

# Cyber-Physical Systems: Educational Problems and Solutions

Ivan E. Ivanov, Vesselin E. Gueorguiev, Desislava V. Georgieva

**Abstract—** The development of qualified multidisciplinary-educated engineers is a complicated task. At the same moments this is a very responsive task. Most of the current social life has for a background the different engineering solutions, support and work. Education of engineers for a today needs is under increasing pressure for non-breakable adaptation to new technical and technological realities. Designing and implementing cyber-physical systems is modern and preferable, and there are work positions in many companies – so students and engineers are interested in it. In the last years the new trend to Internet-of-Things also increased this interest. Unfortunately many of the newly graduated engineers start to look for position change after several months of work or at least for extra courses. Now we have a lot of web programmers and many control engineers but there is a lack of people working in the area of the design and implementation of the cyber-physical systems (or as they were known before real-time embedded systems).

**Index Terms —** cyber-physical systems, education, interdisciplinary approach, problem-based learning.

## I. INTRODUCTION

We want to start with the statement that we won't offer any final solution here in this paper. We focus on some very problematic points in the now-a-day education problems, industry and research requirements and starting level of the engineers oriented to real-time embedded (or now cyber-physical) systems and one possible solution for some of the mentioned tasks.

The engineering education is in a revival for all the last decade. Engineering departments are re-opened, the number of bachelor and master-degree students is increasing, e.g. Br. Yoder [14][15] and A. Scott [16]. Together with this the complexity of engineering sciences is increased either. This follows from the complexity of technologies, the increasing level of multi-disciplinary solutions, the ubiquitous computing and networking and Internet-of-Things (IoT).

We started to focus on the problems in the real-time systems implementation in the industry many years ago. After year 2000 we thought that our industry partners have not understood well specifics of this area: what and how students should be educated if he/she has to meet industry needs after graduating. But after the discussion at several Academic Days and educational symposia we understood that this is not only a problem in our country but in the ICT world.

Talking about cyber-physical systems we understand here mainly embedded control systems, mobile real-time systems and industrial control systems. Other real-time systems will not be discussed here.

- Interdisciplinary problems are envisaged many times and one good explanation of them for the area of embedded systems as is stated by E. Lee [1]: *“Embedded software is harder to design. Embedded software often encapsulates domain expertise... and very small programs may contain highly sophisticated algorithms. Existing software design techniques aren't suitable... Embedded software is often designed by engineers who are classically trained in the specific domain but they have little background in the theory of computation, concurrency, object-oriented design, operating systems, and semantics.”*
- Integration of the computers in the current world is emerging. In all cases we have to think about computer-oriented education for all engineering sciences. Immediately after acceptance of this point of view the definition of “software engineering” and attributes of “good software” in Sommerville [2] has to be envisaged: *“Software engineering is an engineering discipline that is concerned with all aspects of software production”*.

What are the attributes of good software [2]?

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable.
  - Maintainability - Software must evolve to meet changing needs;
  - Dependability - Software must be trustworthy;
  - Efficiency - Software should not make wasteful use of system resources.
- Acceptability - Software must be accepted by the users for which it was designed. This means it must be understandable, usable and compatible with other systems.
- Heterogeneity - Developing techniques for building software that can cope with heterogeneous platforms and execution environments;
- Delivery - Developing techniques that lead to faster delivery of software;
- Trust - Developing techniques that demonstrate that software can be trusted by its users.

If one changes the word software with the word product we will receive almost the definition of a good product.

And we agree with L. Feisel and A. Rosa, that *“Engineering is a practicing profession, a profession devoted to harnessing and modifying the three fundamental resources that humankind has available for the creation of all technology: energy, materials, and information. The overall goal of engineering education is to prepare students to*

*practice engineering and, in particular, to deal with the forces and materials of nature.*" [3]

Following these basics and our experience we agree with the society that the right solution is based on new approach for computer-based engineering education, because the industry needs are (e.g. in I.E. Ivanov et al. [4]):

- Combined elements of engineering sciences
- Broad systems view
- Business knowledge
- New behavioral skills
- Increased mobility between academia and industry
- New understanding of environmental changes.

