Designing and Implementing International RSIC Engineering Curriculum

Ondrej Rysavy[1], Miroslav Sveda[2], Andrew J.Kornecki[3], Thomas B.Hilburn[4], Wojciech Grega[5], Adam Pilat[6], Jean-Marc Thiriet[7]

[1] *Brno University of Technology, Czech Republic, rysavy@fit.vutbr.cz*

[2] *Brno University of Technology, Czech Republic, sveda@fit.vutbr.cz,*

[3] *Embry-Riddle Aeronautical University, USA, kornecka@erau.edu,*

[4] *Embry-Riddle Aeronautical University, USA, hilburn@erau.edu,*

[5] *AGH University of Science & Technology, Poland, wgr@agh.edu.pl,*

[6] *AGH University of Science & Technology, Poland, ap@agh.edu.pl,*

[7] *Grenoble Université, France, jean-marc.thiriet@ujf-grenoble.fr,*

Abstract

This paper provides an overview and a comparison of two educational projects which focus on research and development of international computing curricula. The projects address the area of real-time software intensive control systems where the demand for highly educated and responsible engineers is highly evident. Issues related to the analysis, design, implementation and assessment of such curricula are discussed. The paper also deals with the exchange of students and faculty between international academic institutions, which is the major key to accomplish a truly globalized form of education that may face the new industrial demands.

1. Introduction

The analysis, design, implementation, administration, and assessment of international curricula are becoming increasingly important in the global community of the 21st century. In support of this critical issue, the European Commission and the US Department of Education have funded the ATLANTIS initiative to promote collaboration in higher education between European and American universities.

Recently completed, the ILERT (International Learning Environment for Real-Time Software Intensive Control Systems) ATLANTIS project has been involved in the creation of a bachelor degree international curriculum framework centered on RSIC (Real-Time Software-Intensive Control) systems. The ILERT study explored a mechanism for involving students from multilingual, geographically separated institutions in a coordinated educational experience. The ultimate objective of this two-year ATLANTIS project was creation of an RSIC curriculum model, which can be used by engineering schools both in the USA and the EU.

The main objective of the currently launched ATLANTIS continuation project DeSIRE[2] (Dependable Systems International Research and Education Experience) is to establish a platform for a sustained and consistent mobility exchange of graduate students engaged in RSIC oriented programs. These programs need to produce graduates capable of working efficiently in multidisciplinary teams participating in international collaboration on industrial RSIC projects, which require conformance to specific standards mandated by regulatory authorities.

Roughly a decade ago, one of the author participated in an IEEE sponsored activity that aimed to design a curricula that would be acceptable as undergraduate and graduate degree programs leading to the computer-based system engineer diploma. Although the approach was different, the targets and possible outcomes can be compared and the project can be treated as a source of experience in a certain sense. The project was conducted by the Working Group on Education and Training of the IEEE Computer Society

978-0-7695-3759-7/09 $25.00 © 2009 IEEE
DOI 10.1109/ECBS-EERC.2009.6

ECBS Technical Committee [2]. The objective of the proposed program was to educate and train its students giving them a good understanding of the nature and challenges of the engineering of computer-based systems. They would graduate as engineers capable of defining, developing, implementing, maintaining and evolving complex computer-based systems, using available theoretical and practical methods, techniques, tools and standards. Although CBS spans a wide range of applications, such as telephone and communications systems, real-time embedded computer systems such as process control and computer integrated manufacturing systems, transportation systems, commercial management information systems such as airline reservations, payroll information, stock control, electronic banking systems; avionics systems, missile control systems, and medical instruments, all of them include dependability and safety awareness aspect as one of the most crucial factor confronting the process of development, implementation and deployment of CBSs.

The paper presents in the two next sections the information on both projects. Following section provides a comparison of both approaches from different viewpoints. Both projects differ by mean of implementation approach, as the first project was oriented to multiple implementations of the curricula in a single institution and the second project proposed to deploy the curricula as spanned over more than one institution and to employ the student's mobility. However certain fundamental aspects behind the both projects are at least similar and share the same vision. We will point out both the differences and similarities in the comparison section.

