COMPUTER ENGINEERING EDUCATIONAL PROJECTS OF MIPT-INTEL LABORATORY IN THE CONTEXT OF CDIO

Grigory Rechistov, Arnold Plotkin

Moscow Institute of Physics and Technology, Department of Radio Engineering and Cybernetics

ABSTRACT

Principles and results of ten years of operation of a research/education laboratory of Moscow Institute of Physics and Technology with Intel Corporation (MIPT-Intel lab) in context of conformance to CDIO standards are described. We describe organization of engineering projects of bachelor curriculum along with their assessment. It is shown that, while original principles of organization of these courses match CDIO ideas quite closely, certain amount of work has to be done in order to make such match complete.

KEYWORDS

Computer Science Education, Software Engineering, Microprocessor Engineering, Standards 2, 3, 7.

INTRODUCTION

Since its inception in 2003, the mission of the joint laboratory of Moscow Institute of Physics and Technology and Intel Corporation (MIPT-Intel lab) is to bring up software and microprocessor engineers from undergraduate students of sub-faculty of Microprocessor Technology, Department of Radio Engineering and Cybernetics.

To achieve this goal, acting software engineers from Intel lead educational and research projects as mentors and scientific advisors for students and interns. The projects' topics stem from practical industry problems, including such areas of system software and services as compilers, simulators, microprocessor architecture design etc. Later, students that showed good results are offered with internship that allows them to prepare bachelor theses in course of working on real world's projects at Intel, being a part of large software engineering teams. The MIPT-Intel lab is one of several educational centers Intel has within major Russian State universities (Intel Corp., 2013).

In April of 2013, MIPT joined CDIO initiative (Chuchalin, A.I., 2013). Therefore, an assessment of existing approach was required to make sure that our educational, engineering and research principles meet existing CDIO principles and standards. In this paper, we first describe the laboratory's original philosophy and methods of operation. Then we proceed to demonstrate matches and mismatches of our approach against CDIO requirements. We argue that while the MIPT-Intel lab meets certain CDIO principles from its beginning of operation, there is still a lot of work that has to be done. A number of changes are needed not just inside the laboratory itself, but also at the department and faculty levels.

ORGANIZATION OF EDUCATION PROJECTS

This section describes activities that students and their mentors carry out during four years of educational program as of its current state.

THE FIRST YEAR— PROGRAMMING LANGUAGES

The first year is dedicated to teaching computer languages: C and C++, which are essential for system software engineering. Our experience shows that Computer Science background of recent middle school graduates differs greatly in what programming languages they know and at what depth.

Therefore, our goal set for the first year is to bring participating students to the same terms with essential programming languages, tools and concepts. The main learning outcomes correspond to section "1.2 Core Engineering Fundamental Knowledge" of the CDIO Syllabus (Crawley, E.F., et al., 2011)

The mentors of the first year are assigned to small (3–6) groups of students to aid them in completing individual tasks. Mentors are to carry out some basic routines of a typical software project: keep revision control system in order, perform code reviews, write documentation etc. It should be noted that a significant part of mentors are students of 3rd and 4th years themselves. Therefore, mentoring activities help older students to develop skills of section 3 "Interpersonal Skills" of the Syllabus, including teamwork, leadership and communications in oral and electronic forms, and preserve continuity of education

THE SECOND YEAR—TEAM DEVELOPMENT PROJECTS

In the second stage of the laboratory operation, students choose one of projects (Baida, Y.V. et al. 2010) backed up by one or several mentors. During the year they cooperate inside their corresponding teams to come up with solutions to problems presented to them. Themes of projects differ slightly from year to year, depending on whether there are active mentors willing to support them, and are from the following areas:

- microprocessor design;
- prototyping of hardware on FPGA;
- software simulation technologies;

- compiler technologies.

The essential point is that the 2nd year education and research processes are organized exactly the same way they are done in real-world commercial/open source software projects. Wide popularity of free code hosting sites makes such decision natural and easy, as it helps to cut down infrastructure maintenance efforts. Below is an outline of tools and processes of a typical student project.

- The main collaboration sites: Google Code (Google Inc., 2014) and Github (Github, 2014). All projects are licensed under one of the sites supported free software licenses, either GPL or MIT.
- Source code storage: revision control systems (RCS). We use Subversion or Git. While initially there were concerns that the second RCS, namely Git, would turn out to be too complicated for students, practice showed that such fears were groundless.
- Documentation: Wiki pages, edited by team members to keep technical knowledge (Titov, A.I. et al., 2012).
- Bug tracking system: Issue trackers integrated into hosting sites.
- Communication: through emails, per-project mailing list, instant messaging and teleconferencing services . Weekly contact hours are used to discuss

A number of activities that correspond to the Syllabus are set as goals for this year. We want to underline sections 1.3 “Advanced Engineering”, 2.1 “Analytic Reasoning and Problem Solving” and 3 “Interpersonal Skills”, including 3.3.1 “Communications in English” as we strive to keep all project’s documentation and most of written communication to be in English.