How can be explained current situation from our point-of-view:

- Problems with "pure" sciences
- Problems with current technologies implementation
- Validation/verification problems solutions
- Education in laboratory environments without many real limitations.

One of the ways to solve these problems is the so-called "problem-based learning" approach.

The rest of the paper has the following structure: Part 2 contains an analysis some current problems; Part 3 presents on implementation of the "Problem-based learning" approach for cyber-physical systems education. Part 4 is the conclusion.

## II. ANALYSIS OF CURRENT PROBLEMS

### A. Education in "pure" sciences

Many of the "old" engineering directions like mechanical, electrical, power production, and newer ones (i.e., microelectronics, bioinformatics, software in all aspects) are studied today more or less self-explaining. Looking in the industry and IoT we see that their clear boundaries are no more available. All they are highly computerized and cross dependable, see E. A. Lee [5].

- Integrated requirements analyses

One of the biggest problems in current educational schemes is the problem with integrated requirements analysis. Mostly tutors and students are focusing on only one or two requirements. Unfortunately usually this is not the most important requirement. This is classical problem in the Software Engineering. For embedded software is said by E. [6]: *"Embedded software is that which resides in devices that are not first-and-foremost computers... Embedded software engages the physical world, interacting directly with sensors and actuators."* The same author said in 2008 [8]: *"Cyber-Physical Systems (CPS) are integrations of computation with physical processes. In the physical world, the passage of time is inexorable and concurrency is intrinsic. Neither of these properties is present in today's computing and networking abstractions"*. There is still lack of clear solutions how to educate students to understand these integrated requirements.

- Integrated design of software and hardware

The understanding of the developing process of a CPS is very important: CPS involve both significant hardware and

software design. The problem is that only limited number of students from non-hardware profiles is able to understand how the modern HW works in details, to control it directly and to have common language with HW designers. One of the reasons for this situation is that more and more of the design tools used in studying process are abstract tools. In many cases an 8-bit embedded controller with 2 Kbytes ROM/flash and 512 bytes of RAM is in class "impossible". Even students understand formally hardware and performance limitations without a touch to very limited systems they do not feel it.

The introduction of the new term: "Cyber-Physical Systems" at the boundary of years 2006-2007 by Lee E.A. and the related group from the University of California, Berkeley, was an attempt to describe the new type of interdisciplinary systems covering from very abstract mathematical models and descriptions to physical connection, wires, actuators and sensors.

- Control theory

From the today's point of view the Control theory is part of the ICT. This follows from the facts that: massively the control is implemented using computers; a special part of the mathematics – the computational mathematics was developed to cover the increasing needs of more and more precise and fast calculations. But Control theory in all its aspects is massively based on mathematics. And it needs understanding of Physics and many other sciences.

Unfortunately our observations is that students (and engineers) understanding well control theory do not understand so good computational aspects (except the specific computational mathematics), Real-Time Operating Systems (RT OS), programming languages and similar. The opposite is also true: computer-familiar students are far from control theory in general.

We shall omit the "real-time communication" in this discussion because in general it is usually known only to people working in appropriate companies or students participating in CPS research groups.

The combination of control, RT OSes and real-time communications is simply problematic for the most of the students and young engineers.

### B. Modern technologies and their implementation

Today dominating element in companies' politics is "time-market time". This is understandable because the competition is tremendous. Versatile abstract environments for system design are becoming more and more implemented and used. And this is good because they are based on formal models and they save a lot of formal errors.

- Integrated environments and wizards

The problem here is in the fact that students do not learn how to produce optimized by size or performance systems. Here is the place to show the revealing observation from Andrew Tannebaum written in all versions of his famous book "Modern Operating Systems" about the possibly of the old OSes to handle a lot of tasks on very low-productive computers - *"This is not quite as crazy as it sounds, since in those days people knew how to write small, efficient programs, a skill that has subsequently been completely lost."*

The use of wizards and some types of program generators leads to the loss of program efficiency. In the case of real-time systems this can be compensated only by growth of the processor(s) performance but this needs more energy. Finally students are not able in most cases to optimize deeply characteristics of their systems.

