Designing and Implementing International RSIC Engineering Curriculum

Ondrej Rysavy*, Miroslav Sveda, Andrew J.Kornecki, Thomas B.Hilburn,Wojciech Grega, Adam Pilat, Jean-Marc Thiriet

*Vysoké učení technické v Brně, Fakulta informačních technologií v Brně
Božetěchova 2, 612 66 Brno
rysavy@fit.vutbr.cz

ECBS-EERC 2009, Novi Sad, Serbia

September 8th, 2009

Outline

- ATLANTIS Programme
- Motivation
- Project ILERT
  - Industry survey
  - RSIC Objectives/Outcomes
  - Workload
  - Curriculum Framework
- Project DeSIRE
- Conclusions

ATLANTIS Programme

- Started in 1995
- In support of analysis, design, implementation, administration, and assessment of international curricula
  - 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.
- For more information see program web pages:

  http://ec.europa.eu/education/programmes/eu-usa/index_en.html

Motivation

Global

- Match the needs of global market
- Gain advantages of established transatlantic partnership
- Support EU-US relations

Institutional

- Cooperation with established and recognized institutions (e.g. ERAU, Grenoble)
- Improvement of courses offered to foreign students
- Student exchange (to US, which is otherwise difficult because of scholarship)

ATLANTIS Scope

- **Transatlantic Degree** action providing support to multilateral partnerships of EU and US institutions for the purpose of setting up joint study programmes - including joint/double degrees - and transatlantic mobility of students and faculty;
- **Excellence Mobility Projects**, providing follow-up financial support for student mobility to joint consortia that have a proven track record of excellence in transatlantic cooperation;
- **Policy-oriented measures**, addressing comparative higher education and vocational training issues, and promoting dialogue on recognition of qualifications and accreditations;

ILERT Objectives (2006-2008)

The main objective of the ILERT project was to

- Create a methodology for design and implementation of transatlantic, multidisciplinary engineering program
- Stimulate students to follow careers in the area of real-time dependable safety-critical control systems and expose them to opportunities of international collaboration
- Encourage the exchange of staff and students
- Offer multidisciplinary and multicultural experiences
- Foster technological research base and develop educational research

The ILERT Team

- Four universities
  - Prof. Grega from AGH University of Science & Technology, Poland
  - Prof. Thiriet from Grenoble Université, France
  - Prof. Sveda from Brno University of Technology, Czech Republic
  - Prof. Kornecki from Embry-Riddle Aeronautical University, USA
- Common area of interest
  - Real-Time Software Intensive Control Systems (RSIC)
- Different experiences and expertise
  - Real-Time Systems
  - Computer Control
  - Software engineering
  - Digital Systems
  - Communication and Networking

GRENOBLE UNIVERSITÉS

EMBRY-RIDDLE AERONAUTICAL UNIVERSITY

ILERT Approach

Several activities were accomplished:

1. Analysis of industry requirements on graduates in the RSIC domain
2. Identification of the RSIC learning objectives and student outcomes
3. Research common areas in the four curricula in the RSIC domain
4. Research the potential laboratory infrastructure
5. Analysis of European Credit Transfer System (ECTS) and the mechanism of credit transfer at US Colleges and Universities
6. Analysis of the required assessment and accreditation activities
7. Creation of RSIC curriculum framework

- Part A - General Skills and Attitudes (10 items) the issue was to find importance of general skills, capabilities and attitudes of engineering school graduates when they enter the job market
- Part B - Technical Knowledge Areas (15 items) included items related to specific technical areas and skills
- The responses could be selected as: Essential, Important, Unrelated, and Unimportant
- Example:

How important are the following capabilities when hiring engineering school graduates?