The team of mentors of the second year comprises of the MIPT-Intel lab graduates, recent alumni and PhD students of the department who are interested in raising the next generation of engineers. Many of them have passed through the very same education program just recently. They still remember their own missteps, troubles and victories and are eager to share their experience on how to overcome problems and achieve success.

THE THIRD YEAR—CORPORATE ENVIRONMENT

Students that showed enough skill and diligence after two previous years are offered an intern position at Moscow office of Russian branch of Intel Corporation. Each of them is assigned to one of several engineering teams and takes part in development of computer language compilers, software simulators, binary translation tools or microprocessor design. The projects are large scale, have both internal and external business customers, numerous policies on style, quality, processes etc. A great deal of written and oral communication is carried in English with native speakers. Thus, students are deeply submerged into real everyday work they are going to have in their career.

One of their colleagues is assigned as a “buddy”, and his/her mentoring is dedicated to help the student to learn tools, practices and culture of the company. Additionally, each intern has regular meetings with his/her project manager, and is always welcome to ask other team members for advice or help. Tasks that interns are assigned with are carefully chosen to match their skill and time resources, as they are still have many other courses to attend to outside of internship hours. Yet, it is essential that such tasks have to be parts of the teamwork.

During this year, participating students are expected to learn and demonstrate many items of the Syllabus from section 3 “Communications” and section 4, especially: 4.1.1 “Roles and Responsibilities of Engineers”, 4.1.6 “Developing a Global Perspective” and 4.2 “Enterprise and Business Context”.

THE FOURTH YEAR—RESEARCH AND DEVELOPMENT PROJECT

The final year of bachelor curriculum assumes writing and defending a thesis on an engineering/research project. Each student, which has finished three previous years of the program to the point and has gained some experience in the software/hardware engineering field, is given with an individual task that arises from a practical need of his/her team and product/technology it develops. It can be adding and validating new functionality to the software, finding weak spots and improving performance of its subsystem, realization and testing of new algorithms under real world conditions, etc.

In the process of completing the bachelor thesis, a student is expected to demonstrate capabilities to work both individually and in team, to do literature surveys, hypothesis formulation and testing, employ deep knowledge of science underlying the technology in question. A great deal of students’ efforts has to be put into prioritization and numerous trade-offs, as they are still faced with a requirement to attend lectures and seminars, not just doing their theses. In general, this year allows them to demonstrate abilities to Conceive, Design and Implement.

Each student has a scientific advisor, which has to be member of the same team, and is responsible to help achieve the goal of successful graduation.

STUDENTS INVOLVEMENT AND RETENTION

Table 1 shows average numbers of students participating in the MIPT-Intel laboratory courses, depending of the year of education. While it is out of this paper scope, data on master and PhD students is also represented there.

Table 1. Summary participation rates for period of 2003–2013

Parameter	Value	Comments
Enrolled at first year	60	First pilot started at 2011
Enrolled at second year	30	The main year for experimenting with new ideas on courses’ contents
Selected for internship (3rd and 4th year)	10	Students selected for internship
Graduated with bachelor degree	9	

Graduated with master degree	5	2 year program at the department
Started PhD course	3	3 year program at the department

The seemingly low retention rate demonstrated in this table (apparently only nine students out of 60 graduate) should be explained. At the end of the second year, all students of the Department are distributed between sub-departments, which offer specializations in engineering areas. There are limited enrollment quotas for sub-departments. Therefore, we can only accommodate about ten of students for 3rd and 4th years. Halving of students' number after the first year course can be attributed to several factors, including high pressure they experience from concurrently carried courses on other subjects (more of it will be discussed in the "Open Questions" sections below).

The most valuable insight our department obtained from these facts is that students, after two years, can decide whether they really want to work in the area of software engineering. Some of them decide to start pursuing different career opportunities in other technical/scientific areas that the Department offers. Conversely, students who stayed after two years have an increased commitment to the particular field of engineering which suites their passions and talents.

CDIO CONTEXT

In this section we focus on the selected CDIO Standards that most closely match goals of the MIPT-Intel lab.

STANDARD 2—CDIO SYLLABUS OUTCOMES

Each year of the MIPT-Intel lab has its own learning outcomes formulated and attached to its curriculum. It provides activities for students to help achieve these objectives. The outcomes develop naturally from basic software programming skills (year 1), through communication in numerous formats accepted in the industry and skills of programming in teams (years 2 and 3), to abilities to plan and build complex systems both individually and in teams (years 3 and 4).

