Engineering Experiences in the Design of Advanced Industrial Informatics Systems

Chapters

Using industrial computers to design advanced industrial informatics

J.V. Busquets, J. Albaladejo, A. Perles, J.V. Capella, C. Domínguez, J.M. Martinez, H. Hassan

Universitat Politècnica de València, Valencia, Spain

Abstract: TEMPUS is the European Union (EU) programme that supports the modernisation and reform of Higher Education (HE) and promotes institutional cooperation that involves the EU and EU's neighboring countries (PC). The PC includes Eastern Europe, Central Asia, the Western Balkans and also the Mediterranean region. The Medis Project belongs to the TEMPUS framework. The Medis project that is described in this article develops in this context.

Keywords: EU convergence for Industrial Technology curricula in non-EU countries, Advanced and Distributed Industrial Informatics Systems, Problem Based-Learning methodology.

I.   INTRODUCTION

The Medis project belongs to the European framework TEMPUS project[1]. This involves the development and implementation of a methodology for developing a master of Industrial Technology in order to be able to teach in non-EU Partner Countries (PCs). It is to be implemented using the same credit system of the EU (European Credit Transfer System or ECTS). This system includes three cycles: the Degree, Master, and Doctorate, all in the subject area of Design and Development of Advanced Industrial Informatics Systems. The extra-European universities PCs are: Almaty and Pavlodar in Kazakhstan, Odessa and Kiev in Ukraine, and Petrozavodsk and St. Petersburg in the Russian Federation.

The main objective of Medis project is the design of curricula in PCs for the training of highly skilled engineers in the area of design and development of advanced and distributed industrial systems. To achieve this goal, the Medis project, have linked five European Universities, each specialised in a specific area of computerized industrial systems. Each one will create a specific learning module for designing curricula in PCs. In particular, the USTUTT University at Stuttgart-Deutschland develops the curriculum on *Microcontroller based Systems*. MDU University of Vasteras in Sweden establishes the curriculum from the point of view of the *mobile devices*. The UP College of Porto in Portugal establishes the curricula from the point of view of *Industrial Networks and Field Buses*. The TU College of Sofia in Bulgaria makes the design of the curriculum from the point of view of *Controllers and Simulators*. And the UPV in Valencia-Spain, designs the curricula from the perspective of *Industrial Informatics*.

The work at UPV comprises the design a module of *Industrial Computing* for the curriculum of the PCs using the methodology based on problem solving, *Problem-Based Learning* or "PBL" [1,2,3].

This methodology is a kind of active learning. It is based on problem solving, where students build their knowledge based on problems and situations from real life. Meanwhile, they also exercise in the process of reasoning that will use in their future professional job; evaluating and integrating the acquired knowledge to achieve professional technical expertise in their specialty. The essential feature of the PBL methodology is the use of problems as a starting point for the acquisition of new knowledge, and the concept that the student is the protagonist of their learning progress. With this method, applied to the Industrial Computing, the student is empowered to take the roll of "designer" of an *Industrial Computer Application*, and throughout the course, "designers" attain new knowledge, skills and attitudes as it happens in the day to day in an real project office. Designers must be able to analyze and deal with the proposed design of an industrial computer system, the same way they have to do during their future professional job.

One feature that distinguishes the traditional teaching against the PBL is that, in traditional teaching all the information is exposed first, and then the student will try to apply it to solve a problem. In PBL conversely, the design problem appears first, that motivates and identifies the learning needs. The student looks for the necessary and eventually returns to the problem to refine the requirements and specifications. Another feature of PBL is that allows group work. For this, the students in the theory class are divided into small groups (in our case no more than four each), and it is proposed to all the "designers" to conduct a project (which we call mini-project), as if it were a design contest, as happens in real life.

Furthermore, using this methodology the student acquires other transversal skills: oral presentation, report writing, and project cost estimation.

The Medis project comprises the development and implementation of a methodology for Masters Degrees in the area of Industrial Technology curricula for the PC's. The proposed methodology uses the European Credit Transfer System (ECTS), in the three-cycle system for the recognition of a Degree in the area of Design and Development of Advanced Industrial Informatics Systems. The Medis main objective is to prepare highly qualified engineers for the design and development of advanced and distributed Industrial Informatics Systems. The MEDIS project has been associated five technical universities in the EU. Each of them is specialized in a particular subject, with the aim to create a specific learning module. All modules have the equivalent level to the subjects taught in any Master course of Industrial Informatics in the EU.

The modules will be adapted to these five universities in the PCs: USTUTT (Stuttgart-Deutschland) will use microcontroller based systems. MDU (Vasteras-Sweden) will apply mobile devices. UP (Porto-Portugal) will provide the Industrial networks and Fieldbuses to achieve the control. TU (Sofia-Bulgaria) will design controllers and simulators. Finally, the UPV (Valencia-Spain) in Industrial Informatics.

This paper is structured as follows. Section 1 presents an introduction to Medis project. Section 2 outlines the Medis objectives and the PBL methodology. Section 3 describes the AIISM structure, or learning activities of AIISM: lectures, laboratories, seminars and mini-project. Section 4 details the contents and scheduling in the AIISM. And section 5 summarizes some conclusions.

II. THE MEDIS UPV OBJECTIVES

The Technical University of Valencia (UPV) has a long expertise in the area of Industrial Computer Systems (IBS), not only at Degree level, but also at the level of Master and Doctorate. The *Degree of Computer Engineering and Electronic Engineering and Automation*, the Master of Computer Engineering, and PhD programs *Automation and Industrial Informatics* and *Computer Engineering*.

More precisely, the UPV work consists in adapting the PBL methodology in the curricula of Industrial Technology for the PC's. So, to achieve these objectives in the Medis-UPV project we will undertake the following settings:

- Propose the PBL methodology and develop the resources for teaching an Advanced Industrial Informatics Specialization Module (or AIISM) over an Industrial System. That is, under basic requirements on a small industrial problem, the control of a "Water Tank" (See figure 1). This is a small project, that we call the "Mini-Project". Among others, we prepare the academics resources that consist in: Lectures, Laboratories, Seminars, and the "Water Tank" project or Mini-project.

Figure-1: The Water tank.

- Analyze the curricula of the HE Institutions of the PCs to adapt and integrate the AIISM in the specific study program of each PC.

- Design training courses and perform the formation of teachers, technicians and administrative staff of the PCs.

- Implement the AIISM-PBL methodology in PC's and assists them during its deployment.

- Assess the implementation of the AIISM.

- Disseminate and exploit the results among stakeholders.

The AIISM is conveniently structured with different activities of progressive complexity to facilitate the student-designer teams to develop their projects along the course. The learning sessions will be organized in different activities: lectures, seminars, laboratories and mini-project implementation.

These activities are developed during five hours/day, one day of the week through a PBL methodology, using as a case study or the problem of the control of the "Water tank", (See Figure 1).

The control of the water tank or similar processes will be tackled by different platforms. Each of the other four EU partners has to tackle the control of this "Water Tank" using their corresponding point of view platform.

EU University	Platform
USTUTT (Stuttgart-Deutschland)	Microcontroller based Systems
MDU (Vasteras-Sweden)	Mobile Devices
UP (Porto-Portugal)	Industrial Networks and Fieldbuses
TU (Sofia-Bulgaria)	Controllers and Simulators
UPV (Valencia-Spain)	Industrial Computers

This article only relates to the course structure performed at the UPV, and proposes the PBL methodology on the topic *Industrial Informatics* [1].

III.   THE LEARNING ACTIVITIES IN THE AIISM UPV

The AIISM from the UPV is conveniently structured with different activities using the point of view of the Industrial Computer, with progressive complexity to facilitate the working student teams to develop their projects along the course.

To develop the course, the students have to apply the knowledge acquired from the lectures and the laboratory practices. The proposed learning activities based on the PBL methodology are the following:

- Lectures and problems: The lecturer presents the main ideas of a lecture contents and proposes some application problems which student solves individually. This activity takes about 1 hour.

- Laboratory session: The students must solve a practical problem previously presented during the lecture. In this activity the students work in teams of 2 students. This activity takes about 1 hour and 15 minutes.

- Seminars: A panel discussion at least of 45 minutes is proposed with student teams of 4 students, consisting generally of solving a problem by means of PBL.

- Mini-project: The "students-designers" in team of 4 students, must dedicate to planning, design and develop the control system of the "Water tank". The mini-project is performed during about 2 hours. Weekly, the mini-project is advanced progressively and revised by the tutors.

It is necessary to establish for each activity a prerequisites list in order to provide the background and required skills to obtain knowledge benefits of this activity.

III.1. LECTURES

Lecture is the first step in the learning process for each of the topics in each course's module. The lecturer presents the main topics of the subject contents. It includes some application examples.

In the lecture the teacher takes different roles, ranging from the director of the project, the contractor who requests the implementation of a project for industrial computerization. From these different points of view, the teacher states with a common language without specialization the main elements and objectives are requested to the project. Whereas when he takes the role of project manager, the formulation of requirements is done at more specific professional level, using more technical language. The tutor poses a problem in every subject, and students use all kinds of resources, from the search for components online, literature and application examples.

During a lecture, the teacher previously plans the lecture session based on their professional experience and presents the learning objectives of the session. The teacher contextualizes the subject within the module: the course and the career, based on the problems to be solved and the resources used in this kind of problems. Motivates the subject based on the importance of the problems coped by the subject. Lists and discusses the bibliographic resources that support the concepts that will be presented during the lecture and that students may use to deepen in their study. Presents the key concepts related to the subject, providing the needed details to properly understand them, and specifying the extra available resources and the learning process that students should follow to complete the knowledge's acquisition. Follows a logical order for the argumentation, so that students can acquire the knowledge progressively and uses illustrative examples to clarify the presented concepts. Verifies that students correctly understand the presented concepts and adapts the speech if needed, allowing students to ask questions about the concepts that are not clear enough and observing the students' answers to the control questions. Informs about which skills related to the lecture that will be evaluated. Tells students about the practical application of the presented concepts that will be performed later during the seminars, laboratories and mini-project, and which are related to the lecture, encouraging students to study the issue with sufficient interest.

After the lecture, keeps open communication channels with students, so they can make consultations before the next lecture if necessary. Later, the teacher should analyse the lecture session to improve their professional skills.

Previously to the lecture, the student has studied the recommended previous readings. He is receptive to the teacher's presentation and is proactive. Takes some notes to conceptualize what is being exposed to facilitate its further study, asks for concepts clarification if necessary, answers the questions the teacher addresses to the audience. After the session he should follow the learning method proposed by the teacher.

After the lecture the students will know what is the problem and its importance, how the problem is described, which available alternatives would solve it and the criteria to select an appropriate solving alternative even though they will not have yet the skills to apply the proposed methods. They will have a studying plan that will be organized in the form of a set of labs, seminars and mini-project.

Previously to the lecture the student must be informed about the prerequisites to successfully follow the lecture. If the prerequisites are not satisfied, a formation plan should be supplied.

During the lecture all the conversations in the classroom should be public, addressed to all of the members. The sequence of the lecture should be planned and predictable. The dynamics of the lecture should be attractive to the audience with short and direct explanations. The lecture should not explain all the details of the concepts of the subject, it should

give enough hints to let students autonomously complete the knowledge acquisition. The students must have the chance to formulate questions. Behavior rules must be defined to let the lecture to be productive and the corrective mechanisms should be effective and proportional. Potential distractors of the audience attention should be avoided or minimized.

III.2. LABS

The lab is the first practical exercise that students take to acquire a basic set of skills related to the topic presented in the lecture. The exercises in the labs solve specific and well-defined problems; they are guided, fully documented, and in progressively increasing complexity. The labs provide students with a set of tools and skills that can be used to solve more open problems during the seminars.

Previous to the lab session, the teacher plans the lab session based on his/her professional experience, presents the learning goals of the lab, contextualizes the lab within the subject. Motivates the practical exercise to be performed during the lab based on the importance of the problem it addresses. Lists and comments the equipment, the material and documentation resources needed to perform the lab. Describes the correct utilization of the lab equipment and warns about potential material and personal damage due to inappropriate use. Lecturer also answers student questions during the practical exercise.

During lab, the student, previously, has studied the lab documentation, and has attended the related lectures. He is receptive to the teacher's indications and is proactive. The takes notes to remember the indications and asks for concepts' clarification if necessary. Students work in teams of two students on the practical exercises of the lab. Answers the questions of the teacher related to the exercise. Later, the student should review and document the results of the practical exercises and eventually performs some extra optional exercises.

During lab, the technical assistant, previously, sets the necessary equipment for the lab in each of the workbenches based on the teacher's requests and his/her professional experience. Helps solving problems that could arise related to the equipment, power supply, communications and software, making diagnosis about the safeness and correctness operation of the equipment and replacing damaged components. Later, the technician should analyze the lab session to improve their professional skills.

After the lab the student should have acquired the skills to develop a basic data acquisition system. These skills will be useful in the next related seminars and mini-project exercises.

The student should have attended the related lecture and have read the recommended further lectures and the lab guide. Working teams of two people should have been set.

The lab should be fully documented. The lab guide should start with an introduction that remarks to the concepts of the lecture that are going to be applied in the practical exercise, the goals definition and a list of the material and documental resources that will be needed. The guide should continue with a definition of the practical activities in the following phases of progressive and increasing of complexity: introductory, reinforcement, advanced and optional. The first exercise in the introductory phase should be described step by step. The second exercise in the reinforcement should practice the same concepts and method than before but on a different set of problem data and without the help of the guide in this case.

The third exercise in the advanced phase should practice the application of the previously acquired tools, protocols and skills to solve a small sized and small complexity application problem. A fourth exercise should be defined in an optional final phase to let advanced students to consider further technical questions related to the topic of the lab. The teacher should supervise the practical exercises of the students, answering their questions, guiding them and providing enough hints to let the students find solutions by themselves. Recurrent errors and problems, and interesting student's designs during the lab should be shared with the whole group. Remote desktop sessions of the workbench computer screen could be presented on the slide projector.

III.3. SEMINARS

During the seminars the students must solve problems on the topic of the lecture. They have already exercised on related tools and procedures in the previous laboratories. The type of the activities in the seminar has to do with the application of those tools and procedures to design solutions to specific problems. These problems are manageable parts of more complex problems. The activities involve information searching over the Internet, designs and calculations. Students work in groups of 4 people, but they must explain and share their experiences with the whole group during the seminar meetings. In the seminars of the course different problems and sub-problems related to the design and programming of physical processes controllers are analysed from the perspective of the personal computer control platform studied in this modules.

The seminar starts with a teacher speech remembering about the key concepts presented during the lecture, and the proposition of several related design problems. The problems are decomposed in smaller parts, and the connections between the different sub-problems are discussed. Some of the problems are proposed to the whole group to start a debate, meanwhile other problems are proposed to be selected and solved by the different working teams during the seminar. After the investigation and design activities developed by each of the working teams the spokespeople share the obtained conclusions with the whole group and a second debate starts. The seminar ends with a resume by the teacher and the homework definition for each of the working teams that will prepare the next seminar session.