Here we do not want to say that the use of program generators is not good in general (both authors are co-authors of a family of program generators for control system development). The problem is their use in the process of early education because they abstract students from the real implementation and understanding of their systems. The next problem is to decide how deep students need to know what is generated by the wizards and how some technologies and solutions are implemented in the supporting libraries.

The Object-oriented (OO) approach is one of the pillars of today's ICT. There is no doubt about its place in the computer education. The problems start when the "programming" goes to the "real-time programming". As we see debates in the ICT community how students should study "programming" and what is the best first programming language will not stop ever.

The problem with the OO approach is that it always is shown only with some OO language. The language grabs the attention of the students and understanding of the OO as an approach gets in the background.

What happens if a designer or programmer has to use the OO approach without OO language support? In the many international and national scientific and industrial projects in the last two decades we found that even experienced programmers are not able to move from the starting point without OO language or in the situation of removal of most of the usual OO features (like C++ conforming the MISRA standards).

The problem is in the way of thinking. This is an example where the instruments are substituting the technology. Here we will go beyond the discussion of the problems only but want to mention some solutions proposed years ago.

In 1994 in his fundamental paper J. Udell said [10]: "*Object orientation has failed but component software is succeeding*".

Several years later C. Szyperski [11] repeated: "*Component-based development emerged from the failure of object-oriented development to support effective reuse. Single object classes are too detailed and specific.*"

Unfortunately even now many engineers continue to understand 'component' as 'object' (like C++ or Java objects).

We think that there is a need for more courses, oriented to component-based systems design and implementation.

- Validation, verification and testing

Functionality validation is a very important part of the design process. An example from a diploma work is about a complicated test suit for mechanical and lubricant tests, exploiting a highly precise temperature control system. Only after implementation in metal and control system design and implementation was found that the cooling subsystem is unusable and temperature control is impossible. This followed from the lack of formal functionality validation and formal proving of

all closed loop functionality. Preparing this paper the authors searched the net for university validation and verification courses. We found no courses discussing deeply the real-time software aspects. Again except the specialized courses for computer departments we were not able to find teaching materials for static and dynamic source analyses and how to analyze real-time systems.

Testing is the other problem. Real-time testing is very different from the "standard" testing techniques and it is not well presented in the university courses.

### 3. PROBLEM-BASED LEARNING APPROACH

The old classical approaches still have their place in the education process but there are many new ways to develop flexible, adaptive and well educated engineers. One of these ways is the problem-based learning approach. Just here we want to point that some of the key problems mentioned before are not covered by the following text.

The Problem-based learning (PBL) is an approach engaging students with solving a real problem. It shifts the process from teaching to learning. People are the key element in the education process but their attention capacities are limited. It is necessary to make an abstraction process, keeping apart the irrelevant parts of problems solved and focusing on the most important things. Elements of this process are discussed by M. J. Albaladejo et al. [12].

The example course we will present here is named "Simulation of industrial objects" and its goals are to present how to design and implement Hardware-In-the-Loop simulators together with the control systems. The scope does not include identification and mathematical simulation. It targets discrete and heterogeneous objects with known internal structure and mathematical models. The focus is on the implementation of the real-time simulators, their connections to the control systems. The hardware basis is oriented to single-board controllers (SBC) with extensive industrial periphery. The software background requires skills in programming in C and possibilities to implement tasks for a Real-Time operating system.

The targeted objects are as follows: 1) a liquid tank and its control system – continuous object; 2) a concrete plant – hybrid object. This combination of objects and control needs is very wide and requires deep understanding and complex solutions.

The tank model is based on a known model because the task is the simulation and not identification. The controller is synthesized also. The real tanks (and their laboratory models) have several sensors for external and internal temperature, liquid density, height of the liquid in the tanks, the pump flow rate meter, valve position sensors. Internal temperature is measured using distributed (multipoint) thermometer measuring temperature in several points on different depths in parallel. The density meter and level (height) of the liquid are measured using sensors from fuel tank systems. The tank models include simulators of all real sensors. The controller is based on the same single-board controller used for tank simulation. The simulator and the controller are connected using real physical signal generation/transfer.