2. ECBS Bachelor program

The aim of the ECBS Bachelor program [3] was to educate students about the nature and challenges of the engineering of computer-based systems and applications. After Bachelor degree curriculum implementation, the Master degree implementation is considered to be improved into a more advance form. The program is based on a model that concentrates on a role of CBS engineer that have understanding of system engineering methods and practice and have knowledge and capacity to in software engineering discipline at the same time. In particular, it is responsible for assist the system engineering process and to transfer the information from system models and processes to the computing domain and to propose and design a functional architecture of the CBS and propose and communicate the methods that would be consecutively implemented by software engineers. Later, at the implementation and deployment phases a CBS engineer is responsible for assisting both system engineer to verify the requirements and software engineer to deploy and adjust the CBS to the actual environment.

2.1. Learning Objectives

The objectives of the ECBS program is to give an education and a training to bachelor students in order to make them understand to the challenges of the engineering of CBS. They should be able to define, develop, implement, test, and maintain CBS, applying proper techniques and methods, and using available tools. An inevitable part of education is to emphasize on standards and codec of professional conduct. Also, it is important to develop soft-skills, such as interpersonal communication, ability to explain and demonstrate problems, and team working. Due to the nature of bachelor study program, it is important to balance the degree of knowledge required to gain. The learning objectives are classified in the three levels, as follows: i) the graduate is aware of the problem, ii) the graduate understands the problem, and iii) the graduate can apply a proper engineering method to solve the problem.

The key learning objectives required to be mastered by the graduate on the highest level are defined as follows:

- Develop conceptual models and operational scenarios of CBS
- Elicit and formulate the CBS requirements, suggest the feasible CBS architecture
- Design underlying CBS system and determined a HW and SW functionality
- Analyze the behavior and performance of the CBS design
- Specify the interfaces between the CBS subsystems
- Integrate small scale CBS and test it
- Prepare validation and test plan for CBS and their subsystems
- Recognize the need for further professional education in CBS area
- Be able to actively participate in multi-disciplinary teams

Even from sketched information of learning objectives presented, it is apparent that engineering education necessary includes both technical and non-technical skills.

2.2. Program Structure

The topics to be taken during the ECBS undergraduate program are divided into the following groups:

- Mathematics – engineering mathematics, calculus, mathematical logic, discrete mathematics, probability and statistics.
- Sciences – general physics, and a selected more specific area, such as general chemistry, biology, biochemistry, or biophysics.
- Engineering – mechanics, fundamentals of electrical engineering, electronic circuits, and digital system design, fundamentals and controls, information theory and signal processing, system simulation and modeling.
- Computing – programming, computer algorithms and data structures, operating systems, parallel and distributed systems, embedded systems and real-time systems.
- Communication Engineering – computer communication principles, computer and data networks.
- Software engineering – software architectures and components, software analysis, design and CASE tools, software verification and validation.
- Computer-Based Systems Engineering – CSB methods, modeling, requirements engineering, CBS architectures and design patterns, design and performance analysis, CBS integration, verification and validation.
- Engineering management – CBS project management, procurement and contracts, process, standards, and ethics, organizational patterns.
- Supporting Processes and Skills – communication skills, leadership, teamwork, and engineering documentation.
- Electives – any of the more advanced topics from the related areas, such as advanced database technology, computer graphics and image processing, formal methods, robotics and automation.
- Applications and Projects – contemporary technologies seminar, individual projects, final group project.

As evident from this brief list of topics proposed for the program, the implementation requires integrating courses from different departments as it is unusual that a single department would have capacity to offer all the mentioned topics.

2.3. Program Implementation

The program was implemented in participating institutions. We provide some experiences we learned from the implementation of the program at Brno University of Technology one of the leading technical universities in the Czech Republic [1]. Although the details of the implementation process may be different depending the university capacity, orientation and national difference, we feel that the provided information is relevant, in general. The lessons we learned about implementing the ECBS study fall mainly into the general categories such as "feasibility" and "attractivity," and only sporadically into more technical categories. As the most interesting, the following items appeared:

- It was feasible to launch the ECBS program by a single department, with a small number of students, and in frame of a similar discipline--in this case Computer Science and Engineering (CSE).
- The faculty of the Department of Computer Science and Engineering was attracted by an ECBS program that can utilize the technology-oriented environment of the Technical University to complement a traditional CSE program.
- There was only a partial interest among other departments to participate in the ECBS degree; initially, only the Department of Cybernetics and Automatic Control, the Department of Microelectronics, and the Department of Biomedical Electronics declared their intention to take part in future ECBS program development. The reason may stem from professional competition, because the ECBS degree has been presented systematically as an IEEE-CS initiative.
- Surprisingly, the biggest problem appeared to be how to attract students to a totally new branch of study, where other "classical" branches were more appealing. Because the overall impression we gained from the students involved in the ECBS program was of general enthusiasm, we believe we can overcome this problem in the future by a systematic advertisement and by the awaited increase of attention of local industry.
- The problem of relating courses to one another is a problem that has to be addressed from the beginning in each curriculum implementation, both to minimize repetition and to rely on abstracting ideas and concepts and reapplying them. We strove to solve this problem by designing tracks of related courses.