It's possible that the seminars can be split in several sessions if required by the complexity of the problems.

In the seminar, students practice the concepts presented in the related lectures and verify that these concepts have been assimilated correctly. To relate the previous concepts with

other technical concepts which are usually studied in different subjects, in a contextualized way. Acquire team-working skills. Acquire documentation and presentation skills. Acquire critical searching of information skills.

The student should have attended the related lecture and have read the further recommended readings. Working teams of four people should have been set. A balanced team is recommended with a similar level of initial knowledge for each of its members. One of the members' team will act as spokesperson.

III.4. MINI-PROJECT

During the mini-project students apply the knowledge and skills that they have acquired in the lectures, labs and seminars to develop in an integral way the controller of a physical process. The problem of the mini-project is the highest complexity problem in the course. The working teams in the mini-project are the same as in the seminars. The designs developed by the teams during the seminars are used as components of the mini-project's problem's solution. The teams can use in their own mini-projects, seminar designs that other teams have shared.

After the mini-project, the students should be able to integrate the tools and protocols practiced during the course to develop simple complexity and medium sized applications to control physical process. They should be able to document and present the mini-project process and outcome.

The students should have attended the lectures and completed the practical exercises of the labs and seminars that are related to the module of the mini-project that is going to be developed.

IV. CONTENTS AND SCHEDULING IN THE AIISM UPV

Based in the previous proposals, it is necessary to provide an adequate timeline for the development of the proposals. For example, a reasonable proposal is shown in figure 2.
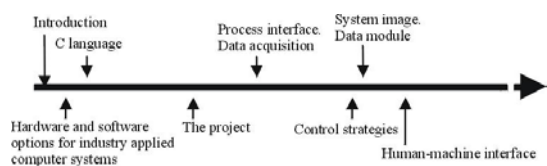

Figure 2. Proposed activities timeline

The following set of activities and time distribution in 15 weeks is proposed (See figure 3).

Type	Topic
INTRODUCTION	
Lecture	Introduction to industrial informatics
Lab	Development environment - Programming the "Hello World"
Seminar	C programming (1) - Basic resources
Miniproject	Presentation of the problem to solve
Lecture	Structure and design of industrial informatics systems
Lab	Event oriented programming
Seminar	C programming (2) - Programming tools
Miniproject	Analysis of the project requirements
COMPUTER	
Lecture	Computer architecture
Lab	Using libraries in C
Seminar	C programming (3) - Libraries
Miniproject	Project formal specification
PROJECT PLANNING	
Lecture	Project management (1)
Lab	Tools for project management
Seminar	Discussing cases of project management systems
Miniproject	Project planning
PROGRAMMING + DATA	
Lecture	Modular programming
Lab	Modular programming in C
Seminar	Modular programming resources
Miniproject	Modular decomposition of the program
Lecture	Data representation and sharing
Lab	Data sharing between C modules
Seminar	Choosing the appropriate data representation
Miniproject	Implementation of the shared data module
PROCESS INTERFACE	
Lecture	Process interface (1) - Introduction and digital input
Lab	Digital input
Seminar	DAQ card (1) - Introduction and digital input
Miniproject	Implementation of the process interface module (1) - DI
Lecture	Process interface (2) - Digital output
Lab	Digital output
Seminar	DAQ card (2) - Digital output
Miniproject	Implementation of the process interface module (2) - DO
Lecture	Process interface (3) - Analog input and output
Lab	Analog input and output
Seminar	DAQ card (3) - Analog input and output
Miniproject	Implementation of the process interface module (3) - AIO
USER INTERFACE	
Lecture	Graphical user interface (1) - Introduction
Lab	Programming GUI controls
Seminar	Graphical user interface for the industry (1) - Basic
Miniproject	Implementation of the user interface module (1) - Basic
Lecture	Graphical user interface (2) - Advanced resources
Lab	Programming a GUI for an industrial application
Seminar	Graphical user interface for the industry (2) - Advanced
Miniproject	Implementation of the user interface module (2) - Advanced
TASKS	
Lecture	Task scheduling
Lab	Basic scheduler
Seminar	Scheduling strategies
Miniproject	Implementation of the task scheduler module
REGULATION	
Lecture	Foundations and continuous control
Lab	Programming regulation strategies (1) cc
Seminar	Control strategies (1) cc
Miniproject	Implementation of the regulator module (1) cc
Lecture	Event-driven control
Lab	Programming regulation strategies (2) edc
Seminar	Control strategies (2) edc
Miniproject	Implementation of the regulator module (2) edc
PROJECT (2) ENDING	
Lecture	Project documentation and presentation
Lab	Tools for project documentation
Seminar	Project documentation strategies
Miniproject	Documentation and presentation of the project

Figure 3. Lectures, Seminars, Labs and Miniproject timetable

The initial set of activities introduces two lectures about basics computer architecture and the applicability of computers to industry.

Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lecture	Introduction to industrial informatics	x														
Lecture	Computer architecture		x													

The following set of activities develops skills on C programming to be applied on the application creation. This is a horizontal requirement of the module, so it is spread along the course.

Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Seminar	C programming basics	x	x													
Seminar	Programming tools		x													
Lab	Development environment		x													
Lab	Hello World		x													
Lab	Modular programming				x											
Lab	Data representation				x											
Lab	Event oriented programming								x							
Miniproject	Modular decomposition				x											
Seminar	Using libraries				x											
Seminar	Data sharing				x											
Miniproject	Data sharing module implementation															

Another basic requirement of an engineer is the correct management of a project. This is the objective of the following activities that are spread along the course. This is also a horizontal content of the module, so it is spread along the course and in a position where student understands its implications.

Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Seminar	Project management				x											
Miniproject	Project design				x											
Miniproject	Analysis of project specificacion				x											
Miniproject	Test and validation															x
Miniproject	Module integration														x	

The following groups of activities work with each aspect of an industrial application.

Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lecture	Process interface						X									
Seminar	DAQ card National Instruments USB-6008						X	X								
Lab	Digital input						X									
Lab	Digital output							X								
Lab	Analog output								X							
Lab	Analog input								X							
Miniproject	Process interface implementation							X	X							

Figure 4: DAQ card NI USB-6008

The first set of activities deals with the connection of the computer to the real world, the so called "process interface" via DAQ card (See figure 4). This tends to motivate the student because they see the interaction with physical reality.

Taking into consideration that the actual student generation is accustomed to stunning visual user interfaces, this is the moment for introducing another motivating set of activities related to this aspect. (See figure 5).

At this point, it is necessary to start coordinating actions inside the application. So the next set of activities introduces the very basics around task coordination/scheduling.

Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lecture	Task schedulling								X							
Lab	Basic scheduler								X							
Miniproject	Task schedulling implementation								X							

And, finally, the student needs to see that your development works. From the point of view of the teacher, it is adequate to introduce here the regulation problem according to the following activities.

Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lecture	Regulation													X		
Seminar	Regulattion approaches													X		
Miniproject	Regulation strategy implementation													X		

V. CONCLUSIONS

This article shows some of the characteristics of the European Project Medis as part of the project TEMPUS. In particular, the paper presents the structure of Advanced Industrial Specialization Technology Module (AIISM) proposed by the Polytechnic University of Valencia (UPV). We are working in collaboration with four European universities to adapt to the model of European Credit Transfer System (ECTS), the curricula of some non-European countries and create an equivalent master in these PCs countries. We also discussed different learning activities based on PBL methodology, and focussed at a design problem of a particular industrial computer application: *The Water Tank*. It has been developed at UPV for the creation of master classes, laboratories, seminars and mini-project. The four other US universities address this same problem from the point of view of their subject.

VI. REFERENCES

[1] Boud, D., & Feletti, G. (Eds.). (1998). The challenge of problem-based learning. Psychology Press.
[2] Norman, G.R. "Problem-solving skills, solving problems and problem based learning", Academic Medicine. Vol. 67, Nº 9, 1992.
[3] Duch, B. Problems: A Key Factor in PBL, Centre For Teaching Effectiveness, University of De Laware. available at www.udel.edu/pbl/cte/spr96-phys.html

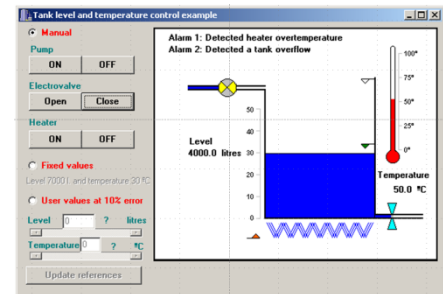| Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lecture	Graphical user interface											X				
Seminar	Graphical user interface for the industry											X				
Lab	A GUI for an industrial applications												X			
Miniproject	User interface implementation												X	X		

Figure 5: GUI for the "Water tank"

Teaching industrial networks with a problem-based learning approach

M. Sousa, L. Almeida, P. Portugal, A. Sousa

University of Porto, Porto, Portugal

Abstract—**Problem-Based Learning (PBL) has been used in several domains for almost two decades as a more efficient way to develop the desired skills in students. We believe this approach lends itself well to teaching Industrial Systems, allowing the students to acquire skills that enable them to solve concrete problems in an industrial context. In this paper we present a course for teaching Industrial Networks using PBL. The course itself is organized into several sections each of which addresses a real-world problem together with methods and technologies for solving it. The students are first shown how, and then required to implement a distributed industrial automation system using networked devices. We use representative network technologies, namely CAN/CANopen, Modbus/TCP, Modbus/RTU and OPC. The course uses a plant floor simulator that allows creating simple to complex configurations of complete production lines. This high reconfigurability allows an easy tuning of the simulator to the students' background and skills at each phase of the course. A version of this course has been instantiated within the Integrated Master in Electrical and Computer Engineering at the University of Porto since 2010 with encouraging results.**

Index Terms— **Industrial systems, industrial networks, fieldbuses, problem-based learning.**

I.   INTRODUCTION

IN order to reduce costs and improve efficiency, industries that extract, process, produce or manufacture goods typically rely on Information and Communication Technologies (ICT). These technologies are used to automate, monitor, and manage to the best possible extent the industrial process. Due to the specificity of these use cases that have very different requirements when compared with traditional software, i.e., that does not need to interact and control real-world objects, a service industry has grown providing both specific hardware as well as ICT solutions to be used in industrial automation. These solutions include software packages that range from SCADA (Supervisory Control and Data Acquisition) to DCS (Distributed Control System), MES (Manufacturing Execution Systems), or SoftPLCs. Example hardware includes PLCs (Programmable Logic Controllers), CNCs (Computer Numerical Controllers), RTUS (Remote Terminal Units) and variable speed motor drives.

Many networking solutions are also available, with each vendor typically focusing on a specific communications technology and/or protocol family. Most of these networks and communication protocols are designed to fulfil the strict timing requirements of the industrial automation domain.

Designing an industrial automation project requires a good knowledge and understanding of the requirements and potential solutions. Transforming the design into reality in turn requires engineers capable of configuring, programming, and installing these domain specific hardware and software solutions. Commercial companies and industry therefore need qualified personnel in the domain of industrial automation, and the initial qualification is typically obtained at higher education institutions.

Problem-Based Learning (PBL) is an approach to teaching that uses problem solving to develop the desired skills in the students [1]. This is particularly well suited to teach industrial systems for which the students need to acquire a set of technical skills [2]. Some previous works already applied PBL to teach industrial networks, particularly Fieldbuses [Mogames, 2012]. In this paper, we will also focus on teaching industrial networks and Fieldbuses. However, when comparing with [3] we will take a more architectural and holistic view.

Our course has been developed within the MEDIS European project which is a collaboration between several universities from both inside the European Union (Bulgaria, Germany, Portugal, Spain and Sweden) and outside (Kazakhstan, Russia and Ukraine). The project is led by Universidad Polytechnica de Valencia (Spain) and aims at designing and instantiating courses focused on teaching the skills required for designing and implementing industrial automation solutions.

This paper describes our proposed course on Industrial Networks. An adapted instance of this course has been applied

at the University of Porto's Faculty of Engineering, since 2010, with encouraging results.

We start with a section explaining the course outline and organization. The following section focuses on the teaching approach, and the organization of the classes themselves. The following sections explain better the material and skills covered by each of the chapters of which the course is composed. The final section provides an outlook of how this course can be implemented, focusing on a similar course currently running at University of Porto.

II. COURSE ORGANIZATION

The Industrial Networks and Fieldbus course has been organized to follow the teaching approach proposed by the MEDIS project, which focuses on a problem based learning. The students are asked to solve small but typical and representative problems in the industrial automation domain, and all the knowledge and skills required to complete these tasks are presented and practised with a particular problem in mind. The problems are chosen so that by the end of the course the students have had the opportunity to cover all the fundamental industrial networking knowledge and skills required of an industrial automation engineer.

The course itself is organised into 7 chapters, with each chapter focusing on a specific set of skills and problem domain. Industrial automation solutions typically follow a hierarchical organization (ISA 95 [4]), with the top layers focusing more on managing the enterprise wide resources (e.g. buying of raw materials, scheduling of work orders, handling orders), the middle layers on monitoring and managing the production systems, the lower layers in controlling the production systems, and the lowest layers consisting of the interface between the machines and control hardware/computers (basically the sensors and actuators). The 7 chapters also somewhat follow this organization, with some chapters focused on the approaches and solutions used for communication between the systems inside the same layer or across layers, and other chapters used to convey the hierarchical organization itself.

More specifically, the course has the following chapters:
- Course Chapter 1 - Introduction
- Course Chapter 2 - Modbus/TCP
- Course Chapter 3 - Discrete Event Control
- Course Chapter 4 - Modbus/RTU
- Course Chapter 5 - CAN
- Course Chapter 6 - CAN Open
- Course Chapter 7 - Hierarchical Control

Chapter 1 introduces the field of industrial communication networks, and provides an overview of how these are organized. Chapter 2 is dedicated to the Modbus/TCP communication protocol, which is typically used between devices of layer 2 (of ISA 95). In this chapter the learning activities focus on a hands on approach of implementing the protocol, and some very basic tests with a model of a manufacturing cell.

Over the following 2 weeks (chapter 3) focus is placed on control strategies for discrete event systems, and students are provided the opportunity to start developing the control application for their mini-project.

Chapter 4 focuses on the RTU and ASCII versions of the same Modbus protocol already covered in chapter 2. These protocols are commonly used to interconnect devices on ISA95 layers 1 and 2. For this reason, the practical implementation work will focus on the interaction between a computer and a micro-controller, both using this protocol. This chapter has the additional objective of showing how communication protocols are themselves organized in layers, according to the OSI (Open Systems Interconnection) model.

Chapter 5 is dedicated to another protocol commonly used to interconnect devices in ISA 95 layer 1 to those of layer 2, namely CAN and CANopen. This protocol mostly covers the 2 lower layers of the OSI model, therefore most of the implementation effort goes towards getting two micro-controllers to exchange very basic and simple data frames.

