/	/	6/12	2/8	0/6	3/7

From Table 1 it can be seen that background understanding and CVS access problems increased mostly at the beginning of project. Having in mind project flow stages, we notice the following:

- Carefully selected scientific and technical papers by teacher, as well as recommendations for book chapters are provided by the system right after first login (Fig. 2). This enables good and fast startup for students' projects.
- As system provides to students anonymous read-only access to all finished assignments, it supports critical stages of the project such as *Architecture choice* and *Application of design technique* by allowing studying of other students work (Table 1). Leaving the finished assignments on CVS, we enable students to study the examples of different architecture design styles, to notice the differences in arithmetic, and to become aware of area-time-power compromising techniques.
- Version control tools enabled the learning process to be clearly divided into stages (Fig. 1) and successfully monitored (Table 1).

6. CONCLUDING REMARKS

The implementation of low-cost version control project-based learning environment in senior-level digital signal processing course is presented in this paper. The environment is based on CVS version control server. The major issues presented in the paper are: implementation environment of the platform; platform requirements with detailed comments; system settings and assignments of roles of all actors; comments on system security management. The system model was developed and described using a UML sequence diagram. It enables good and fast startup for students' projects, and supports critical stages of the project such as *Architecture choice* and *Application of design technique* by allowing studying the other students' work. Proposed environment enables the learning process in special purpose DSP architectures course to be clearly divided into stages and successfully monitored.

REFERENCES

[1] A. Breiter, G. Fey, R. Drechsler, "Project-Based Learning in Student Teams in Computer Science Education", Facta Universitatis, Series: Electronics and Energetics, Special Issue on Computer Science Education, eds. M. Stojcev and I. Milentijevic, Nis, Serbia, Vol. 18, No. 2, April 2005, pp. 165-180.

[2] L. Glassy, "Using Version Control to Observe Student Software Development Processes", Journal of Computing Sciences in Colleges, vol.21 n.3, February 2006, pp. 99-106.

[3] K. Reid , G. Wilson, "Learning by Doing: Introducing Version Control as a Way to Manage Student Assignments", Proceedings of the 36th SIGCSE technical symposium on Computer science education, St. Louis, Missouri, USA, February 2005, pp. 272-276.

[4] J. Vesperman, "Essential CVS", O'Reilly, USA, 2003.

[5] B. Sussman, B. Fitzpatrick, M. Pilato, "Version Control with Subversion", O'Reilly, USA, 2004.

[6] K. Parhi, "VLSI Digital Signal Processing Systems (Design and Implementation)", John Wiley & Sons, New York, 2000.

[7] http://L3.elfak.ni.ac.yu/algarh/, Faculty of Electronic Engineering, University of Nis, Serbia, 2006.