Unfortunately, this branch of study in Brno was stopped after 3 years. It was mainly because of funding reasons and also due to lack of support from faculty to pursue the program into the next regular accreditation. It appeared that it was not further possible to cope with this complex program within the limited number of faculty and without the engagement of the other departments of the university and at the same time to provide the program to larger number of students. It revealed that such in-house approach is not always applicable as the short of resources significantly limits the scope of the program implementation and the number of students involved.

3. RSIC Bachelor Program

The Real-Time Software Intensive Control System (RSIC) bachelor program as proposed in the frame of ILERT project (see [4] and [7], for details) is oriented to delivery an international educational experience in the area of safety-critical computer based systems. It aims not only to provide a set of learning objectives but also to develop a methodology for the implementation of the curriculum in an international fashion respecting the national differences of the educational process. It should be an answer to the industry that calls for the graduates that are able to work on the complex projects in international teams that involves understanding and applying diverse standards, guidelines and processes.

3.1. Learning Objectives

The classification of learning objectives is done by defining only two classes. One is a “know how” that describes a task one performs. Another is a “knowledge” that describes a topic that one understands and is able to convey knowledge about.
Expected graduate’s profile contains the learning objectives defined as follows:

- Demonstrate professionalism in work and grow professionally through continued learning
- Contribute to society by behaving ethically and responsibly
- Communicate effectively in oral, written, and newly developing modes and media
- Assume a variety of roles in teams of diverse membership
- Demonstrate understanding of analysis and design to implement software-intensive systems
- Demonstrate understanding of analysis and design to implement control systems
- Apply advanced software engineering techniques to implement real-time concepts
- Implement a rigorous quality assurance process
- Implement hardware{software integration
- Demonstrate knowledge of the principles and techniques needed for the analysis and design of a system form dependability perspective
- Use a well-defined development process

As it is evident, the stress is put also on non-technical skills, which is required by expected position of the graduate in an international team. The individual objectives were derived on the basis of results of industrial survey [5] done at the beginning of ILERT project. From this survey, the highest ranked items in technical field are:

- Knowledge of software design and development techniques.
- Knowledge of quality and reliability techniques.

In non-technical field the ideal graduate should posses the following skills:

- Ability to understand the problem, analyze it and bring up the solution.
- Team working on the one side and ability to independently solve the problems on the other.
- Communication skills, ability to address critical comments.

3.2. Program Structure

The basic organizational unit for the framework is a RSIC “component”. A RSIC component is a curriculum unit which covers theory, knowledge and practice which supports the RSIC curriculum objective and outcomes. Table 1 describes the RSIC components in six identified RSIC areas: Software Engineering, Digital Systems, Computer Control, Real-Time Systems, Networking, and Systems Engineering.

The RSIC Curriculum Framework does not specify the way in which component topics might be formed into modules or courses. Component topics might be focused in one or two courses, or spread among several courses, along with other non-RSIC topics. The curriculum framework includes more detailed specifications of each component: prerequisite knowledge, component learning objectives, information about required facilities and equipment, and guidelines and suggestions for course design and delivery. The RSIC curriculum framework also makes recommendations about non-RSIC courses or units that should be part of a RSIC program, as prerequisite courses or to supplement the components as part of a full degree program.