1. Work as a part of a multidisciplinary team
2. Analyze, understand and define the problem

Scope of industry survey

- Participants fill in survey on-line.
- 370 companies were asked, 43 responses were received (11% response rate)

Country	Company Name
USA (12 responses)	Avidyne, Raytheon, Hawker Beechcraft Corporation, Stuart W. Law Company, Boston Scientific, Teledyne Controls, Boeing, Honeywell Aerospace, Hamilton Sundstrand
Poland (14 responses)	CSN-STANEL Automatyka, ABB Corporate Research, Astor, Abis, RAControls, InTeCoFEV Polska, Pumpa, Tarbonus, Multiprojekt, Computer Systems for Industry, ComArch, INVENTIA, MPL Technology
Czech Republic (10 responses)	Tescan, ANF DATA, B+R Automatizace, Honeywell, Freescale Czech Republic, ADC Czech Republic, CAMEA, Flextronics Design, ANeT Ltd., Schneider Electric CZ
France (7 responses)	CIRTEM, National Research and Safety Institute, ST Microelectronics, IRSN Radioprotection and Nuclear Safety French TSO, Leroy Somer, Airbus S.A., Euro-Systems

A. General Skills

most important

A1 - Work as a part of a multidisciplinary team

A2 - Analyse, understand and define the problem

A3 - Think independently and search for solutions

A10 - International social, cultural and political issues

A8 - Lead a team

least important

B. Technical Skills

most important

B6 - Software design and development concepts, methods and tools

B4 - System specification and design methods

B10 - Software program construction

B7 - formal methods applied to system development

B2 - specific application domain familiarity

least important

Some other observations

- It is interesting to note that USA companies are in favor of multidisciplinary, international teams, while for Europeans the issue does not seem to be critical
- European countries seem to weight more in problems solving skills.
- The mathematical knowledge is required - the analytical methods are implemented for the software algorithms

RSIC Objectives

- General program objectives

[A] Demonstrate professionalism in their work and grow professionally through continued learning and involvement in professional activities.
[B] Contribute to society by behaving ethically and responsibly, and by incorporating knowledge of history and culture into one's professional decisions.
[C] Communicate effectively in oral, written, and newly developing modes and media.
[D] Assume a variety of roles in teams of diverse membership.

- Specific RSIC program objectives

[E] Demonstrate understanding of analysis and design as applied to modern software-intensive control systems
[F] Demonstrate understanding of processes and techniques and the role of modern engineering tools necessary to engineering practice as applied for creation of software-intensive systems
[G] Demonstrate understanding of quality assurance and hardware/software integration for creation of safe and dependable systems

RSIC Outcomes

The program outcomes help to identify the topics and activities necessary for preliminary curriculum design.

1. An ability to apply knowledge of mathematics, science, and engineering to solve technical problems
2. An ability to design and conduct experiments, and an ability to analyze the data
3. An ability to analyze and understand the operation of a control system or component to meet desired needs (feedback, stability, system dynamics, robustness)
4. An ability to apply advanced software engineering techniques to implement real-time concepts (timing, scheduling, concurrency, synchronization)
5. An ability to support assurance of the quality of a software-intensive system across its life-cycle including assurance of its dependability using established standards and guidelines (verification, validation, testing, safety, reliability, security, standards, guidelines)
6. An ability to integrate hardware and software on variety of platforms with various interfaces and protocols
7. An ability to use a defined lifecycle process in development of software-intensive system
8. An ability to function on multi-disciplinary teams

Partner's expertise areas

- Based on RSIC outcomes, details of specific areas were supplied, e.g.:

1. Software Engineering
   - software engineering concepts and practices, software lifecycle models, project management, software processes, software construction methods and practices, software modeling and formal representation; software requirements; software architectural and module design; testing and quality assurance; software maintenance; and notations and tools.

- Next, courses were selected to match the specific areas, e.g.:

AREA	ERAU	AGH	BUT	LAG
Software Engineering	o SE300 (E) Software Engineering o SE310 (E) Analysis and Design of Software Systems o SE320 (E) Software Construction o SE420 (E) Software Quality Assurance	o TSD (P) Technologies for software design o TIS (P) Testing and implementation of software	o IUS (E) Introduction to Software Engineering o MPR (E) Project Management	o GE-L-UE1.4 (F) Embedded computer project management

