

Research meets Education: DRES D, a virtuous circle

Carlo A. Curino, Marco D. Santambrogio, Donatella Sciuto

Politecnico di Milano - DEI

Via Ponzio 34/5

20133 Milano, Italy

{curino, santambr, sciuto}@elet.polimi.it

ABSTRACT

Research and Education have been often perceived as a dichotomy. It has often been hard to couple them in a productive and virtuous cycle. With this paper we would like to discuss our attempt in this direction, briefly presenting the approach and the positive results obtained. The key idea is involving students, by means of projects and theses, in the research activities connected to the courses in their curricula. We monitored the activities and the productivity of the DRES D project, in the period 2004-2007, where students have been involved in the research activities of the group, and report the first encouraging results.

1. INTRODUCTION

While a suitable balance and a careful guidance of the students is of key importance, we advocate an holistic approach to the problem. This scenario creates a virtuous cycle that can transform a zero sum game, as the research/education is often perceived, as a win-win game, in which both the quality and result of research and the students experience are positively influenced.

Our goals can be summarized as follows:

- improve the student involvement in the research issues behind the disciplines of their curricula, in order to favor a more critical approach to learning;
- offer the research groups more design and coding *power*, increasing the productivity of the group;
- guide the students in this novel and critical approach;
- guarantee the high quality of prototypes and research results.

In the following we will introduce a platform for teaching hardware reconfigurability, discuss the key role of software refactoring to achieve high quality software, and discuss the experience of the DRES D project [1].

2. THE R&E PHILOSOPHY

We believe that Research can obtain great benefits from Teaching and the other way around. In particular, involving students in research activities will heavily increase the design and coding power of a research group. On the contrary, from an educative point of view, giving to the student the chance to be involved in real projects will mean giving them the chance to experience real design and development

challenges and by guiding them during the design and development we can, in a maieutic way, teach them how to approach real life projects. From our experience the students will be more motivated and will both learn more, and be more prone to research activities. Moreover the students will have the chance to see direct applications of their theoretical background in real cases. One can certainly argue about the quality of the produced solutions, it may seem the overall quality of design and code can decrease by unexperienced students designing core components. For this reason we advocate a continuous Refactoring [2] of the developed solutions. The overall quality of solutions will remain very high and the refactoring overhead is definitely less than the speed up obtained by heavily involving students in the design and development process. As a result the overall development quality will remain good, fully exploiting the work done by students together with the experience introduced in terms of refactoring by the teachers, while the students will have the chance to challenge themselves against difficult problems, discussing new solutions, and learning the critical approach w.r.t. of existing solutions, typical of research itself.

3. HOW: THE DRES D EXPERIENCE

As it has been widely studied and proved, many emerging applications in communication, computing and consumer electronics demand that their functionality stays flexible after the system has been manufactured; therefore the design of embedded systems has added the reconfigurable computing field as a new direction to enlarge the system solution space. The DRES D project focuses on reconfigurable computing for embedded system, trying to achieve the ambitious goal of introducing a complete methodology, that allows to easily implement on an FPGA, (Field Programmable Gate Array), a system specification, taking as input its high-level description, such as SystemC, JHDL [3, 4] and exploiting the capabilities of partial dynamic reconfiguration and hardware/software codesign methodologies.

The use of Object Oriented languages to teach reconfigurable hardware has already shown very good results during the last few years. The projects of some hardware courses at Politecnico di Milano [5] have been carried out using SystemC, a C++ derived HDL, as a development language. The results were really positive, although students are not really exposed to C++ during their curriculum, and Java has better type system, which enables a very good support of software tools, for debugging, refactoring etc.. In this sense, thanks to our experience with JHDL, we advocate

this language as an effective education mean. In our personal experience testing the JHDL approach to Hardware Modeling, the choice of Java as a modeling language has proved to be an effective one. While the conceptual differences of the procedural or Object Oriented programming approach with respect to the Structural Hardware Design remain the most difficult obstacle for traditional programmers who approach hardware design, the use of an already known language reduced the psychological impact of the new task.