The concrete plant is a complex hybrid object. The object

is based on a simplified version of a real plant. It has number of silos with elevators, tanks, mixers and transport pipe. The most complicated element is the design of a distributed object simulator (multi-computer) and two-layer control system. Each sub-system of the plant has its own controller connected to the upper level controller. The design is based on component approach. Each sub-system is a separate computer component communicating with the neighbor subsystems via physical signals or via industrial communication network.

At this place the need for mixing traditional and problem-based learning becomes clear. Students need to have some basic knowledge about control systems, operating systems and programming.

The target was to create a working environment for the students similar to the real environment in their future work:

- to guarantee fundamental knowledge of industrial computer control systems as basis for the development of further developments;
- to accustom students to work in teams when solving industrial problems;
- to encourage students to use practical skills to improve their problem solving abilities;
- to develop the ability to adapt to any new computer based systems;
- to study how to integrate both stand-alone and distributed computer systems working real-time using specific tools and pre-designed validated and verified building blocks.

The course and all education process are organised around the following scheme: lectures, laboratory sessions, seminars and micro-projects.

- Lectures: lecturer presents main ideas of lecture contents and proposes some application problems which students solve individually. This is mainly done by a presentation, interrupted by questionnaires and discussion groups.
- Laboratory session: Students solve a practical problem previously presented during lecture. Students work in small teams. The lecturer gives some advices.
- Final discussion: Each student team presents its methods, challenges and solutions of the laboratory session.
- Seminars: a panel discussion with student teams.
- Mini-project: dedicated to planning, design and development of elements of the control system or the simulator. The mini-project is performed by teams of 4 students. Weekly, the mini-project is advanced progressively.

The sequence of problems to be solved includes the next elements: 1) generation of an interrupt-driven system for analog and discrete signals acquisition and generation; 2) implementation of the same data acquisition and signal generation using an open-source real-time operating system; 3) implementation of a closed-loop system based on two communicating to each other SBC – one for the controller and one for the object. This is the first introduction of a hardware-in-the-loop simulation. All implementation is simplified to avoid focusing on control and to target the implementation of the interacting computers.

Where the problems and what are the didactic results:

- Students are oriented not only to simulate the flow of the liquid through the tank and to implement the height control algorithms but to transfer the calculations from the high-precision calculation environment (the MATLAB) to the middle or low-precision environment (the single-board controller).
- Students have to embed the operation of industrial peripheral devices receiving and transmitting physical signals to the implemented mathematical simulator.
- Students have to implement sub-models of the control devices (mainly the pump) which are connected via communication lines to the controller and to run real communication. (The communication protocol program comes ready-to-use).
- The controller is build using the structure designed using MATLAB but implemented and tuned using the laboratory control systems program generator.
- Both the object simulator and the controller are implemented using one and the same single board 32-bit ARM based controller.
- The controller and the simulator are connected through an additional communication line to control the simulation time flow. The time-flow control possibility is included in the simulation and control algorithms.
- For the concrete plant object the new element is the design of a distributed object simulator (multi-computer) and two-layer control system. Each sub-system of the plant has its own controller connected to the upper level controller. Using the pre-defined or pre-designed models of the sub-systems students are composing a complex simulator. The design is based on component approach.

The presented course and learning approach is included as a selectable bachelor course in the Faculty of Automatics at the TU Sofia. It was designed under a TEMPUS project and presented for use to the project partners.

The analysis of the experimental run of the course show that the PBL approach is very promising but needs more basic links to the fundamental courses.

#### 4. CONCLUSIONS

Education in Cyber-Physical Systems found many new problems in current university courses and now-a-day educational methodologies. The severe gap has developed as a result of:

- Exponential growth of ICT
- Growing use of Internet-of-Things-enabled devices and services
- Organisational structure and working methods in the industry
- Globalisation and labour mobility
- Inability of education system to meet demand in sufficient quantity.

The problem-based learning shows that the shift from the more conservative approaches to it is not easy and both students and teachers have to adapt them to this educational approach. The feedback from the students experimented the

course draft is that this is a better way than the classical lectures-exercises couple which is based on a loosely coupled stream of lectures and exercises. But the results show also that even using many abstractions and libraries of pre-defined and pre-designed elements and blocks the amount of information is big and overloads students.