Workload

- Europe ECTS suggests workload for the entire degree programs:

school	# of semesters	# of weeks	total hours	hours/week	ECTS/semester
ERAU	8	16	1900-2200	~16	N/A
AGH	7	15	2400-2900	~25	30
BUT	6	13	2300-2450	~30	30
LAG	6	15	2500-2700	~29	30

- Approximate data from a typical curriculum sample:

	General Education	Math and Science	RSIC	TOTAL
ERAU	384	512	1152	2 048
AGH	315	555	2080	2 850
BUT	208	312	1885	2 405
LAG	252	672	1596	2 520
average	**290**	**513**	**1679**	**2481**
distribution	***12%***	***21%***	***67 %***	***100%***

RSIC Curriculum Framework

- The proposed 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.
- In order to verify the practicality and efficacy of the RSIC Curriculum Framework, each ILERT partner analyzed how the Framework could be applied to their program.

RSIC Curriculum Component Specification

- Components
  - Software engineering
  - Digital systems
  - Computer control
  - Real-Time Systems
  - Networking
  - System engineering
- Component specification:
  - Description
  - Prerequisites
  - Learning outcomes
  - Facilities and equipment
  - Guidelines and suggestions

Component	*Software Engineering*
Description	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.
Prerequisite Knowledge	Ability to design, implement and test small programs (100 lines of code), written in a commonly used high-level programming language.
Learning Outcomes	Upon completion of this component, students should be able to • Describe the major problems in the development of a large, complex software system. • Describe and discuss issues, principles, methods and technology associated with software engineering theory and practices (e.g., planning, requirements analysis, design, coding, testing, quality assurance, risk assessment, and configuration management). • Describe issues, principles, methods and technology associated with the use of formal modeling in software engineering. • Describe, discuss, and apply the commonly accepted principles of software quality assurance (reviews, inspections and testing). • Apply requirements engineering principles of elicitation, analysis, and modeling to the development of a requirements specification. • Describe and analyze different software architectures views and styles. • Describe and discuss the structured and object-oriented design methodologies. • Describe and discuss the principles, methods and practices of software evolution. • Show capability with various software engineering tools used for formal software modeling, requirements engineering and software design. • Working as part of a team, use a defined software development process to develop a high-quality modest sized software product (1000 lines of code).

Project DeSIRE (2008-2012)

Bilateral agreements between EU and US universities

Similar conditions to ERASMUS program

Mobility parameters:

- 5000 EUR for EU student to partly cover costs
- 5000 USD for US student to partly cover costs
- 7 students from each university
- One full semester, for graduate program

Cost of living in Florida is TOTAL \$5,345 – \$5,745

Almost two years of experience

Not as successful as expected:

EU -> US

- Rather low interest between students (except Poland)
- Difficult to go through the paper work
  - EU students need special F-1 Student Visa
- Inconsistent admittance procedure at US side

US -> EU

- Minimal interest to study in (continental) Europe
- Rather low number of courses offered in English
- Difficulty to adapt for US students in EU

Corrective actions

Increase attractiveness of partners

- Extend the list of courses outside the RSIC area
- Active promotion by visiting professors
- Invitation of local industry partners to participate (e.g. by providing internship for incoming students)

Impose additional advantages

- Reduce tuition fee during the mobility period
- Allow to remotely study courses at home institution during a mobility period

Relax admittance procedure

Conclusions – Possibilities

- International academic mobility is seen by governments as a part of global competitiveness
- Transatlantic mobility provides a new international and comparative academic experience for students, develops their intercultural skills and improves their opportunities for employment
- Mobility students are able to focus on an area of concentration not available at home institution, due to the opportunity to take classes, get engaged with host institution faculty, to host institution unique laboratories
- Home institution students interacting in the classrooms with the mobility students gain better understanding and appreciation of international and global aspects of modern world
- Movement of people, ideas, and culture across Atlantic present new opportunities to promote culture and contribute to understanding between the nations

Conclusions – Difficulties

- Relatively easy to define a curriculum
- Harder to find the way to overcome the national and local policy and regulations
- Difficult to establish truly international programme
- Interestingly, it is also difficult to find enough well motivated students for short time mobility

ATLANTIS ILERT Home Page

http://www.ilert.agh.edu.pl/