Once a working CAN network has been built, the students spend the following 3 weeks studying the CANopen protocol. This protocol works over the CAN bus, but adds a relatively complex application layer based on objects, and with distinct device profiles for distinct types of devices (basic I/O, speed drives, ...). Both the event-triggered (asynchronous PDOs) and time-triggered (synchronous PDOs) approaches are considered.

The final chapter focuses on the hierarchical control architecture, as well as OPC. The classic version of OPC (OLE for Process Control) is an example of an inter-connection technology commonly used between layers 2 and 3. The new version, OPC-UA (Unified Architecture) is also covered.

III. CLASS ORGANIZATION

Like all other courses in the MEDIS project, this course is organized in 15 weekly sessions, each of which is from 4 to 5 hours long. Teaching activities follow the problem based learning approach, and the students are asked to work on a project during the course.

The project is presented in the first week, and work starts with requirements analysis. Special emphasis is placed on the real-time requirements of the system under control, as the delays imposed by the networks will heavily affect the achievable performance of the control application. In each of the following weeks a specific part of the project will be analysed and implemented, where the implementation will typically require the knowledge and skills presented during that week.

The project consists on developing a control application to automate a small manufacturing cell, consisting of conveyors and machines. The conveyors transport work-pieces to and from the machines, and these operate on the work-pieces transforming them from one type to another. The simulator is available in Free and Open Source and is called Shop Floor

Simulator [7].

The manufacturing cell is simulated by a Java-based application. All input/output communication between the simulated manufacturing cell and the control application is done using the Modbus/TCP communication protocol, with the manufacturing cell working as a Modbus server and the control application as a Modbus client.

The manufacturing cell simulator is highly configurable particularly concerning the size of the plant floor, the instantiation and location of each conveyor and machine (milling, 3D portal robot, automated warehouse, pushers), as well as the tools on each milling machine and the transformations each tool/milling machine pair will have on each work-piece type. Several conveyor types are supported, namely linear, rotator, and slider. All conveyors are able to handle the simple bi-directional transportation of a work-piece. Additionally, the rotator conveyors are able to rotate on a vertical axis therefore aligning themselves with conveyors on the x or the y axis, whereas the sliding conveyors can slide on rails on an axis. Using these conveyors, machines and automated warehouses, manufacturing cells ranging from the very simple to the very complex can be easily simulated.

Fig1. Manufacturing Cell Simulator

The course typically uses a relatively simple layout including only two machines and several conveyors, as the focus of this course is on the communication protocols and not on the control system software itself. However, more complex layouts are certainly possible, and the chosen layout should be adjusted to match the skills of the students taking the course.

All conveyors and machines are controlled through digital controls (switch motor on/off) which are mapped onto Modbus coils. Feedback is provided through simulated digital sensors (detect presence of work-piece, end-effect sensors, etc.) which are mapped onto Modbus 'input discretes'. A simple conveyor, for example, has two actuators (one motor for moving in each direction), and one sensor (senses a work piece on the conveyor). More complex machines (e.g.: 3 axis portal robot, milling machine, rotating conveyor, sliding conveyor) have many more actuators and sensors.

The control application will therefore consist of a discrete event control system, that can be easily modelled using any state machine modelling language (e.g., UML state chart, IEC 60848 Grafcet, IEC 61131-3 SFC, etc). In order to guarantee that all students are capable of implementing the control application, this course includes a two two week long chapter on the modelling and implementation of discrete event control

applications based on state machines.

At the beginning of each weekly session a problem (or a particular detail of a larger problem) is presented, and the remainder of the class is dedicated in finding and implementing a solution to this problem. This consists of a preliminary presentation by the teacher of the relevant technologies already available, as well as the underlying mechanisms and theory in which they are based. This is followed by a short seminar where the students are tasked to research similar technologies or possible design patterns that may be applied when solving the problem at hand.

Each session then continues with laboratory practice. Here the students are expected to implement a small part of the solution to the problem. During this laboratory phase the students follow a sequence of tasks that are provided and explained. The first contact the students have with each new technology is therefore well supported and their actions guided. This approach allows the students to gain confidence in the knowledge transmitted and gained during the first two phases of the weekly session.

The last phase of each weekly session is devoted to advancing the course project. Here the students are expected to work with a higher degree of autonomy, and gain increased confidence in their skills.

IV.   COURSE CHAPTER 1 - INTRODUCTION

This is the first chapter of the course and addresses the problem of computer communication within industrial systems focusing on the respective communication requirements and on the network technologies that were developed along the years to cater for such requirements. This historical perspective also covers the evolution of the architecture of such systems to the Computer-Integrated Manufacturing (CIM) architecture, leading to the definition of multiple levels with different communication requirements. From these requirements we present and characterize the specific network technologies developed for the industrial context, namely the fieldbuses and, more recently, Real-Time Ethernet.

The goal of this chapter is to allow the students to understand why industrial systems evolved into a CIM architecture as well as the characteristics of such architecture, the information flows therein and the networking technologies that support it. The focus is on the lower levels of the CIM architecture, typically Process/Machine level and Cell level, within which we can find the most stringent time requirements.

The specific network technologies will be the subject of the seminars that the students will need to prepare, including both general Internet protocols, general-purpose Local Area Network (LAN) technologies, Real-Time Ethernet and Fieldbuses.

The laboratory component of this introductory chapter develops around concrete protocol inspection using a network analyser, namely Wireshark [8], and a LAN running on Ethernet or WiFi and preferably with Internet connection. It

allows observing network frames, their composition, the protocol layers, etc. Some simple experiments with Round-Trip Delay measurements will allow observing network delays. Finally, when Matlab is available, the Truetime toolbox can be used to observe the impact of network delays in distributed feedback loops.

The course mini-project is defined in this chapter, consisting of a small manufacturing cell with two machines and several conveyors. The students are organized in groups of two or three and are requested to define the requirements of the cell as well as a preliminary control architecture.

V. COURSE CHAPTERS 2 & 4 - THE MODBUS/TCP AND MODBUS SERIAL PROTOCOLS

The second and fourth chapter of the course focuses on the Modbus protocol and its variants. The second chapter is dedicated to Modbus/TCP, and the fourth to Modbus Serial. The goals of these two chapters is to allow the students to understand the hierarchical nature of networking protocols by using a protocol whose hierarchical nature is evident and simple. A second goal is for the students to become acquainted with the client/server interaction model through the use of Modbus/TCP, as well as the master/slave interaction model of Modbus/RTU. The students should also come to understand that this last interaction model may be used as a protocol to control access to a shared communication medium (i.e. a MAC - Medium Access Control protocol).

The two Modbus chapters are interrupted by chapter 3, as the students must first learn how to develop discrete control system control programs before they can practice using the Modbus protocols in a realistic scenario.

The Modbus/TCP chapter starts by presenting the Modbus data model, with the typical master/slave or client/server interaction model. This proceeds with the hierarchical nature of the Modbus protocols with a brief overview of an example network. Next the Modbus application layer is studied in detail, including the format and semantics of the main Modbus functions, as well as the error messages that can be expected. Focus then continues with the mapping of the application layer frames into the TCP protocol.

The first laboratory session during this chapter is used to allow the students to implement their first TCP based client/server application (echo), using the sockets API. In the second laboratory session this application is extended to implement the Modbus protocol. For the project work, the students are asked to develop a very simple control application that merely controls workpieces on 3 conveyors. The Modbus/TCP code implemented in the laboratory sessions is used to connect to the plant floor simulator.

In the chapter dedicated to Modbus Serial presents the two possible mappings of Modbus onto a serial physical layer, namely the RTU and ASCII mappings. Error detection with CRC (Cyclic Redundancy Check) and LRC (Logitudinal Redundancy Check), as well as the parity bits of the serial communication layer are discussed.

The laboratory sessions of the Modbus Serial chapter are used to first implement a Modbus serial master on a PC (Personal Computer), which is tested against an existing Modbus slave. In the second laboratory session a Modbus slave is implemented on an Arduino board, and this is tested against the master written in the previous week.

During this chapter, the students are asked to use the Modbus series implementation on the PC and Arduino to map I/O on the Arduino onto the memory area of the controller application running on the PC. This is then used to add functionality to the controller software. The Arduino's I/O are used as an HMI (Human Machine Interface) based on buttons and lights, and the control application is augmented with mode based operation – running, stopped, suspended, etc. The mode changes are activated through the HMI buttons, and the current mode is shown using the lights.

VI. COURSE CHAPTER 3 - DISCRETE EVENT BASED CONTROL

This third chapter deals with computer control of real world systems that can be modelled as discrete event systems. Although this topic does not actually deal with any industrial communication paradigms or technologies, these automation systems are often distributed in nature and require the use of industrial networks. It may therefore be said that these systems function as one of the main 'users' of industrial networks, and provide the main requirements that these network need to satisfy.

Additionally, to use an industrial network in a scenario that approaches real-life, it becomes necessary to build a distributed control program. The main goal of this chapter is therefore to provide the students with sufficient tools and skills to build a discrete event system based application, that may later be distributed amongst several nodes. This application will then serve as a foundation in which the industrial communication technologies are practised on.

Many different modelling tools may be used to model discrete event systems, most of them some variation of the basic Mealy or Moore Finite State Machines. One that has found a lot of traction in the industrial automation world is Grafcet, since it allows multiple active states simultaneously, and therefore eases the modelling of the execution of sequences in parallel, in a graphic structure. This modelling approach has been standardised as IEC 60848, and is therefore adopted for this chapter. Another reason to use Grafcet is that it has also been adopted and extended by the IEC 61131.3 standard that defines programming languages for PLCs (Programmable Logic Controllers) - the SFC 'programming language' (Sequential Function Chart) is actually an extension of the Grafcet modelling approach.

During the 2 weeks dedicated to this chapter, the students learn the modelling tools features, and use it to model basic automation sequences. Special focus is placed not only on the sequential execution processes, but also in the structures commonly used for modelling parallel sequences. Special

focus is also placed on the design patterns used to handle the sharing of common resources whose access must be atomic. Healthy implementation is very important and, to the target students, is to be done using general purpose programming language C.

In the seminar part of this module, students are also asked to research tools for discrete event control, get acquainted with their characteristics and economical value. The final discussion on the summary table of the tools found in the market will lead the students into knowing the market and recognizing the economical value of the area targeted by the course.

In the lab part of the module, students are invited to use Modbus TCP to control a simulated plant. First in a sequential manner but then in a configuration involving shared resources.

The project part of the module, reserved for a second week of classes, is to lead the student into integrating the plant developed in the lab portion of the module into a larger manufacturing assembly – to create the full course project.

VII. Course Chapters 5 & 6 - The CAN Based Protocols

During chapters 5 and 6 the students are provided with an opportunity to explore the CAN bus as well as the CANopen communication protocol. These two technologies were chosen for three main reasons. The first is that the students are once again provided with an example of the layered structure of the communication protocols, as the CANopen protocol is one of several possible protocols using the CAN bus physical and data link layers.

The second reason is that these technologies support the event based interaction, in contrast to the poll based (client/server or master/slave) interaction model used by the Modbus protocols. This allows the students to come into contact and use this communication paradigm. The third reason, somewhat related to the second, is the fact that the CAN bus uses a different medium access control method that is not based on the master/slave approach commonly used on other fieldbus networks (including Modbus/RTU).

The CAN bus chapter is two weeks long. In this chapter focus is placed on the basic physical properties of the CAN bus, as well as its medium access control method. This method is based on synchronising the beginning of transmission of every frame, allowing collisions that are then automatically resolved by the physical bus that implements a wired-or. The chapter also covers the structure of CAN bus frames, and the message based addressing model employed by this bus.

During the laboratory sessions the students are given the opportunity to build a small CAN bus network using Arduinos [5] with a CAN bus shield [6]. Simple messages are exchanged between the Arduinos conveying the fact that a specific set of buttons on each Arduinos has been pushed by the user. This information is then shared with the Modbus/RTU slave previously implemented in the Arduinos. In the project sessions this same information is conveyed to the control application where a new operational states (on, off, standby) are added. The user should then be able to change the state of the control application by pressing on the physical buttons on any Arduino on the CAN bus. Similarly, the current state of the control application is reflected on the leds of all the networked Arduinos.

The following CANopen chapter is three weeks long as the structure of this protocol is a little more complex. During this chapter the students have the opportunity to become acquainted with the data model supported by the CANopen protocol, the structure of the CANopen frames, as well as both the event and time triggered approaches supported by CANopen.

During the first week special emphasis is placed on the data model, as well as the internal data structures that each CANopen network node is mandated to include – a structure known as the Object Dictionary (OD). An implementation of this data structure is done during this first week. The laboratory session provides hints of the possible implementation design patterns that may be used, taking into account the properties of the OD – a hierarchical data structure, with each node typically containing only a very few of a large possible number of numbered entries. Since CANopen nodes are typically embedded devices with constrained memory size, efficient data structures for sparse memory entries are proposed.

Although the semantics of the data included in the OD is explored during the first week, it is only on the second week of this chapter that the communication protocols used to exchange the data stored in the OD are covered. Both the SDO (Service Data Objects) and the PDO (Process Data Objects) are mentioned, although only the second are implemented in this second week. During the third week both the synchronous and the asynchronous transmission of the PDOs are explored, and implemented in the Arduinos. Each Arduinos that was connected to the CAN bus network in the previous chapter now becomes a full-fledged CANopen node. One of the nodes is configured to run as a CANopen master (sending the SYNC synchronisation messages), while the others act as slaves.

During this chapter, the course project is adapted to use the CANopen protocol, instead of the simpler CAN bus network used in the previous chapter. The functionality of the discrete control application is not however significantly changed.

VIII. Course Chapter 7 - Hierarchical Control

The last chapter addresses the problem of defining a suitable architecture for industrial systems that supports the integration of all heterogeneous equipment therein, from the shop floor to the management. The goal is to allow the students to understand the information flows present in integrated industrial systems and from there to understand their communications architecture. The students then learn about the attempts that were made to come up with such architecture and the difficulties that struck some related standardization efforts. The students then learn about several

integrated architectures that have been proposed by different industrial communications equipment manufacturers.

The different flows of information are identified and their characteristics studied, in particular with respect to time constraints and amount of information per transaction. This will show the need for a layered architecture with different technical solutions for each level. A brief survey of related standards and solutions offered by the manufacturers of current industrial communication systems is also included.

During the second week dedicated to this chapter, the focus changes to the problem of defining a suitable communication protocol to transfer process data between layers 2 and 3 of the ISA 95 model. Although this data transfer typically does not have tight timing constraints, it may require the transfer of more complex data structures. This is in contrast to the communication protocols used inside layers 1 and 2 of ISA 95 model, with strong real-time constraints and very simple data structures (for example CANopen and Modbus previously studied).

In this context classic OPC is mentioned as an example of an intergration technology that is often used for transferring process data between the plant floor and the manufacturing execution control levels. Although the classic version of OPC actually defined an API (Application Programming Interface) for the exchange of data between two programs, when used together with DCOM middleware (Distributed Component Object Model) on which it is based, it becomes possible to transfer data over a communication network.