Validation of students' skills is performed at the end of every year both by formal quizzes and by tests, and, most importantly, by demonstration and assessment of projects they worked on by their mentors, peers and management.

STANDARD 3—INTEGRATED CURRICULUM

Disciplinary courses that are read to students are: Introduction to software engineering, Compiler technologies, Computer languages C/C++ and Java, Integrated circuit design, Binary translation technologies, Computer architecture and Software simulation. Additionally, students have freedom to choose one or more of the following classes that suit their interests and career plans: FPGA design, MIPS CPU simulation, Advanced software simulation, Scheme complier implementation. Each year mentors typically offer, as experiment, a couple of new courses for the 2nd year projects. Some of them only last for one year, but some stay. Examples of such courses are: Multimedia codecs engineering, Parallel simulation, HTML Web workers, Compilers for LLVM framework.

Each of mentioned classes targets a specific practical area of computer software and hardware engineering. As we already described earlier, the integrative component of the education consists in that students are engaged in real world projects, using real world tools, both for producing final product and for communicating, and are subjected to real world issues of engineering: management peculiarities, imminent deadlines, requirements of peer reviews, quality assurance and automated testing, resource constraints etc.

STANDARD 7—INTEGRATED LEARNING EXPERIENCES

Integration of real-world engineering into education process was the most important goal of the MIPT-Intel lab even before joining the CDIO Initiative. Therefore, a number of our efforts were tailored to ensure conforming to this standard. What we want to prepare is a young engineer able to work in the industry immediately after graduation, and is able to grasp both business and social aspects of engineering. We seem to achieve this goal well. To back our claim it should be noted about 50% of interns participated in the lab were later employed at the Moscow's office of Intel.

CDIO SYLLABUS

The correspondence of steps of the MIPT-Intel lab educational program to certain items of the Syllabus was already shortly demonstrated in previous sections of this paper. Here, we want to summarize it in the Table 2.

Table 2. Correspondence of CDIO Syllabus (condensed) items to years of education at the MIPT-Intel lab. Levels of expertize students are expected to learn/demonstrate: A—aware, F —familiar, U—utilize

	Year 1	Year 2	Year 3	Year 4
1 Disciplinary Knowledge And Reasoning				
1.1 Knowledge Of Underlying Mathematics And Science	U	U	U	U
1.2 Core Fundamental Knowledge Of Engineering	F	U	U	U
1.3 Advanced Engineering: Fundamental Knowledge, Methods And Tools		A	F	U
2 Personal And Professional Skills And				

Attributes				
2.1 Analytical Reasoning And Problem Solving	F	U	F	U
2.2 Experimentation, Investigation And Knowledge Discovery	F	U	F	U
2.3 System Thinking	A	F	F	U
2.4 Attitudes, Thought And Learning				F
2.5 Ethics, Equity And Other Responsibilities			A	A
3 Interpersonal Skills: Teamwork And Communication				
3.1 Teamwork	F	U	U	U
3.2 Communications	U	U	U	U
3.3 Communications In Foreign Languages	F	U	U	U
4 Conceiving, Designing, Implementing, And Operating Systems In The Enterprise, Societal And Environmental Context				
4.1 External, Societal And Environmental Context			A	A
4.2 Enterprise And Business Context			A	F
4.3 Conceiving, Systems Engineering And Management		F		U
4.4 Designing		A		U
4.5 Implementing		F	F	U
4.6 Operating		A	A	F
4.7 Leading Engineering Endeavors			A	A
4.8 Entrepreneurship			A	A

OPEN ISSUES AND WAY TO RESOLVE THEM

Now, we would like to focus on two issues, which, as we see them, are limiting efficiency of our educational programs. Then we will outline an ongoing effort to restructure bachelor curriculum.

EDUCATION FOCUS

The focus of education at MIPT is defined by historical concepts; it is shifted towards preparing engineers in the field of applied physics. Those legacy principles can distract from the main goal—teaching how to become an engineer. It implies studying a number of fundamental disciplines, e.g., General and Theoretical physics, Mathematical analysis of functions of many variables, Complex analysis, Partial differential equations etc. These courses are mandatory for all students of all faculties. While such fundamental knowledge is valuable for engineering students in order to have adequate vision of nature and its manifestations, and is critically essential for other faculties at MIPT (e.g., for the Department of general and applied physics), the area of software engineering does not benefit from it directly. We feel that a better job can be done at balancing fundamental and engineering classes. Informal surveys of recent alumni support our opinion—many of them wish that they spent more time on building team skills and learning engineering context during their student years at MIPT.