3.3. Program Implementation

The implementation of the international education program poses significant challenges. It should be understood that it is impossible to simply send students to several institutions in order to provide them enough education that would lead to recognizable diploma. To allow international form of study, especially in the case that both EU and US universities are involved, some mechanisms need to be devised. These were examined in the frame of ILERT project. The key aspects that need to be addressed are as follows:

- Identification of competencies of individual partners in curriculum delivery content. The course set that form RSIC curriculum was identified and the contribution of individual institutions to the curriculum was defined [8, 9], e.g. ERAU has string educational record in software engineering, while control systems are in proficiency of AGH. There are courses on basic level and advanced level. It is assumed that basic level courses are available at all sites, while advanced level courses are those delivered by institution that is in the position of expert in this field.
- Mobility requires not only to define the process of sending a student for spending semester or two abroad, but also to evaluate the stay, select the candidates, help them to choose the proper set of courses, and also, to find appropriate funds to cover the stay. Currently running project DeSIRE[2] is oriented to practice the student mobility, thus, identify the obstacles and find the way to make this approach sustainable for further supporting the idea of international RSIC study program.
- The methodology for credit transfer, which is the basic instrument for evaluating and assessing the students. As different countries use different systems to count student workload these needs to be unified and conversion functions needs to be given. A comparison and solution to this issue is given in [9].

Table 1. RSIC Components

Software Engineering
Software engineering concepts and practices, software lifecycle models, project management, software processes, software modeling and formal representation; software requirements; software architectural and module design; software construction methods and practices, testing and quality assurance; software maintenance; and notations and tools.
Digital Systems
Concepts, methods, techniques, and tools used to support the design of combinational and sequential digital circuits and the design of fault tolerant and advanced networked hardware components.
Computer Control
Concepts of feedback control, time and frequency domains, continuous and discrete models of dynamical systems, state analysis, stability, controllability and observability, controller design, implementing control algorithms in real-time, integrated control design and implementation, use of analysis and design tools.
Real-Time Systems
Timing and dependability properties of software intensive systems, RTOS concepts and applications, concurrency, synchronization and communication, scheduling, reliability and safety, etc.
Networking
Data communication, network topology, analysis and design, information security, algorithms, encryption, bus architectures, wireless, etc. distributed control and monitoring
System Engineering
System engineering concepts, principles, and practices; system engineering processes (technical and management); system requirements, system design, system integration, and system testing; special emphasis on the development of a RSIC system, the integration of RSIC elements.

To gain at least preliminary experience with the potential issues related to full curriculum implementation we conducted a joint-project that included all participating institution. The importance of the intensive communication was identified during this activity. Regardless the complexity of the problem solved by dispersed group of students, the communication is crucial to the project and makes it either success or turns it into the trouble. This, of course, imposes higher load on faculty involved in international education program as it requires additional flexibility and endurance in solving the emerging problems. For further information on results, please consult [10].

4. Comparison and Conclusions

Comparing the IEEE ECBS and ATLANTIS ILERT + DeSIRE[2], we can sum up that while both international initiatives have been focused to the very close engineering areas (RSIC systems appear a sub-domain of CBS), some differences should be outlined:

- ECBS was provided in frame of the IEEE CS Technical Committee and its implementation required additional activities, while ATLANTIS is run by universities that (hopefully) would utilize the developed educational programs.
- ECBS was aimed at multiple single country implementations, while ATLANTIS aims at transatlantic international implementations.
- ECBS considered limited teacher mobility, while ATLANTIS is based on both teachers and students' mobility.
- ATLANTIS aims at utilizing education capacity not limited to a single institution but virtually available anywhere in the world. This may

increasing the potential but requires dealing with organization and administration issues intensively.

Although elevating the program to the international level includes several obstacles from organizational, administrational and mainly bureaucratic point of view, the gain of knowledge sharing, students' enhanced experience acquired during mobility periods encourage to prepare and implement the international specifically oriented program in all participating institutions.

Having nearly one year of additional experience with DeSIRE^2 mobility project, we can provide some concluding observation:

Mobility. Although the possibility of overseas mobility might seem to be attractive among students, the reality indicates that finding a capable student who wishes to stay semester abroad is relatively hard. This is more problem in the case of US students who in many cases have relatively high tuition fees and are afraid of the quality and their personal revenue from hosting university in Europe. On the other hand, neither students in Czech Republic nor France are so engaged in the mobility possibility. Only Poland partner has more students for overseas mobility than can be accepted.

Distance Learning. The necessity of distance learning appears to be a need if the program should be extended among large number of students. It is mainly because of the cost of mobility. Although mobility should still be an integral part of the international program it can be reduced and offered only to exceptional students. Very successful tool in such program is to implement a joint project, which draws the students in the active participation on collaborative tasks, which increase their confidence in communication skills and additional technical capabilities.