The new version of OPC, OPC-UA (Unified Architecture), is also mentioned. Although the main objective was to break with the Microsoft based technologies (DCOM) and become vendor neutral, this new version introduces a new data model that is more flexible. Each data point (or object) may contain many links to other data points (or objects), with objects residing in any network connected node. It is important for students to become conversant with this new paradigm, where the data points may be organised in a mesh (instead of hierarchically).

During the laboratory and project sessions of the first week of this chapter, students practice the development of GUIs (Graphical User Interface) dedicated to supervision and control at a higher control layer. This GUI is built using a SCADA package which connects using the Modbus/TCP protocol to the plant floor. A very simple open source SCADA package is suggested, both to reduce costs, and to simplify the first contact of the students with these types of software solutions.

During the second week the laboratory session is dedicated to experimenting with classic OPC based commercial solutions. Several Modbus-OPC gateways are suggested. Being the last week of the module, the project session is dedicated to the presentation of the projects by the students.

IX. CONCLUSION

Problem-Based Learning has been demonstrated to facilitate the development of practical skills in students, which is particularly useful in the context of teaching industrial automation. Therefore, an European project named MEDIS was launched to bring several universities together to develop a set of relevant courses in that domain following the PBL approach.

In this paper we described the course on Industrial Networks and Fieldbuses that we developed in the scope of such project. This course follows an architectural holistic view to industrial systems, focusing on the networking technologies that support systems integration. The course is organized in several chapters, each with lectures, seminars, laboratory and mini-project. In particular, the laboratory uses technologies that are easily available, facilitating the course deployment. An instance of this course has been applied and ran with encouraging results at the University of Porto in Portugal.

REFERENCES

[1] Markham, T. Project Based Learning. Teacher Librarian, 39(2), 38-42. 2011.
[2] Noravian, Armineh, Building Technical Capital Among Community College Students in Advanced Technological Education. SSRN, May 23, 2014. http://ssrn.com/abstract=2555124 doi:10.2139/ssrn.2555124
[3] Omar Mogames, Unveiling fieldbus network technology through project-based learning, FIE 2012, IEEE Frontiers in Education Conference, pp. 1-5, 2012. doi:10.1109/FIE.2012.6462259
[4] ANSI/ISA-95.00.01-2000, Enterprise-Control System Integration
[5] Arduino Uno Board web site (last visit 12 May, 2015) http://www.arduino.cc/en/Main/ArduinoBoardUno
[6] CAN bus Arduino shield web site (last visit 12 May, 2015) https://www.sparkfun.com/products/10039
[7] Andre Restivo - Shop Floor Simulator (last visit 12 May, 2015) https://github.com/arestivo/sfs
[8] Wireshark website, (last visit 12 May, 2015) https://www.wireshark.org/

M. de Sousa received the B.S., M.S. and Ph.D. degrees in electrical and computing engineering from the University of Porto, Portugal, in '92, '96 and '05. He has been teaching at this same university from '92, and is currently an Auxiliary Professor and a researcher at INESC TEC (Institute for Systems and Computer Engineering of Porto - Technology and Science). His research interests revolve around the industrial automation domain, and include programming languages for industrial control, and embedded real-time systems. He has published several papers in these topics, as well as 4 book chapters on programming industrial controllers and their networking using Modbus.

Luis Almeida graduated in Electronics and Telecommunications Eng. in 1988 and received a Ph.D. in Electrical Eng. in 1999, both from the University of Aveiro in Portugal. He is currently an associate professor in the Electrical and Computer Engineering Department of the University of Porto (UP), Portugal, and a senior researcher

in the Telecommunications Institute at UP where he coordinates the Distributed and Real-Time Embedded Systems (DaRTES) lab. Among several appointments, he was a member of the IEEE Technical Committee on Real-Time Systems (2008-2013), Program and General Chair of the IEEE Real-Time Systems Symposium (2011-2012 respectively) and Vice-President of the RoboCup Federation (2011-2013) being Trustee of this organization since 2008. His research interests include those related to real-time networks for distributed industrial/embedded systems including for teams of mobile robots.

Paulo Portugal received the Licentiate, MSc and PhD degrees in Electrical and Computer Engineering from the University of Porto (UP), Portugal, in 1992, 1995 and 2005, respectively. Currently, he is Associate Professor in the Department of Electrical and Computer Engineering at the University of Porto and a researcher at INESC TEC (Institute for Systems and Computer Engineering of Porto - Technology and Science). He is author or co-author of more than 100 technical papers in the areas of industrial communication systems, real-time networks for embedded systems and dependability modeling and evaluation.

Armando Sousa received his B.Sc., M.Sc. and Ph.D. degrees in electrical and computing engineering from the University of Porto, Portugal in the years of 1995, 1997 and 2004, respectively. He is currently an Auxiliary Professor in the same faculty and an integrated researcher in the INESCTEC (Institute for Systems and Computer Engineering of Porto - Technology and Science), in the CROB group (Centre for Robotics and Intelligent Systems). He received several international awards in robotic soccer under the RoboCup Federation (mainly in the small size league). He has also received the Pedagogical Excellence award of the UP in year 2015. His main research interests include education, robotics, data fusion and vision systems. He has co-authored over 50 international peer-reviewed publications and participated in over 10 international projects in the areas of education and robotics.

A learning module for control of embedded systems through mobile devices

S. Cenevska, M. Malekzadeh, A. Thekkilakattil, R. Dobrin

Mälardalen University, Västerås, Sweden

Abstract— This paper describes the learning activities proposed to implement the module of "Mobile Devices as Remote Monitoring and Control Systems for Embedded Applications".
This module is structured into lectures and labs, seminars and mini-projects that provide, firstly, the necessary knowledge and possibilities to practice towards learning the fundamentals of programming mobile devices and secondly, to remotely monitor and control embedded systems.

In this paper, the structure of the module, the pre-requisites, the learning activities and the schedule are provided. The developed methodology is mainly based on problem based learning (PBL) and other accepted active learning techniques. They intend to create a realistic working environment similar to what students will experience in their future career. This model is based on the educational goals proposed by the Accreditation Board for Engineering and Technology (ABET)

Index Terms—embedded systems, mobile applications, control systems, Teaching module

I. INTRODUCTION

Embedded systems are ubiquitously deployed in a large number of industrial applications, and the need of their monitoring and control increases with the complexity of the systems in which they operate. At the same time, the usage of mobile devices has exploded over the past years and brings an attractive possibility to use them in the context of control and monitoring of embedded application. However, monitoring and control of embedded systems is typically characterized by high dependability requirements, e.g., reliability as well as security, which introduces a wide range of new challenges to the mobile monitoring approach.

The use of mobile devices now exceeds PC and that trend will continue in the future (Figure 1 [1]). Mobiles are preferred devices when it comes to work, surfing or communication. Time spent on the internet, by device, has shown that the popularity of the phones has drastically increased. It is easy,

fast and affordable. Thus programming mobile applications takes a good place in the market. The usage of the most popular nowadays mobile operating system, Android, confirms this trend. The popularity of Android is increasing due to the fact that it is free and open source, and Google play is a "free market" for the Android applications. No costly
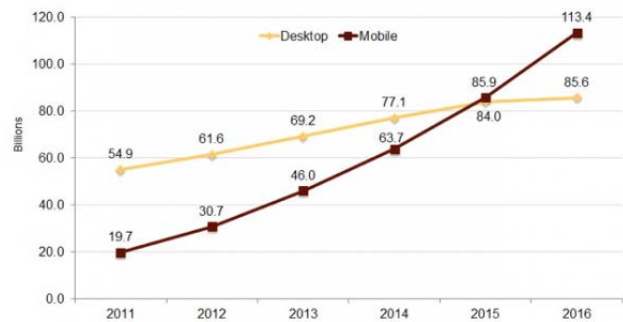
Figure 1 Mobile vs. PC usage

obstacles development as well as the rich and secure application integration attracts applications developers.

Within the EU project MEDIS ("A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics Systems") a learning module is to be created which addresses mobile devices as remote monitoring and control systems for embedded applications. A problem-based learning (PBL) methodology will be used [1,2] which will train highly qualified engineers. The wider objective of the MEDIS project is the integration of a problem-based learning methodology in industrial technology masters degrees for the formation of high qualified engineers in the design of advanced industrial informatics systems based on microcomputers, industrial computers and mobile and cloud computing platforms.

The developed methodology is mainly based on problem-based learning (PBL) and other accepted active learning techniques with the deliberate intention of creating the realistic working environment which students will experience in their future career. This model is based on the educational goals proposed by the Accreditation Board for Engineering

and Technology (ABET) [12]. The general aims of the approach are as follows: to guarantee that the student has the knowledge about the fundamentals of the specialization, to encourage the students to work as part of a team in solving industrial problems and to encourage students to apply practical skills in order to improve their problem solving abilities in the situations they will meet in their working environment. Furthermore, to develop the capacity to adapt to any new computer based systems that may appear in the future due to the rapid advances in this area. Taking into account this methodology, all the learning activities are driven around a reference project. The specification, design, implementation and validation of the course "mobile devices as remote monitoring and control systems for embedded applications" are taught using the liquid tank application and different mobile devices bridged over communication infrastructures based on Ethernet, WI-FI or Bluetooth protocols. The size and complexity of this system is adequate to support the explanation of the essential concepts.

II.   THE REFERENCE PROJECT

The specification, design, implementation and validation of remote monitoring and control systems are taught using the liquid tank application together with a communication infrastructure, as well as mobile devices to be used as remote controllers. The communication infrastructure consists of modules that provide Ethernet, WIFI and Bluetooth connectivity. The liquid tank application is a closed loop system in which the temperature of the liquid in the tank is regulated at a reference temperature using an analog temperature sensor, a digital over-temperature sensor, and a heating element. The amount of liquid in the tank is regulated using an analog level sensor, a digital overflow sensor, a pump and an electronically controlled solenoid valve.

Figure 2 shows a diagram of the sensors and the actuators used in the tank application.
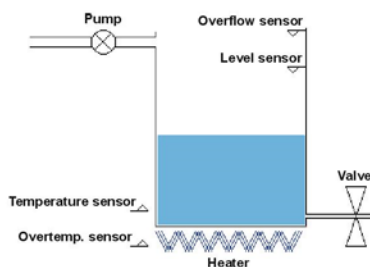

Figure 2. Diagram of the reference model

Specifications of the sensor data and the signals to drive the actuators are given below:

The sensors and actuators can be connected to a personal computer (PC) using a Data Acquisition (DAQ) module such as National Instruments USB-6008 DAQ, or any Arduino microcontroller board that is equipped with USB connectivity. If this method is used, then the Ethernet, WIFI and Bluetooth connectivity of the PC can be used as the communication infrastructure. Alternatively, these signals can be directly connected to a simple microcontroller module that has built-in Ethernet, WiFi or Bluetooth module, such as the "WiFi DipCortex" board which has a built-in WiFi module. Other examples are "Arduino Yún" (for both WiFi and Ethernet) and "Arduino BT" (for Bluetooth). As a last option any basic microcontroller board can be used together with add-on boards that provide WiFi, Ethernet or Bluetooth connectivity.

III.   THE LEARNING MODULE

The learning module consists of a series of self-contained, but related, chapters that cover some of the fundamental concepts in mobile device programming, security and reliability. Each chapter aims to balance between theory and practice by having 1) a lecture that introduces a concept, 2) a lab that aims to give practical knowledge on the concept, 3) a mini-project in which the students explore more about the taught concept and finally 4) a seminar which provides the students with an opportunity to discuss and present what they learned. The training strategy is largely based on problem-based learning in which the lectures give an overview of the concepts, and using these lectures as a base, the students work in groups identifying what they need to know and find relevant information using the references in the lectures as a basis. The topics of the lectures were selected based on some of the fundamental challenges that software developers face when building monitoring and control applications for embedded systems. The major high-level challenges involved in remote monitoring and control that forms the basis for the design of this course is given in figure 3.
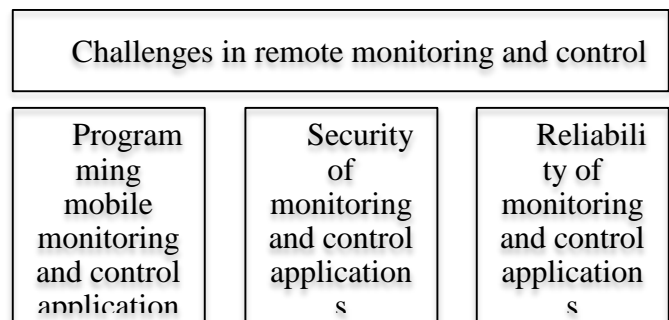

Figure 3 Major challenges in remote monitoring and control of embedded systems

In the following, we discuss each chapter in detail by outlining the main activities in the lectures, labs, mini-projects and seminars.

I. Chapter 1: Fundamentals of Remote Monitoring and Control

This section describes the first chapter of the MEDIS module that MDU is developing. The first chapter presents an introduction of the fundamentals of remote monitoring and control.

Lectures: This chapter will give an introduction to the fundamentals of remote monitoring and control of embedded systems as well as the liquid tank system, which will be used in the course. There will be 2 lectures covering this chapter, consisting of remote monitoring and control, and an introduction to the liquid tank system, as well as the communication between them through wireless technology.

At the end of the lecture, the students should understand:
- The different kinds of architectures that can be used for remote monitoring and control, such as
  - Direct communication
  - Communication through a web based application using a client-server approach
- The working principles of the liquid tank system
- The principles interfacing the liquid tank to a PC or a microcontroller board equipped with communication modules

Labs: The first lecture will implement 2 main laboratories that follow the structure of the lectures.

In the first lab, the students are required to write a two-page description of the parts of the water tank controller and explain how the components work. They should also be able to describe the controller in terms of its size and complexity and investigate methods to interface the water tank controller with mobile devices over a communication infrastructure based on Wi-Fi.

The second lab involves testing the water tank controller. In this lab the students are expected to get a clearer understanding of the controller and to be able to think of interfacing it with Android. Consideration of how to represent the pressure, the temperature and other components of the model is a must. The ideas should be discussed during the seminar. The students should also start with the initial steps such as installing Android Studio. The students are provided with a document that introduces them to the world of Android.

Seminars: In the seminar the students are expected to discuss the working of the water tank controller and its different aspects. They are also expected to discuss how to modify the water tank controller to e.g., randomly skip values transmitted through Wi-Fi to simulate faulty scenarios.

Mini-projects: The students are expected to read about the use of a water tank controller system as an example of a typical control system. They are expected to hand in a two-

page report that describes its basic principles associated with the control theory, along with some real world examples.

II. Chapter 2: Integrated Development Environments

This section describes the second chapter of the MEDIS module that Mälardalen University developed. The chapter presents an introduction to the mobile application development in Android, iOS, BlackBerry and Windows Mobile platforms. Moreover, it presents the cross platform development including PhoneGap, Appcelerator Titanium and Xamarin. The chapter is organized into two lectures as described in the following section.