Introducing a factor of variability into curriculum, when students can choose, to some extent, which disciplines they will attend, will help to free their time for more targeted studying of practically important engineering subjects. In the same time, this will allow to keep fundamental science classes for those willing to invest their time in them.

INNOVATION ACTIVITY

The sections 4.7 “Leading Engineering Endeavors” and 4.8 “Entrepreneurship” of the CDIO Syllabus are usually bound to draw special attention at time of a curriculum construction. Historically not much importance was given to the subject of entrepreneurship in context of engineering professions at MIPT. It is obvious now that such knowledge is critical for our graduates, because many of them start their careers within startup companies or even create their own businesses.

TOWARDS CHANGES IN CURRICULUM

Surprisingly, some answers to demonstrated issues can be found in existing master’s programs of the Department. At the master degree level, sub-departments, including our own, already offer courses that are built around needs of specific engineering and industry areas. Their experience is now being transformed to bachelor curriculum. A hard part to make this happen was to persuade stakeholders that we need to shift students' first encounters with engineering experiences from last years of master to first days of bachelor.

To find an approach towards specifying (or reformulating) our own syllabus for system programming and computer hardware design, we work to study best practices found by universities that participate in the CDIO Initiative from its inception, e.g., Chalmers University of Technology, and its Mechanical Engineering programs (Chalmers, 2011).

Existing educational plans at the university and sub-faculty levels did not historically include business aspects directly into education plans. Fortunately, there are recently emerged courses on entrepreneurship at the Department level, which are now offered for bachelor program students.

CONCLUSIONS

In this paper, we have demonstrated how the education program of the MIPT-Intel lab corresponds to the goals, methods and approaches of the CDIO Initiative. We focused on aspects tied to syllabus outcomes, integrated curriculum and learning principles of our work.

While we demonstrate strong correspondence between our education approach to engineering and ideas formulated in CDIO documents, we feel that, to improve efficiency of education significantly, these ideas have to be accepted and adopted on higher levels, including departmental and faculty levels. Achieving this constitutes our strategic goal: creating strong impact of CDIO ideas to education transformation at Moscow Institute of Physics and Technology.

ACKNOWLEDGEMENTS

The authors thank anonymous reviewers for their thorough feedback that was invaluable for improving this paper.

REFERENCES

Baida, Y.V. et al. (2010). The experience of operation of joint MIPT–Intel educational laboratory [in Russian]. *Proceedings of the 53th MIPT Conference “Present problems of fundamental and applied sciences”,* Moscow, November 2010, Vol. 12, Pp. 34–35.

Chalmers University of Technology, (2011). Programme syllabus – The Mechanical Engineering Programme (300 ECTS, B.Sc+M.Sc)

Chuchalin, A.I. (2013). TPU-Skolkovo Project: Modernization of BEng Programs in Russia. *Proceedings of the 9th International CDIO Conference*, Massachusetts Institute of Technology and Harvard University School of Engineering and Applied Sciences, Cambridge, Massachusetts, June 9–13, 2013

Crawley, E.F., et al. (2011). The CDIO Syllabus v2.0. An Updated Statement of Goals for Engineering Education. *Proceedings of the 7th International CDIO Conference*, Technical University of Denmark, Copenhagen, June 20–23, 2011

Github, Inc. (2014). Github. Build software better, together. https://github.com/ [online, accessed April 9, 2014]

Google, Inc. (2014). Google Code Project Hosting. http://code.google.com/ [online, accessed January 27, 2014]

Intel Corp. (2013). The list of Intel student laboratories in Russia [in Russian]. http://www3.intel.com/cd/corporate/education/emea/rus/highered/student/373553.htm [online, accessed November 14, 2013]

Titov, A.I. et al. (2012). Functional and performance simulation of a Multimedia Digital Signal Processor (MDSP) Wiki. http://code.google.com/p/mdsp/wiki/TableOfContents [online, accessed April 10, 2014]

BIOGRAPHICAL INFORMATION

Grigory Rechistov, PhD, is a senior tutor of the Microprocessor Technologies sub-department, Department of Radio Engineering and Cybernetics, Moscow Institute of Physics and Technology. His current educational and research activities focus on system software engineering, educational projects in system programming and software simulation technologies.

Arnold Plotkin, Dr Tech Sciences, Prof., is a deputy director of the Microprocessor Technologies sub-department, Department of Radio Engineering and Cybernetics, Moscow Institute of Physics and Technology. He is responsible for development of educational programs on system software and microprocessor research for bachelor and master programs.

Corresponding author

Grigory Rechistov

Department of Radio Engineering and Cybernetics (FRTK)

Moscow Institute of Physics and Technology

9 Institutsky per.

Dolgoprudny, 141700

Moscow District

Russia

Phone +7 495 6414500 ext 5361

grigory.rechistov@phystech.edu