We experienced also a second difficulty, the maturity of the tools, which is the typical drawback in the reconfigurable computing field. In fact, most of the research tools, although providing the needed features, lack simple add-on for the deployment and the seamless integration of the various components. In a sense although all the difficult tasks are accomplished successfully by the tools the designer still needs to be aware of too many details of the actual implementation of the tool-chain. The proposed approach combines the benefit of JHDL, the Xilinx tool chain, like ISE [6], and the DRES D tool-chain [7]. We think the overall framework to be an effective platform for learning and experiencing with Hardware design. This approach produced in the period 2004-2007 good research and education results. Figure 1 shows a graph of the number of students involved in the DRES D projects demonstrating how the reconfigurable computing area can attract a growing number of students. In such a context it is necessary to provide to the students

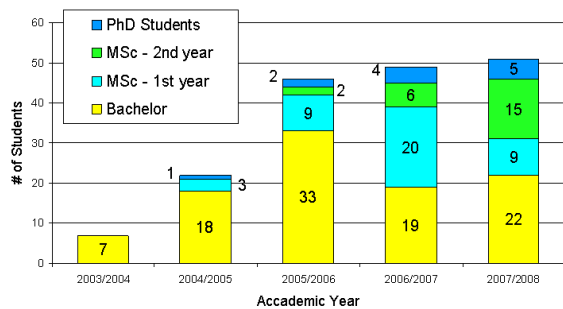


Figure 1: Students trend in DRES D

an environment where they can work and experiment a motivating experience and it is exactly at this point that the DRES D project plays a key-role with its activities and its design tool chain. Figure 2 shows also how the productivity of the team¹ can benefit from such a trend in the number of involved students. Figure 2 presents the number of papers, realized by the DRES D team, accepted to international conferences. The clear growing trend proves the effectiveness of involving students in research activities.

3.1 Managing the risks

Tightly coupling research and educational activities is not an easy task, a continuous monitoring activity is required. One of the most evident risk is that research often moves on the edge of a discipline in an area in which terminology and theory are still blurred and sometimes not clearly defined. This while being an acceptable scenario for experienced researchers can be a dangerous environment for students. In

¹the team has grown just in the number of students involved, without any increment in the number of faculties.

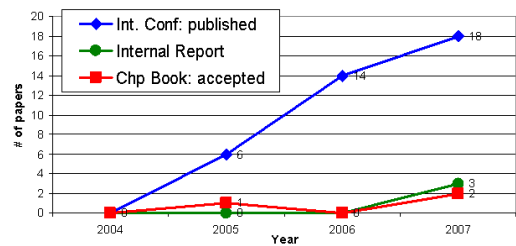


Figure 2: Number of papers published in DRES D

this sense we believe that is of key importance to provide a *protected* environment for the students providing them for example an established, clear and stable vocabulary. To this purpose we report the effort we devoted to establish an unambiguous terminology in the world of hardware reconfigurability. While working on the extension of the DRES D methodology [8] we realized that the field of reconfigurable computing is quite a young field and still there is a lack of common terminology. In this sense we dedicated some effort trying to formalize the terminology. The result of this process has been the definition of an OWL-DL [9] ontology. The current ontology specifies about a hundred terms, providing a brief description of each comment, and about 25 relationships (other than subclass). The ontology is still under development, and we plan not only to complete the class description, but also to *populate* the ontology with the most relevant reconfigurable architectures. The current ontology has to be considered an evolving dictionary, to be discussed and shared with other research groups to build a shared and agreed vocabulary.

4. REFERENCES

- [1] DRES D Team. Dred website - www.dred.org, 2007.
- [2] Martin Fowler. Refactoring. In *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, pages 701–701, New York, NY, USA, 2002. ACM Press.
- [3] Brad Hutchings, Peter Bellows, Joseph Hawkins, Scott Hemmert, Brent Nelson, and Mike Rytting. A cad suite for high-performance fpga design. In *FCCM '99: Proceedings of the Seventh Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, page 12, Washington, DC, USA, 1999. IEEE Computer Society.
- [4] P. Bellows and B. Hutchings. Jhdl - an hdl for reconfigurable systems. In *FCCM '98: Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines*, page 175, Washington, DC, USA, 1998. IEEE Computer Society.
- [5] Politecnico di milano website - www.polimi.it, 2007.
- [6] Xilinx Inc. *ISE Quick Start Tutorial*. Xilinx Inc., 2006.
- [7] V. Rana, M. D. Santambrogio, and D. Sciuto. Dynamic reconfigurability in embedded system design. In *IEEE International Symposium on Circuits and Systems*, pages 2734–2737, May 2007.
- [8] C. A. Curino, L. Fossati, V. Rana, F. Redaelli, M. D. Santambrogio, and D. Sciuto. The shining embedded system design methodology based on self dynamic reconfigurable architectures. In *13th Asia and South Pacific Design Automation Conference*, page To appear, 2007.
- [9] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview, W3C Recommendation, 2004.