Lectures: This chapter includes two lectures focusing on mobile application development platforms and cross platform development respectively [13]. The first lecture gives an overview of Android, iOS, BlackBerry and Windows Mobile platforms, their integrated features, dedicated Integrated Development Environment (IDE) and Software Development Kit (SDK) [14], [15], [16], [17]. The second lecture gives an overview of the cross platform development, then, describes a few frameworks for it such as PhoneGap, Appcelerator Titanium and Xamarin [18], [19], [20], [21].

At the end of Chapter 2, students are supposed to:

- Be familiar with dedicated platforms for mobile application development more specifically Andriod and iOS,
- Have the basic understanding of the cross platform development approach and PhoneGap, Appcelerator Titanium, Xamarin frameworks.

Labs: Chapter 2 includes 2 labs. The first lab is about installation of Android IDE and getting familiar with Android and Java programming through a basic Android application development. The second lab includes a brief introduction to the basic application development, getting familiar with the User Interface (UI) components and making the connection of the representation of the Water Tank Controller design with these components (widgets). This lab also includes exercises that are useful for the Water Tank Controller development. References and links for further reference are also provided.

Seminars: Students are required to make a short report of Android Studio features and its work-flow in the first seminar. In the second seminar, students are supposed to run the basic *HelloWorld* app in emulator and make a report of the steps they've taken and compare it when running the app on a real device.

Mini-projects: In the first Mini-project, students should make a report of Xcode, its features and interface builder using the links and references that are provided. In the second Mini-project, students are supposed to make a report of Rhodes

platform features and compare its features with platforms discussed in the second lecture of Chapter 2.

III. Chapter 3: Basic App Development

Chapter 3 covers inter-device communication for Android and iOS platforms within 2 lectures. The main focus of the lectures is on Bluetooth and Wi-Fi connectivities. The chapter is organized into two lectures as described in the following.

Lectures: This chapter covers inter-device communication in Android and iOS platforms within two lectures. The first lecture gives an overview of Android connectivity using Bluetooth, Near Field Communication (NFC) and Wi-Fi Peer-to-Peer (Wi-Fi P2P). Moreover, it presents Bluetooth communication major tasks and the steps towards developing a Wi-Fi P2P application. The second lecture starts with the networking overview in iOS followed by multi-peer connectivity. Multi-peer connectivity for iOS platform includes features, use cases and essentials phases for communication between devices.

At the end of the lectures, students will be able to

- Understand inter-device communication functionality for iOS and Android platforms,
- Develop simple apps in Android with inter-device communication functionality using the dedicated IDE or a cross platform development framework.

Labs: This chapter also includes 2 labs. In the first lab, students are supposed to create a broadcast receiver for Wi-Fi P2P intents in Android platform. Further in the second lab, their task is to create a Wi-Fi P2P application in Android. The useful material will be provided within each lab.

Seminars: This chapter has two seminars. In the first one, students will discuss the steps of creating P2P connections with Wi-Fi, e.g., setting up permissions and broadcast receiver, peer discovery.
In the second seminar, they should compare network service discovery vs. Service discovery using Wi-Fi P2P connection and discuss how they work and in what sense they differ from each other.

Mini-projects: This chapter also has 2 mini-projects. In mini-project 1, students should compare wireless connectivity in Android and iOS based on corresponding lectures. In mini-project 2, they should make a short report of Session Initiation Protocol (SIP) connectivity in Android and discuss how it works, the procedure to make and receive calls, the approach requirements and limitations.

IV. Chapter 4: Graphical User Interface

Chapter 4 includes two lectures and gives an overview of the Graphical User Interface (GUI) in general as well as the GUI for Android and iOS platforms.

Lectures: This chapter has 2 lectures. The first lecture covers GUI notion, major components and major paradigms in computer science. Moreover, it describes the basic GUI standards that are formed based on the contributing human factors [22].

The second lecture covers GUI for Android and iOS. Firstly, it presents an overview of the GUI in Android followed by *View* and *ViewGroup* description to build Android UI elements. It also describes Android layouts, defining the visual structure of a UI, attributes, ID and common layouts. Moreover, XML is described as an approach for layout declaration in Android. This lecture continues talking about the GUI in iOS, how to design a UI and define the interactions.

At the end of the lectures students will be able to:

- Understand the GUI in general and how to develop them efficiently in their applications
- Develop simple GUIs in their applications for Android platform

Labs: The chapter includes 2 labs. Lab 1 is about creating an Android layout in XML that includes a text field and a button. In lab 2, students will develop the first lab further such that when the button is pressed, it responds by sending the content of the text field to another activity.

Seminars: There are 2 seminars in Chapter 4. The first seminar is about action bar that is one of the most important design elements for the application's activities. Students are supposed to discuss the steps required to add action bar to their UI.

In the second lab within this chapter, students made a simple application that shows an activity (a single screen) with a text field and a button. In the second seminar, they should discuss how the application starts a new activity when the user clicks the Send button.

Mini-projects: This chapter includes 2 mini-projects. In mini-project 1, students should discuss the ways to intercept the events from user's interaction with an Android application. More specifically, they should discuss *Event Listeners* and *Event Handlers* interfaces.

In mini-project 2, they should discuss the steps to build a dynamic UI using Android fragments.

V. Chapter 5: Security

This chapter will cover the security related issues that a system connected to the internet might experience, suggest solutions and discuss the role-based access control approach. 2 Lectures will be conducted mainly focusing on security in mobile communication, security and control.

At the end of the lecture, the students will be able to:
- Understand the basics of security related issues
- Start implementing a simple role-based access control to their projects

Lectures: Lecture 1 in this chapter introduces computer security and the need for it in embedded systems. It outlines the template for a threat model [3][4] after positioning security in the big picture of dependability [2]. Later on, some of the most common computer security threats [5], [6], [7], [8], [9], [10] are discussed using the threat model discussed earlier, and a real world example where computer security was compromised, leading to significant losses, is discussed. Lecture 2 presents some of the ways to secure computer systems that are of interest to this course. Specifically, it discusses cryptography, virtual private networks and role-based access controls to secure computer systems. Further, the role-based access control mechanism is discussed in detail.

Labs: This chapter consists of two labs aimed at strengthening the fundamentals of students in the area of computer security. In the first lab, students use a famous open source web application called webgoat [11] that is widely used for training security professionals to actually exploit some of the threats discussed in the lectures. This will enable students to understand the intricacies involved in securing embedded systems. In the second lab, the students will implement role-based access control for the mobile application that controls the water tank controller.

Seminars: In the seminar corresponding to the first lecture, the students are expected to discuss the solutions of the lab. They are also required to discuss the possible strategies that could have been adopted to prevent a STUXNET like incident. In the seminar corresponding to the second lecture the students are expected to discuss the solutions for the lab. They are expected to write a one page report for each seminar summarizing the discussion as well as their thoughts and understandings.

Mini-projects: In the first mini-project, the students will write a 4 page report on how they can apply security concepts for embedded systems, e.g., merits, demerits, recent advances, open problems In the second mini-project, the students will perform a literature survey on encryption techniques, with particular focus on mobile devices and embedded systems, e.g., state of the art, impact on size, weight and power constraints, emerging challenges, open problems

VI. Chapter 6: Reliability

This chapter is on reliability of embedded systems and usage of fault tolerance and testing for dependable systems design. Three lectures will be conducted focusing on reliability, fault tolerance and testing of mobile applications.

At the end of the chapter, the students will be able to:
- Have an overall view of reliability and how it is related to security,
- Perform a simple reliability analysis using the project specification suggest a design for fault tolerance,
- Understand basic testing methodologies.

Lectures: The first lecture in this chapter introduces reliability starting from the big picture of dependability as well as positions it with respect to security. It then introduces some terminology and presents some case studies where the compromises of reliability of critical systems lead to catastrophes. It then introduces three commonly used techniques to improve reliability namely: fault tolerance, testing and safety analysis. Fault tolerance and testing are introduced in detail in the second and third lecture respectively. The second lecture presents one of the ways to improve reliability of computer systems, namely using fault tolerance. This lecture focuses on fault tolerance using redundancy. The most common types of redundancy viz. temporal redundancy and spatial redundancy are outlined. Additionally, the concept of N-version programming that is central to the concept of redundancy is explained. Finally, the challenges involved in improving reliability of mobile applications are outlined. The last lecture in this chapter deals with software testing as a means to improve reliability. The students are introduced to different software development life cycle (SDLC) models. They are then given an overview of testing in the big picture of different SDLCs. Specific SDLC models that give importance to testing are outlined. Further they are introduced to different phases/types of testing, after which they are given an overview of a test automation framework. Finally, unit testing is introduced to the students with a short example on how this is done.

Labs: This chapter consists of three labs aimed at strengthening the fundamentals of students in the area of reliability. In the first lab, students modify the water tank controller application such that it simulates an erroneous behaviour e.g., by dropping messages randomly. They will collect data representing the behaviour of the water tank controller for some number of simulations. In the second lab, the students will implement temporal and spatial redundancy and collect data representing the behaviour of the water tank controller application under both the strategies. In the third lab, the students will test their mobile application using unit testing.

Seminars: In the first seminar, the students will read and discuss seminal research articles and write a short summary along with their reflections. 1) N. G. Leveson, "High-pressure

steam engines and computer software," in Proceedings of the 14th International Conference on Software Engineering, 1992 2) M, Jones, What really happened on Mars? 3) N. G. Leveson, The role of software in spacecraft accidents.

In the second seminar, the students will discuss their lab in the class, e.g., type of redundancy more suitable for mobile applications, effects of redundancy on security, design of reliable applications.

In the final seminar, the students will read and discuss the following, and write a short report along with their reflections: 1) On the Effectiveness of Test-first Approach to Programming Erdogmus, H, IEEE Transactions on Software Engineering, 2005 2) An Analysis and Survey of the Development of Mutation Testing , Yue and Mark Harman, IEEE Transactions on Software Engineering, 2011.

Mini-projects: In the first mini-project, the students will submit a report summarizing the articles and the discussions during the seminar. The report should include their reflections on the reliability aspects of mobile applications that control embedded systems.

In the second mini-project, the students will summarize the following articles, while also includes their reflections about the same [?] (this projects aims to complement the second lab discussed above): 1) Algirdas A. Avizienis, The methodology of N version programming. 2) J.C Knight and N.G. Leveson, An experimental evaluation of the assumption of independence in multiversion programming, IEEE Transactions on Software Engineering. 3) A A. Avizienis, M R Lyu, and W Schutz, In Search of Effective Diversity: A Six-Language Study of Fault-Tolerant Control Software. 4) J.C Knight and N.G. Leveson, A reply to the criticisms of the Knight & Leveson experiment, ACM SIGSOFT Software Engineering Notes.

As the final mini-project in this chapter, the students will form groups of two, and run the test cases that they developed earlier in the lab on each others' code and write a report summarizing the results while also discussing the results in the context of N-version programming, e.g., did the two code fail on the same test case, is N-version programming effective?

VII.   Chapter 7: Research in remote monitoring and control

This chapter is specially dedicated to the students, as a part of the problem based learning strategy, to further explore some specific problem/research area and present their findings.

Lecture: The lectures in this chapter will be handled by the students who will explore some specific research topic, identify seminal papers and present the research to the entire class. The starting point for the students to find topics is given in the form of references.

Labs: The labs are allocated to those students who have not completed their previous assignments or want to work on some of the research findings that they made during the preparation of the lectures.

Seminars: The seminar session is allocated to the students who have not completed their previous seminars or to those who want to discuss the topics they have presented in the lectures.

Mini-project: Here again, the slot is allocated to students to finish off any incomplete mini-project or to those who want to do more research to further, e.g., publish their work.

IV.   CONCLUSION AND FUTURE WORK

In this paper, we present the design of, and the rationale behind, a learning module for control of embedded systems using mobile applications that we developed under the aegis of the EU-TEMPUS project called MEDIS. The choice of the course structure has been largely driven by the Problem Based Learning (PBL) approach in which students, based on the problems that they identify, drive the learning. The course, which involves a set of hand picked chapters, each containing a lecture, lab, seminar and a mini-project, was specifically chosen to support PBL by giving students more opportunities to explore topics that interest them. The main challenge that we faced was to strike a balance between the possible maximum course credits, which was fixed, and the workload. The main challenge that this course faces is related to the integration of this course to the existing curricula. Different universities may need to place this course at different points in the programme depending on the background of the students. Universities can provide this course as a foundation level course with additional classes in programming and networking, while others may give it as an advanced level course by discussing more advanced topics in security and reliability

Future work involves improving the course after getting feedback from the implementation phase as well as including more exercises, *e.g.,* in the context of next generation cyber-physical systems.

REFERENCES

[1]   HTTP://SEARCHENGINELAND.COM/ANALYST-MOBILE-TO-OVERTAKE-PC-FOR-LOCAL-SEARCH-BY-2015-119148

[2]   Basic concepts and taxonomy of dependable and secure computing, Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C., IEEE Transactions on Dependable and Secure Computing, 2004

[3]   Threat Modeling: A Process To Ensure ApplicationSecurity,http://www.sans.org/reading-room/whitepapers/securecode/threat-modeling-process-ensure-application-security-1646

[4]   Template Sample: Web Application Threat Model, https://msdn.microsoft.com/en-us/library/ff649779.aspx

[5]   Understanding Denial-of-Service Attacks (Security Tip (ST04-015)), https://www.us-cert.gov/ncas/tips/ST04-015

[6]   Anoymous,

https://en.wikipedia.org/wiki/Anonymous_%28group%29

[7]   The   Real   Story   of   Stuxnet,
http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet/

[8]   Denial-of-Service   Attacks,
http://www.cert.org/historical/advisories/ca-1997-28.cfm

[9]   SQL   Injection,   https://technet.microsoft.com/en-us/library/ms161953%28v=SQL.105%29.aspx

[10] Social   Engineering: Manipulating   the   Source,
http://www.sans.org/reading-room/whitepapers/engineering/social-engineering-manipulating-the-source-32914

[11]   Webgoat

https://www.owasp.org/index.php/WebGoat_Installation

[12] H. Hassan, JM. Martínez, C. Domínguez, A. Perles, J. Albaladejo: Innovative Methodology to Improve the Quality of Electronic Engineering Formation through Teaching Industrial Computer Engineering, IEEE Transactions on Education Vol: 47, Nº 4, pp: 446-452, 2004.