#### REFERENCES

- [1] E. A. Lee, What's Ahead for Embedded Software? IEEE Computer, 2000 - 9, pp. 18-26
- [2] I. Sommerville, Software Engineering, 8th Edition, Addison-Wesley, 2007, ISBN 13: 978-0-321-31379-9
- [3] L.D. Feisel, A. J. Rosa The Role of the Laboratory in Undergraduate Engineering Education, Journal of Engineering Education, January 2005, pp. 121-130
- [4] I. E. Ivanov, V. Gueorguiev, D. Georgieva, Programming for Embedded Devices: Algorithms + Data Structures  $\neq$  program, 2nd Annual International Workshop "Computer Science and Education in Computer Science" 2006, Borovec
- [5] E. A. Lee, Are "Embedded Systems" Just Systems Made with Small Computers?, Artist International Collaboration Days, Education Day, In conjunction with EMSOFT, Philadelphia, PA, 2003, <http://ptolemy.eecs.berkeley.edu/presentations/03/emsoftEducationDay10-11-03.pdf>
- [6] E. A. Lee, 1999, <http://ptolemy.eecs.berkeley.edu/papers/99/HMAD/html/title.html>, page 4
- [7] E. A. Lee, S.A Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, 2011, LeeSeshia.org
- [8] E.A. Lee, Cyber Physical Systems: Design Challenges, Technical Report No. UCB/EECS-2008-8, 2008, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html>
- [9] A. Tannenbaum and H. Bos (2008, 2015), Modern operating systems, fourth edition, Prentice-Hall Inc.
- [10] J. Udell, Componentware, Byte, 1995, 19(5), March, pp. 46-56
- [11] C. Szyperski Component Software: Beyond Object-Oriented Software. Reading, MA: ACM/Addison-Wesley, 1998
- [12] MathWorks, Generate and verify embedded code for prototyping or production, <http://www.mathworks.com/embedded-code-generation/>, [last accessed: 21.01.2015]
- [13] M.J. Albaladejo et al., A Hybrid Structured Methodology for Developing Computer-based Industrial Computer Systems International Conference on Frontiers in Education (FECS). Las Vegas (EE.UU.), 2012, pp. 321-327.
- [14] Br. Yoder, Engineering by the Numbers, 2016, <http://www.asee.org/papers-and-publications/publications/college-profiles> [last accessed: 30.05.2016].
- [15] Br. Yoder, Engineering by the Numbers, 2013, <http://www.asee.org/papers-and-publications/publications/college-profiles> [last accessed: 30.05.2016].
- [16] Scott A. (2013) Chemistry Makes A Comeback In U.K. Universities, Schools, Chemical & Engineering News, <http://cen.acs.org/articles/91/i11/Chemistry-Makes-Comeback-UK-Universities.html> [last accessed: 30.05.2016].

**mag. eng. Ivan Evg. Ivanov, PhD**, is an associate-professor in the Faculty of Automatics, Technical University of Sofia, Bulgaria. He starts his academy activities in 1982 as an assistant-professor in TU-Sofia. His research interests include: embedded and mobile systems; program generators; validation and verification of software projects, systems and applications; medical systems and apparatus. He has been participated in many science and R&D projects. He has more than 90 papers and many R&D projects.

**mag. eng. Vesselin Evgueniev Gueorguiev, PhD**, is an assistant-professor in the Faculty of Computer Systems and Control, Technical University of Sofia, Bulgaria and a lecturer in New Bulgarian University of Sofia, Bulgaria. He starts his academy activities in 1992 as a

PhD student in TU-Sofia. His research interests include: embedded and mobile systems; program generators; validation and verification of software projects, systems and applications; computer graphics and animation; stereoscopic visualization; medical informatics. He has more than 80 papers and he has been participated in many research projects.

**mag. arch. Desislava Velcheva Georgieva, PhD.** is an assistant-professor in the Informatics Department, New Bulgarian University – Sofia. Her research interests include: computer graphics, 3D realistic visualization, computer animation, computer aided medical imaging, computer-based medical system, She has been participated in many research projects and she is an author of many science papers and publications