Multidisciplinarity. As evident from the analysis of the partners' curricula [7], the study programs span the broad area of CS and CE. Even if two partners provide courses in the same area, the approach is different and there is not exactly the same platform shared. It seems to be not a problem as students exposed to different methods can be better prepared to gain capabilities in acquiring technical skills in their professional life, during which they are expected to learn new technologies and methods. However we recognized that these courses are compatible in the sense of intended learning outcomes and goals.

The experiences described in this paper reveal the main obstacles and identify the main blocks to build the RSIC international study program. Although at this moment no truly international curriculum implemented to such extend among several participating institution exists, it is generally accepted that joint educational programs integrating intercultural and intersystem differences contribute to new demands of RSIC industry facing globalized economics and help to increase the position of participating institutions on the educational market. Both this observations encourage the further activities towards the full implementation.

5. Acknowledgements

This research has been supported by the EC and the US Department of Education through the ATLANTIS program through EC grant: 2006-4563/006 001 and US grant: P116J060005, and through the Atlantis DeSIRE^2 Excellence in Mobility program, funded by the EC under EACEA Agency Call No. EACEA/02/2008 and by the US Department of Education FIPSE Program under Call CFDA 84.116J.

References

[1] Sveda, M., Dvorak, V., Hruska, T., Drabek, V.: "ECBS Master Degree Study at the Technical University of Brno", *Proceedings of the IEEE Conference and Workshop ECBS'98*, Maale Hachamisha, Israel, IEEE CS Press 1998, pp. 306-312.

[2] Lavi, J.Z., Manion, M., Melhart, B., Pyle, I., Sveda, M.: "Engineering of Computer-Based Systems--A Proposed Curriculum for a Degree Program at Bachelor Level", *Proceedings of the IEEE Conference and Workshop ECBS'98*, Maale Hachamisha, Israel, IEEE CS Press 1998, pp. 369-376.

[3] Lavi, J.Z., Gonzales, R.M., Manion, M., Sveda, M.: "Engineering of Computer-Based Systems Enhancement Courses--Proposed Course Outlines, v.2", *Proceedings of the IEEE Conference and Workshop ECBS'99*, Nashville, TN, USA, IEEE Computer Society Press 1999, pp.336-343.

[4] Kornecki, A., Hilburn, T., Grega, W., Sveda, M., Thiriet, J-M., "ILERT – International Learning Environment for Real-Time Software Intensive Control Systems", *Proceedings of RTS'07 - International Conference on Computer Science and Information Technology*, Wisla, Poland, Oct 2007, pp. 943-948.

[5] Pilat, A., Kornecki, A., Thiriet, J-M., Grega, W., Sveda, M., "Industry Feedback on Skills and Knowledge in Real-Time Software Engineering." *19th EAEEIE Annual Conference*, Tallinn, Estonia, June 29 - July 2, 2008 (4 pp., CD-Edition).

[6] Hilburn, T.B., Thiriet, J.-M., Kornecki, A.J., Grega W., Sveda, M.: "Credits and Accreditation in the U.S.A. and Europe: Towards a Framework for Transnational Engineering Degrees." In: *Innovations*, iNEER, Arlington, VA, US, 2008, pp. 29-42,

[7] Thiriet, J.-M., Hilburn, T.B., Kornecki, A.J., Grega, W., Sveda, M.: "Implementation for a US-European Transnational Degree in Real-Time Software Engineering." *Proceedings of the 19th EAEEIE Annual Conference*, Tallin, Estonia, June 29 - July 2, 2008 (6 pp., CD-Edition).

[8] Hilburn, T.B., Kornecki, A.J., Thiriet, J.-M., Grega W., Sveda, M.: "An RSIC-SE2004 Curriculum Framework." *Proceedings IMCSIT*, Wisla, Poland, 2008, pp. 633-638,

[9] Kornecki, A., Hilburn, T., Grega, W., Thiriet, J-M., Sveda, M., "A Common US-Europe Curriculum: an Approach for Real-Time Software Intensive Systems", *Proceedings of the 2008 ASEE Annual Conference & Exposition*, Pittsburgh, PA, Jun 22-25, 2008, pp. 1359-1371.

[10]Grega W., Pilat A., (2008), *Real-time Control Teaching Using LEGO® MINDSTORMS® NXT Robot*, Proceedings of RTS'08 - International Conference on Computer Science and Information Technology, Wisla, Poland (ISSN 1896-7094, CD-Edition)