[13] Allen, S. and Graupera, V. and Lundrigan, L.: Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution

[14] Android, https://developer.android.com/index.html

[15] iOS, http://developer.apple.com/iphone

[16] BlackBerry, http://na.blackberry.com/eng/developers/

[17] Windows Mobile, https://dev.windows.com/en-us/getstarted

[18] PhoneGap, http://www.phonegap.com/

[19] PhoneGap, http://wiki.phonegap.com/Roadmap

[20] Appcelerator Titanium, http://www.appcelerator.com

[21] Xamarin, http://xamarin.com/

[22] Moore, A. and Redmond-Pyle, D.: Graphical User Interface Design and Evaluation: A Practical Process

A learning module for simulation based programming of microcontrollers within production systems

M. Seyfarth

Universitat of Stuttgart, Stuttgart, Germany

Abstract—**This paper describes the conception of a hardware-in-the-loop simulation system based on a microcontroller platform. It is used for teaching highly qualified engineers in the design and development of advanced industrial informatics systems.**

Index Terms—**Microcontroller, Hardware in the Loop Simulation, Teaching module**

I. INTRODUCTION

Microcontrollers control various automated systems and machines. Their application field ranges from the control of consumer goods (coffee machine, washing machine, consumer electronics, home automation,…) to control devices in automobiles (ABS, airbag,…) or drive controllers up to controlling simple industrial automated processes (mixing units, conveyor systems,…). In these cases, microcontrollers are adapted to each application concerning performance and equipment and, compared to industrial PC and PLCs are very cost-effective. In addition, the electrical power consumption is lower than that of conventional control systems.

Within the EU project MEDIS („A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics Systems") a learning module is to be created, which introduces application and programming of microcontrollers in the area of advanced and distributed industrial informatics systems (AIISM) to the students. A problem-based learning (PBL) methodology will be used [1,2], which will train highly qualified engineers.

II. MICROCONTROLLERS AND CONTROL OF INDUSTRIAL PROCESSES

An industrial process is a set of interacting operations in a technical system, by which substances, energy or information are transformed or stored [3] (Fig. 1).
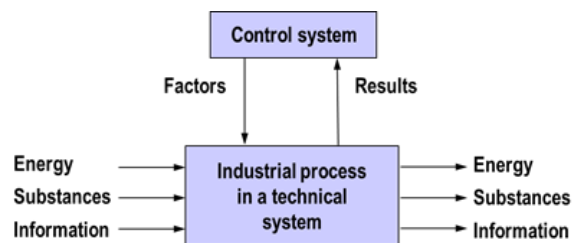

Fig. 1. Control of industrial process

Operations in industrial processes can be divided into three types:

*   Continuous or dynamic operations (e.g. chipping process during turning in a machine tool). These operations are modelled by differential equations or transfer functions.
*   Sequential or discrete event type operations (e.g. a production sequence like "Clamp raw material", "Move tail stock", …). These operations are modelled by flow diagrams, state machines or petri nets.
*   Discrete object type operations (e.g. transport of raw material on a conveyor). These operations are modelled by queue models, graphs or petri nets.

A control system affects a technical process so that a desired goal is reached [3] (Fig. 1). There are two different types of processes:

*   Control process [3] means the selective influence on output variables through input variables of the system. Indicative of a control process is the open line of impact. Only impact of known disturbances can be compensated.
*   Closed loop control process [3] means the continuous measurement of a variable, the comparison with a reference variable and influencing of the system in the sense of alignment to the reference variable. Indicative for the closed loop control is the closed line of impact, in

which the variables to be controlled affect continuously themselves. A closed loop control process can get instable, but the impact of unpredictable disturbances and parameter changes in the system can be compensated.

The overall structure of a control system is shown in Fig. 2. Sensors monitor the industrial process. They usually convert physical values (e.g. position, temperature, …) into electrical signals. An information processing system (e.g. controller) collects the measurement data, computes the necessary actions, in order to achieve the desired goals, visualizes data to a human and collects manipulations of a human (e.g. button pushed). By the conversion of control signals, actuators influence the industrial process (e.g. open a valve, stop a motor, …).
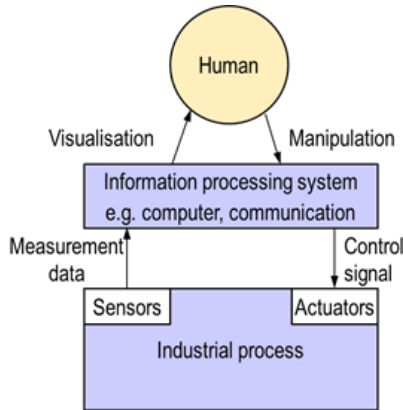

Fig. 2. Structure of a control system

If the control system is a digital microcontroller, the measurement data and control signals are computed in a time-discrete way. The values of the signals are digitized and are computed value-discretely.

There is a wide range of different microcontrollers. Essentially, they differ in processing speed, available memory, available interfaces, vendor and price. A microcontroller is a single chip that contains the processor (the CPU), non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock, an I/O controller unit and other periphery (e.g. timer, counter, analog digital converter, …) [4] (Fig. 3).
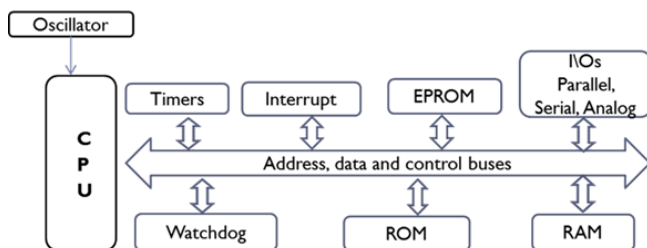

Fig. 3. Architecture of a microcontroller [wiki.zimt.uni-siegen.de]

The learning module in the MEDIS project is based on an "ARDUINO Due" microcontroller. It is a widespread, open source, easy-to-use, powerful and economical microcontroller board using an Atmel SAM3X8E ARM Cortex-M3 CPU [5]. The microcontroller is programmed in the computer language "C" or "C++", provides 54 digital Input/Output pins, 12 analog inputs, 2 analog outputs and 512 KB of flash memory.

III. HARDWARE OF SIMULATION ENVIRONMENT

The following steps are needed for learning how to control an industrial process by means of a microcontroller:

- Establishing a proper understanding in theory of industrial processes and programming of microcontrollers
- Practical training using application examples with increasing complexity.

Especially the training with practical applications serves for a deeper understanding of programming microcontrollers.

The basic idea of the learning module is the provision of a hardware-in-the-loop simulation (HILS) [6,7] for the application examples. This means that the real microcontroller hardware, which is used as control system for the industrial process (control microcontroller), is used. In order to avoid cost for real test stands for the application examples, simulation models are used. These reproduce the behavior of the industrial process. The simulation models are run on a second microcontroller platform (simulation microcontroller).
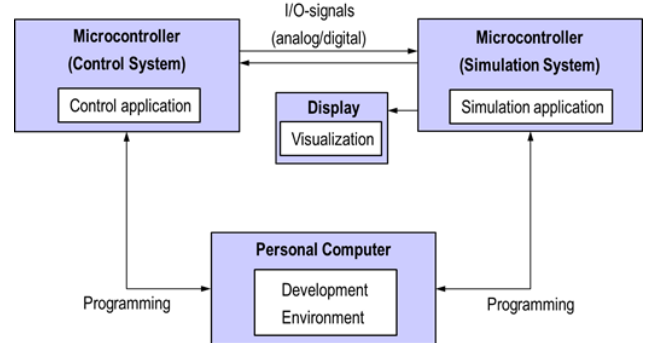

Fig. 4. Hardware Structure of HILS environment

Fig. 4 shows the hardware structure of the HILS environment. One microcontroller performs the simulation application. This includes the temporal behavior and the sensors and actuators of the industrial process. To access the simulation model, the I/O-system of the microcontroller is used. For example, the simulation microcontroller sets an analog output in order to indicate that an analog sensor (e.g. temperature sensor) has a certain value. On the other hand, by activating a digital input an actuator can be switched on (e.g. pump motor).

Activating and reading out of the I/O system of the simulation microcontroller is performed by the control microcontroller, which is programmed by the students for controlling the industrial process. For example, the control microcontroller sets a digital output connected to an input of the simulation microcontroller to start a pump motor in the simulation model. This means that instead of the amplifier circuit of a real pump motor the output of the control microcontroller activates the input of the simulation microcontroller, which starts the pump algorithm in its behavior model.

During the electro-technical connection of the microcontrollers, the following aspects need to be considered:

- A digital input can be connected directly with a digital output, or vice versa, if both microcontrollers operate at the same voltage level. Otherwise, a converter circuit with an operational amplifier per digital in- and output pair must be provided. In order to protect the digital inputs and outputs against wiring errors of the students and short circuit currents, a resistor must be interposed, limiting the current flowing to 2,2 mA (see data sheet of "ARDUINO Due" [8]).

- An analog output can be connected directly with an analog input, if the provided voltage levels are identical. Most microcontrollers, however, have no or only very few direct analog outputs (e.g. "ARDUINO Due" has only two), so that an external digital analog converter is used (Linear Technology LTC2620 [9]). The microcontroller controls this module via the SPI-bus. Voltage adjustment can also be achieved with this module.

- A pseudo-analog output (pulse-width modulation) cannot be connected directly to an analog input. The analog input of a microcontroller operates with an analog-digital converter, which is not able to convert the PWM signal correctly. Therefore, the PWM signal must be converted via a passive low-pass filter (of 2nd order) into an almost smooth analog voltage ([10]).

All components mentioned above are mounted on an adaptor board, which allows also an individual wiring between the two microcontrollers by using a breadboard (Fig. 5).
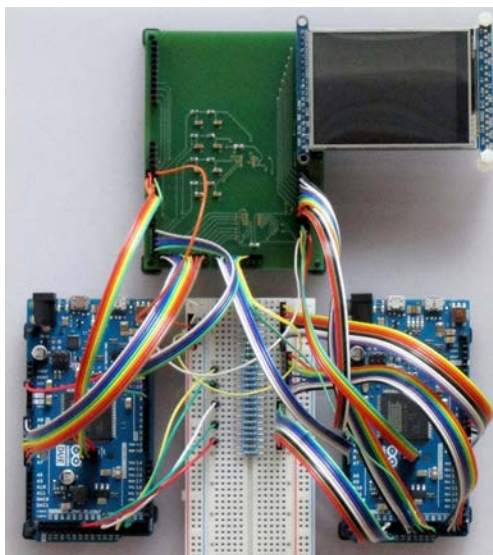

Fig. 5. Hardware Design

The graphical display (upper right corner in Fig. 5) visualizes the simulation model. For example, the filling level of a tank, the temperature of a liquid, the movement of a parcel on a conveyor, …

Altogether the controller hardware and all additional components are mounted on a Plexiglas disk (200x300 mm2), so that it is a compact, well ordered and easy to transport test

rig. The power supply is provided by the personal computer with a USB cable, which is also used for programming the microcontrollers.

IV. SOFTWARE OF SIMULATION ENVIRONMENT

The simulation model must behave like a real industrial process. The microcontroller for the control system should not notice that it only controls a simulation model. When selecting the simulation models, increased emphasis was put on practical industrial processes. The two simulation models presented in the following are taken from the process engineering field (liquid tank) and from the production-technological field (material flow system).

The simulation models run on the simulation microcontroller being controlled by the control microcontroller. The industrial process is considered here as a closed system that can be affected via defined interfaces (sensors and actuators, see Fig. 2).

A. Simulation model of a liquid tank system

In the chemical industry, but also in food production, liquids must be mixed according to given recipes and after that bottled. The liquids are stored most of the time in large tanks and flow, through pipes and valves, into other tanks for the mixing process. There, the liquids can be tempered and actively mixed by agitators. The process parameters (e.g. amount of liquid, temperature) must be monitored continuously.

Because of its practical relevance, the good comprehensibility, the manageable complexity and the modular extensibility, this industrial process is most suitable for learning the programming of control systems based on microcontrollers.

The industrial process "Liquid tank system", in general, contains different tanks, their interconnection by pipes, sensors for measuring the filling level and temperature of the liquid, actuators for filling the tank (pump, valve), mixing and heating the liquid (Fig. 6).
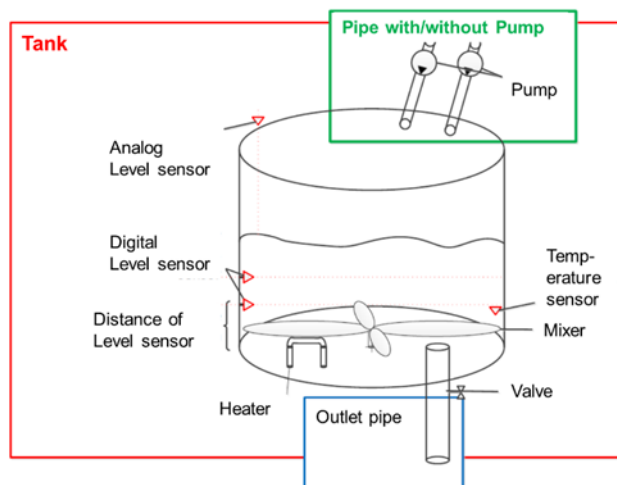

Fig. 6. Objects in liquid tank system

The simulation model is based on an object-oriented approach. Objects can be instantiated multiple times, depending on the desired application. The class hierarchy is shown in Fig. 7.

Class Tank: This is the main class of the simulation model. It describes a single tank with its dimensions: radius and height. It stores the information about the amount of liquid in the tank and its temperature. It recalculates these values depending on the liquid flow and heater power.

Class Analog Temperature Sensor: This class models an analog temperature sensor, which measures the temperature of the liquid in the tank. It can be instantiated multiple times. A real temperature sensor produces an analog voltage signal, in proportion to the temperature. One important attribute of this sensor is the identification of the output pin, where the sensor sends his current temperature value, coded as a specific voltage level. An analog input pin of the microcontroller for the control system must be wired to this output pin of the microcontroller for the simulation system.

Class Analog Filling Level Sensor: This class models an analog filling level sensor, e.g. an ultrasonic distance sensor, which measures the distance to the surface of the liquid. The analog voltage signal describes the measured distance and is sent to an analog output pin of the microcontroller for the simulation system. There can exist multiple instances of this class.

Class Digital Filling Level Sensor: this class models a digital filling level sensor, e.g. an optical sensor, which produces a digital signal when under water. This digital signal is sent to a digital output pin of the microcontroller for the simulation system. Multiple instances of this object can be created at different heights of the tank system.
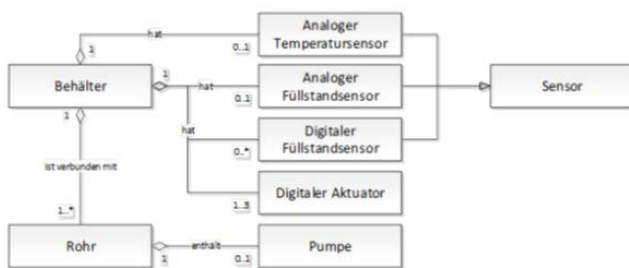

Fig. 7. Class diagram of tank model

Class Digital Actuator: this class models all digital actuators of the tank system: mixer, valve and heater. There is a digital input pin on the microcontroller for the simulation system, which activates the object. E.g. if the input signal is "1" the valve is open. Multiple instances of different actuator types can be created for the simulation model.

Class Pump: this class models a pump. The object is actuated by a pulse width modulated digital signal, to determine the number of revolutions per minute of the pump between 0…100%. This results in a corresponding flow of liquid, depending on the maximum output flow of the pump. In the simulation model the pump is connected to an analog

input pin of the microcontroller. A pump is always assigned to one pipe object.

Class Pipe: this class models pipes between different tanks or pipes to fill the tank. A pipe object is characterized by its length and diameter (to calculate the flow), and the identification of the tanks which are connected by this pipe.

All classes are coded in C++ language for the microcontroller Arduino DUE. Each object in addition contains a visualization, to display its state on the liquid crystal display. The configuration of a specific tank system is done in the C++ code directly by constructors and parameter lists. A detailed example is given in chapter V.

B. Simulation model of material flow system

The control and organization of the material flow in production and assembly plants are essential parts of these systems. Workpieces must be transported from the warehouse via conveyor lines to different processing stations until they are transferred to another storage place as finished products. Examples of processing stations are milling, drilling, painting and assembly stations. The processing stations are connected by conveyor lines with branches.

This industrial process is very apt for learning the programming of control systems based on a microcontroller, because it is practical, easily comprehensible, not so complex and it also has a modular extensibility and configurability.

The industrial process „Material Flow System" in general contains different processing stations, stocks, their interconnection by conveyor belts with branches, sensors for measuring workpieces (e.g. position, weight, dimension, color, …) (Fig. 8).
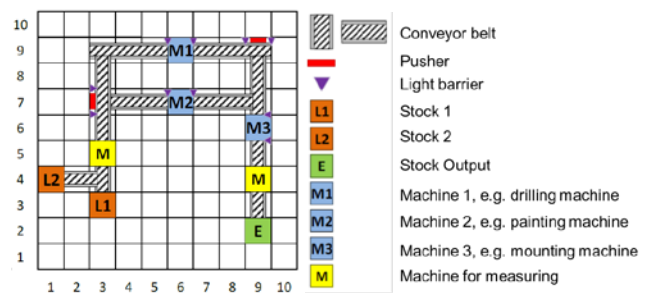

Fig. 8. Objects in material flow system, example of configuration

The simulation model is based on an object-oriented approach. Objects can be instantiated multiple times, depending on the desired application. The class hierarchy is shown in Fig. 9.

Class Production Line: This is the main class of the simulation model. It combines all objects of the configured production line and calculates the dynamic behavior of all objects. All objects must be placed on a 10x10 grid, and conveyor belts can only be placed horizontally or vertically.

Class Conveyor Belt: This class models one conveyor belt. Multiple instances of this class can be created. A conveyor

belt transports workpieces through the production line. A conveyor belt is actuated by a pulse-width modulated motor. The motor is controlled by an analog input pin (giving the speed) and a digital pin for the direction of movement. A pulse-width modulated digital output pin and a digital output pin of the microcontroller for the control system must be wired to the input pins of the microcontroller for the simulation system.

Class Pusher: This class models a pusher, which is mounted at a branch of the conveyor belts. The pusher is, for example, actuated by a pneumatic cylinder and pushes a workpiece ahead of itself to the next conveyor belt. If the pusher is activated too early or too late, the workpiece will fall off the conveyor belt. The pusher is controlled by one digital input pin of the microcontroller for the simulation system.

Class Stock: This class models a stock of workpieces. One stock can contain different types of workpieces. If the digital input pin of the stock is activated, it pushes one workpiece to the conveyor belt.

Class Stock Output: At the end of the production line there must be one object of this class. It visualizes the number of workpieces on the liquid crystal display that reached this position of the production line. It deletes workpieces from the internal software database.

Class Light Barrier: This class models a light barrier mounted to a conveyor belt at any position. Multiple objects of this class can be instantiated. The light barrier activates a digital output pin of the microcontroller of the simulation system if a workpiece is within reach of the light barrier.

Class Machine: This class models different types of machines. Currently, there are three types of machines implemented: Drilling machine, painting machine, mounting machine. Each machine object delivers two digital sensor values if a workpiece enters or leaves the machine. Each machine object is activated by a digital signal, which starts the operation of the machine. A parameter configures the processing time for one workpiece.

- Drilling machine: this machine drills holes into the middle of the workpiece.
- Painting machine: this machine colors a workpiece. The desired color is given to the machine by an analog input signal.
- Mounting machine: this machine mounts a round workpiece on the hole of a quadratic workpiece (see class Workpiece)
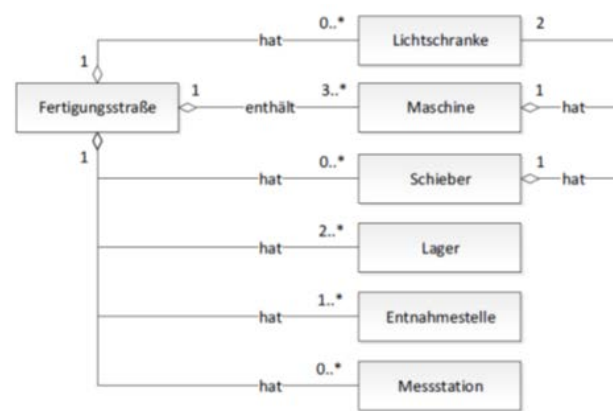

Fig. 9. Class diagram of material flow model

Class Measuring station: This class models a measuring machine in order to identify the properties of a workpiece. There are two digital output signals and one analog output signal. The meaning of the signals is described in the Class Workpiece.

Class Workpiece: This class is not shown in Figure 9, but is fundamental for the simulation model. Each object of this class represents one workpiece. Currently, two types of workpieces have been implemented: A cylinder and a cube (can be identified by the measuring machine, which is signaled by a digital output of the measuring machine). Each workpiece can be colored (by the painting machine) and the color can be identified by the measuring machine (analog output signal coding different colors). The drilling machine can drill the cubic workpieces and the measuring machine can identify this drill hole (digital output signal of the measuring machine) [10].

All classes are coded in C++ language for the microcontroller "ARDUINO Due". In addition, each object contains a visualization to display its state on the liquid crystal display. The configuration of a specific material flow system is done in C++ code directly by constructors and parameter lists.

V. Application and Testing

This chapter shows an example of an application for a three tank system (see Fig. 10). The system consists of three liquid tanks. Two basic liquids must be mixed and stirred in a third tank at a defined temperature, before the end product can be released via a valve.

The following table shows the sensors and actuators in the individual tanks as well as their connection to the input and output pins of the microcontroller for the simulation system.

TABLE I
LIST OF SENSORS AND ACTUATORS FOR THE THREE TANK SYSTEM

Tank	Sensor/Actuator	Pin Number
1	Ultrasonic sensor for filling level	AO0
1	Temperature sensor	AO1
1	Mixer	DI15
1	Input pipe with pump	AI5
2	Ultrasonic sensor for filling level	AO2
2	Temperature sensor	AO3
2	Heater	DI17
2	Input pipe with pump	AI6
3	Ultrasonic sensor for filling level	AO4
3	Temperature sensor	AO5
3	Heater	DI24
3	Mixer	DI25

AOx: analog output, pin x, Dix: digital input, pin x

All tanks have the same geometrical dimensions: Radius 100 millimeters, height 500 millimeters and they are all empty in the beginning. All outlet pipes have a diameter of 20 millimeters.

To instantiate three tank objects in the simulation model, the following C-code is necessary:

```
#define amount_container = 3;
ContainerClass Container[amount_container] = {
ContainerClass(100,500,0,10,0,0,0),
ContainerClass(100,500,0,10,1,screen.height()/2,0)
,
ContainerClass(100,500,0,10,2,0,screen.width()/2)
};
// ContainerClass(radius, height, initial filling,
radius outlet pipe, tank ID, X-position, Y-position)
```

Pipes connect the tanks to each other. Fig. 10 shows five pipes. For the simulation model, four of them have to be instantiated. The outlet pipe of tank 3 is not needed, because there is no connection to another tank:

```
#define amount_branches = 4;
BranchClass Branches[amount_branches] = {
//Branches before any container
BranchClass(20,1,255,0,0,0,0),
BranchClass(20,1,255,1,0,1,0),
//Branches between containers
BranchClass(20,1,0  ,2,0,2,0),
BranchClass(20,1,1  ,3,0,2,0)
};
// BranchClass(diameter, length, start tank ID
(255=none), own ID, initial flow rate, end tank ID,
initial temperature)
```
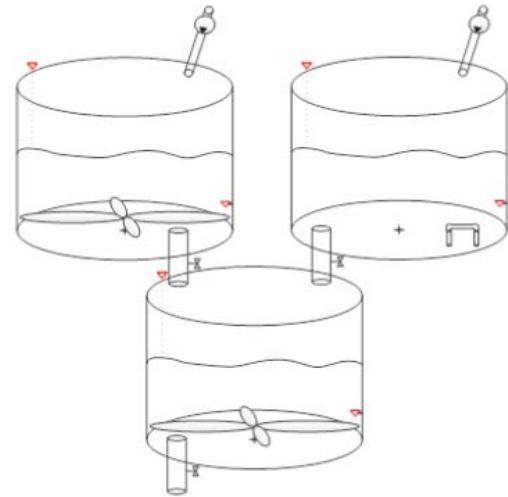

Fig. 10. Configuration of simulation model for three tanks

In step three all actuators are instantiated. The information from table I concerning the ID of the input/output pins must be considered:

```
#define amount_digital_actuators = 7;
Digital_ActuatorClass
Digital_Actuators[amount_digital_actuators] = {
Digital_ActuatorClass(0,1,false,15,0),
Digital_ActuatorClass(0,2,false,16,0),
Digital_ActuatorClass(1,0,false,17,1000),
Digital_ActuatorClass(1,2,false,19,0),
Digital_ActuatorClass(2,0,false,24,1000),
Digital_ActuatorClass(2,1,false,25,0),
Digital_ActuatorClass(2,2,false,26,0)
};
// Digital_ActuatorClass(tank ID, type (1=heater,
2=mixer, 3=outlet valve), initial state, digital
pin, power of heater)

#define amount_pump_motors = 2;
Pump_motorClass Pump_motors[amount_pump_motors] =
{
Pump_motorClass(5000,0,0,0,5,50),
Pump_motorClass(5000,0,0,1,6,20)
};
// Pump_motorClass(max flow rate, initial status,
current flow rate, pipe ID, analog pin, incomimg
liquid temperature)
```

The process must be monitored. Sensors measure the filling level of each tank and temperature sensors measure the temperature of the liquid in each tank. The information from table I concerning the ID of the input/output pins must be considered:

```
#define amount_analog_temp_sensors = 3;
Analog_temperature_sensorClass
Analog_temperature_sensors[amount_analog_temp_sensor
s] = {
Analog_temperature_sensorClass(0,8,0,1),
Analog_temperature_sensorClass(0,8,1,3),
Analog_temperature_sensorClass(0,8,2,5)
};
//       Analog_temperature_sensorClass(initial
temperature, resolution in bits, tank ID, DAC output
pin)
```

```
  #define amount_analog_fluid_sensors = 3;
  Analog_fluid_level_sensorClass
Analog_fluid_level_sensors[amount_analog_fluid_senso
rs] = {
  Analog_fluid_level_sensorClass(0,8,0,0),
  Analog_fluid_level_sensorClass(0,8,1,2),
  Analog_fluid_level_sensorClass(0,8,2,4)
  };
  // Analog_fluid_level_sensorClass(initial filling
level, resolution in bits, tank ID, DAC output pin)
```

These objects define the configuration of the simulation system. The simulation system visualizes the tanks, pipes, sensors and actuators on the liquid crystal display and calculates the liquid flow, the filling levels and the temperature depending on the state of the actuators. The microcontroller for the control system must be connected to the input and output pins of the microcontroller for the simulation system (as described in chapter III). A control application must be implemented on the microcontroller for the control system.

A typical industrial process for this three-tank system could be:

1. Fill tank 1 and 2 up to a filling level of 50%. The liquid in tank 1 must be stirred permanently, the liquid in tank 2 must be heated to 50°C.
2. Mix both liquids in tank 3; for that, first fill liquid 1 into tank 3 up to a level of 20%, after that fill liquid 2 into tank 3 up to a level of 30%, while stirring the mixture in tank 3.
3. Heat the liquid up to 50°C while stirring it.
4. Empty tank 3.
5. Mix another portion in tank 3 with the same steps.

The sequence of events of the industrial process can be monitored on the liquid display. The student sees the consequences of his control program directly.

VI. CONCLUSION

The learning module enables students to learn the programming of microcontrollers in the area of industrial processes. The proposed learning system contains a simulation model for a liquid tank system and a material flow system supplemented by a control system based on a microcontroller. The advantage of this hardware-in-the-loop simulation system is its flexibility, modularity and cost efficiency. Different industrial processes can easily be modelled.

REFERENCES

[1] H. Hassan, J. Martínez, C. Domínguez, A. Perles, J. Albaladejo: Innovative Methodology to Improve the Quality of Electronic Engineering Formation through Teaching Industrial Computer Engineering, IEEE Transactions on Education Vol: 47, Nº 4, pp: 446-452, 2004.
[2] H. Hassan, J. Martínez, C. Domínguez, A. Perles, J. Albaladejo, JV. Capella: Integrated Multicourse Project Based Learning in Electronic Engineering, International Journal of Engineering Education, Vol. 24, Nº 3, pp: 581-591, 2008.
[3] N.N.: Norm DIN IEC 60050-351: International electrotechnical vocabulary - Part 351: Control technology.
[4] U. Brinkschulte, T. Ungerer: Microcontroller und Mikroprozessoren, Springer, Berlin 2002.
[5] N.N.: Arduino website. http://www.arduino.cc (20.04.2015).
[6] S. Röck: Hardware in the Loop Simulation of Production Systems Dynamics, Journal of Production Engineering Research & Development, Volume 5, Issue 3 (2011), Page 329, Springer 2011.
[7] S. Röck: Echtzeitsimulation von Produktionsanlagen mit realen Steuerungssystemen, Jost- Jetter Verlag Heimsheim, 2007.
[8] N.N.: Atmel ATSAM3X8E Cortex-M3 MCU datasheet. http://www.atmel.com/devices/sam3x8e.aspx (22.04.2015).
[9] N.N.: LTC2620 – Octal 12-Bit Rail-to-Rail DAC. http://www.linear.com/product/LTC2620 (22.04.2015).
[10] B. Molter, P. Ockert: Konzeption und Entwicklung einer Simulationsumgebung für Mikrocontroller, Studienarbeit, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW), Universität Stuttgart, Stuttgart 2014.

Dipl.-Ing. M. Seyfarth finished his degree as "Diplomingenieur Elektrotechnik" at Friedrich-Alexander University Erlangen, Germany in 1994. One of his specialization fields was Closed Loop Control.

He works at the Institute for Control Engineering for Machine Tools and Manufacturing Units since 1996. His fields of research are Open Control systems and Reconfigurable Machine Systems. Mr. Seyfarth is course manager of the Mechatronics courses of University Stuttgart and teaching in the area of Control Systems and Fluidics.

Implementation methodology for the formation of highly qualified engineers on the information systems department at Al-Farabi Kazakh National University

Zh. Yessengaliyeva, Sh. Mussiraliyeva

Al-Farabi Kazakh National University, Almaty, Kazakhstan

Abstract—

Index

I.   INTRODUCTION

L atest requirements in production, business and society have a strong influence on teaching methods and content of engineering education today. Modern engineering projects cover technical, but also management, social, economic, and many other aspects that significantly affect their complexity. Specificity of tools and mechanisms for formation of a new quality in a competence-based education system requires a careful study of Kazakh and foreign experience in innovative education as well as ways of introducing them into traditional educational system. The problem of training's quality is a central issue in their demand for domestic economy and international recognition, which is directly related to the content of education and implementation technology of educational programs. Nowadays, throughout the world there is an active transformation of engineering education aimed at developing competencies required by professionals, in particular at ability of their application in practice.

Al-Farabi Kazakh National University is the leading institution of the system of higher education of the Republic of Kazakhstan. The university consists of 14 functioning faculties, 98 departments, 20 scientific research institutes and centers, a techno-park; more than 2 000 professors, doctors, scientific candidates and PhD's, more than 100 academicians of the largest academies, about 30 honored figures of the Republic of Kazakhstan, more than 30 laureates of State and nominal awards of RK and 40 laureates of the young scientists' awards, 45 fellows of state scientific fellowships. More than 18 000 students and masters at the multilevel system of higher professional education study in the university. We cooperate with 418 largest international universities of the world on the realization of joint international educational programs, exchange programs for students and internship.

The Information Systems Department of KazNU named al-Farabi is committed to providing undergraduate and graduate students with the knowledge and skills required to plan, develop, and deploy technology-based business solutions. Students are equipped with a solid understanding of the strategic role of information systems in organizations and the influential role of technology in society. The department trains specialists in the field of information systems, recruits to the specialty:

Bachelor's Degree

6B070300–Information Systems;

6B070200– Automation and Control;

Master's Degree

6M070300– Information Systems;

6M100200–Information Security Systems;

6M070200– Automation and Control;

Ph.D.

6D070300– Information Systems,

6D100200– Information Security Systems,

6D070200– Automation and Control.

Teachers of IS department are taking part in the three educational TEMPUS projects now.

II.   TEMPUS PICTET

Project is designed for strengthening of the links between education and ICT business by enhancing the RU-KZ system of professional ICT-training by using EQF, building effective System of professional ICT-Education, setting up Network of ICT-training centers in RU/KZ universities for enforcing infrastructure and framework. Wider objective is the

improvement of the quality of ICT-specialists preparation by building an effective System of professional ICT-education and strengthening links with business environment in RU and KZ.

The project specific objectives:
• To create e-qualifications framework for RU and KZ
• To establish a network of ICT-training centers in PC universities
• To create the new methodology of professional ICT-training
• To elaborate new curricula for ICT-qualifications and introduce them into practice of professional training, providing with teaching materials and learning environment
• To develop a quality assurance mechanisms ensuring the quality of professional training in ICT
• Dissemination of the System of professional ICT-education, new methodology, teaching materials and methods among ICT-educators and ICT-business.

III. TEMPUS PROMIS

PROMIS stands for PROfessional network of Master's degrees in Informatics as a Second Competence.

Project's main Goals:
• Extension of ERAMIS network in Central Asia
• Improvement of the academic quality of the network
• Improvement of the vocational focus of the master degrees in computer sciences in different ways
• Establishment of strong relationships between companies and universities
• Adapting teaching process to the students working in parallel of the studies
• Improvement of the network by sharing courses and pedagogic material

IV. TEMPUS MEDIS

In winter of 2014 an international project MEDIS (A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics) started as part of TEMPUS IV - 6th Call for proposals program. Given the above, this project is an excellent opportunity to adopt innovative teaching methods of European universities for quality education of engineering students. MEDIS project aims to adapt master training programs for engineering specialties from EU partner universities by including in programs the Advanced Industrial Informatics Specialization Module (AIISM) aimed at training professionals in engineering design and development of industrial information systems using microcomputers, industrial computers, mobile and cloud computing platforms for decentralized control and management of complex processes. Such courses will contribute to preparation of highly skilled engineers who will be able integrate easily into labor market. Moreover, involvement of such engineers in production will help increase productivity and competitiveness of companies, thus contributing to development of the whole society [1].

The objectives of the project are:
• offer PBL (problem-based learning) methodology and develop resources for teaching the proposed modern industrial informatics specialization module (AIISM);
• review the curricula of universities in partner countries and adapt AIISM in order to integrate it into specific training programs in each partner country;
• develop training courses and train teachers, support and administrative staff in the partner countries;
• implement AIISM-PBL methodology in the partner countries and contribute to its further use;
• evaluate the results of AIISM implementing;
• disseminate experience and results of the project among interested parties.

On 19-22 of February, 2014 an organizational meeting of project participants was held in the Polytechnic University of Valencia (Spain), and on 23-24 of October, 2014, a regular meeting of MEDIS project executors was held in the Technical University of Sofia (Bulgaria). At this stage, teachers of information systems department carried out a possibilities analysis of AIISM integration in the master curricula and trainings for trainers for courses under consideration has started. Project participants from the EU university partners have developed educational programs based on problem-oriented and project-oriented approaches to learning.

The methodology aims to create a working environment for students that are as close as possible to the working environment in industrial companies.

The teaching consists of:
1. Lectures: that describes the main idea by showing examples of corresponding applications.
2. Seminars: that discusses solutions to the stated real problem proposed by the group.
3. Laboratory classes: where students apply the proposed solution.
4. Mini-projects: that teaches to plan, design and develop solutions for effective control of distributed and complex manufacturing processes.

The developed methodology is mainly based on problem based learning (PBL) and other accepted active learning techniques with the intention of creating a realistic working environment which the student will experience in his future career. This model is based on the educational goals proposed by the Accreditation Board for Engineering and Technology (ABET) [2] and different experiences [3,4].

Proposed method of learning requires specially equipped laboratory classes (industrial computers, microcomputers, simulators, etc.). At this stage of the project works on the purchase of related equipment is being carried out. AIISM module consists of 5 sub-modules:
• Industrial computers;
• Microcomputers;
• Mobile and cloud computing platforms;

- Industrial networks and protocols;
- Industrial controllers and simulators.

Project executors from Kazakhstan and foreign countries:
- Universitat Politecnica de Valencia (Spain)
- Malardalen University (Sweden);
- Technical University of Sofia (Bulgaria);
- University of Stuttgart (Germany);
- University of Porto (Portugal);
- al-Farabi Kazakh National University (Kazakhstan);
- S.Toraigyrov Pavlodar State University (Kazakhstan);
- Saint Petersburg State Polytechnic University (Russia);
- Petrozavodsk State University (Russia);
- Odessa National Polytechnic University (Ukraine);
- Ukrainian National Technical University (Ukraine).

According to the plan of MEDIS project the new curriculums were elaborated for the specialties of Information Systems, Information Security Systems, Automation and Control. Five new courses will be introduced during 2015-2016 educational year. Parts of curricula for corresponding specialties are shown in tables 1,2,3.

TABLE 1. 6M070200– AUTOMATION AND CONTROL

Module	Code	Discipline name	Credits	Semester		
				I	II	III
Basic disciplines						
1.2. Elective module	Elective module #2 ASUSTMSS AU 5206	Microcomputers	3	1+1+1		
	Elective module #3 IK 5207	Industrial computers	3	1+1+1		
	Elective module #4 KSUTP 5208	Industrial controllers and simulators	3		1+1+1	
Specialized elective disciplines						
2.2. Elective module	Elective module #3 MOKP 6307	Mobile and cloud computing platforms	3			1+1+1
	Elective module #4 6308	Industrial networks and protocols	3			1+1+1

TABLE 2. 6M070300– INFORMATION SYSTEMS:

Module	Code	Discipline name	Credits	Semester		
				I	II	III
Basic disciplines						
1.2. Elective module	Elective module #2 ISP 5205	Industrial networks and protocols	3		1+1+1	

T<small>ABLE</small> 3. 6M100200–I<small>NFORMATION</small> S<small>ECURITY</small> S<small>YSTEMS</small>:

Module	Code	Discipline name	Credits	Semester		
				I	II	III
Basic disciplines						
1.2. Elective module	Elective module #3 MOKP 5307	Mobile and cloud computing platforms	3	1+1+1		

R<small>EFERENCES</small>

[11] Welcome to MEDIS. 2015. Retrieved from www.medis-tempus.eu.

[12] ABET. 2012. Criteria for accrediting engineering programs. Retrieved 03-21-12 from http://www.abet.org/DisplayTemplates/DocsHandbook.aspx?id=3143.

[13] Hassan Mohamed Houcine and others "A Hybrid Structured Methodology for Developing Computer-based Industrial Computer Systems" International Conference on Frontiers in Education (FECS). Las Vegas (EE.UU.) 2012

[14] Houcine Hassan Mohamed and others "An innovative proposal for the Industrial Informatics subject". International Conference on Engineering Education (ICEE). Valencia (Spain) July 2003

[15] Yessengaliyeva Zh., Mussiraliyeva Sh. "A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Information Systems (MEDIS project)". International Conference "Competence-oriented system of knowledge evaluation". Kazakhstan, Almaty, 17-18 january 2015.

[16] al-Farabi Kazakh National University. 2015. Retrieved from http://www.kaznu.kz/en.

Advanced problem-based learning: special features of the modern industrial control systems development training (based on the experience of the European universities participants to the TEMPUS MEDIS project)

O.N. Galchonkov, N.V. Loziienko

Odessa National Polytechnic University, Odessa, Ukraine

Abstract— **A content of the problem-based learning and advanced problem-based learning are highlighted. Integration of the APBL in the modern industrial system control specialists training has been described. Authors declared modern European experience in the field of modern industrial system control specialists training gained in the frames of TEMPUS MEDIS project within which 5 European universities share their experience with six universities in Kazakhstan, Russia and Ukraine.**

Index Terms— problem-based learning, advanced problem-based learning, industrial system control, training courses modernization

I. INTRODUCTION

The main aim of the article is to cover the key approaches and principles of the advanced problem-based learning (APBL) of the modern industrial control systems development. One of the distinctive features of modern production is continuously expanding automation, and constant growth and update of the range and complexity of software and hardware. This requires continuous training of specialists in new methods and means of production problems solving.

The modern university turns into a lifestyle designer, a structure that gives students the skills for continuous adaptation [1]. What we traditionally call culture today takes the form of a culture of learning in the field of professional activity, particularly of how to maintain a high level of relevant professional skills in the constantly changing environment. With the help of advanced problem-based learning the students are to realize that for future professional activity the flexibility of obtained skills and focus on further lifelong learning are just as important as a formal diploma and knowledge at the level of the current state of the industry.

According to Maslow's theory, a person has two needs that affect the success of both training and further professional activities [2]. This is the need to freely ask questions and to express oneself freely, and the need to know and understand. Consideration and development of these needs in students within the studied subject field is one of the main objectives of advanced problem-based learning.

II. THE MAIN FEATURES OF PROBLEM-BASED LEARNING

The constant intensification of industrial automation has induced the appearance of problem-based learning. The overcoming of an inconsistency between the limited university education duration and the constant growth of the proficiency level the students should have in order to solve particular problems of the modern production automation in the 1990s led to the replacement of the traditional approach to learning with the problem-based learning (PBL).

The traditional training methodology is characterized by the following:

1. Lectures have the key part in training.
2. The course is divided into separate modules studied one after the other with no specific emphasis on their joint usage.
3. The laboratory researches, practical trainings and course works are dedicated to certain topics and are of minor importance.
4. The professor has a central role.
5. The number of students is large (25-30 people).
6. Examinations are the main assessment method.

The usage of this methodology leads to that the students are in possession of vast academic knowledge, the major part of which will never be used at a particular manufacturing where they will be working. At the beginning of their work they need a lot of time to adapt to the software, devices and the development technologies used in the modern manufacturing. At the same time the enterprises are aimed at the production, so they are in need of qualified specialists in the first place, and are not exactly oriented towards the extra training of the university graduates.

The key principle of PBL is that the training of the students

involves solving of different cases, with the students themselves forming the questions necessary for the case solving and finding the answers to them on the basis of relevant information and joint discussion [3, 4]. There are no lectures at all or very few of them and they are dedicated only to the topics connected directly to particular cases. The classes are given for small groups. The subject syllabus is not divided into topics, but integrated into problem-based cases. The professor's key function as a knowledge source is replaced with the role of a tutor which only controls the process of case solving and, when necessary, channels it appropriately. The knowledge assessment is not single-pointed, but integrated into the process of the case solving.

The usage of such methodology leads to that the students gain practical experience in solving of the problems specific for their specialty, starting from their definition and finishing with their implementation and maintenance. The student don't gain the academic knowledge of their major in full, but they acquire a full set of problem-solving skills, starting from the definition and finishing with the practical implementation which allows them to successfully solve by analogy the problems, for the solving of which they didn't have the initial basic academic background. In other words, the graduates become the specialists fully ready to work in a narrow practical area and are capable of taking the solution of a new similar problem to the practical level quickly and confidently. The cooperation of the universities with each other and with the industry provides the covering of all the needs in all the possible specializations due to their distribution among different universities.

III. ADVANCED PROBLEM-BASED LEARNING

However, the technological progress never stalls, as new software, hardware, applications, etc. are being developed. It leads to the constantly growing difference between the education level of the school graduates which enroll at a university and the required proficiency level of the university graduates. That is why a necessity arose for the transformation of classical PBL into advanced problem-based learning (APBL). The main specific feature of this methodology is that several interconnected subjects are studied together and the solved cases are not limited by the contents of one course, but require a complex usage of all the courses and may be solved with the help of different sets of hardware and software [5…10]. The students learn all the possible solutions of the cases which allows them to freely operate all the possible range of technologies and choose the best variant in each practical situation in the future.

Thus, not the logic of scientific knowledge but professional problems is set in the core of education. APBL allows the transition from the objective principle of the education content forming to the creation of integrated training courses that reflect professional activity comprehensively. The nature of knowledge itself is changing. The criterion of "knowledge for the activity" becomes fundamental for the selection of educational content.

The APBL requires more complex courseware as compared to the classical PBL as it is necessary to plan simultaneous learning of several subjects (course units) which are studied side-by-side and develop the cases, the solving of which suggests complex usage of the decisions offered by different courses. Due to the complexity of methodological support there is a danger of unnecessary immersing into technical details and not fully taking into account the basic principles of learning [2] that provide resulting efficiency and ensure the continuity of the learning process:

- training is the process that requires active conscious effort; sequence of presentation and complication of educational material should form the students' confidence of that they will be able to master at least 70% of the proposed knowledge and skills, and that it will generate a positive state of satisfaction with the process of mastering of new knowledge and a conscious effort in the learning process;

- the professor's attitude to learning should be adaptive: he/she should control the proportion of the volume and complexity of tasks to the students' current opportunities; the tasks should not be too easy, otherwise the interest for learning will be lost, students should make every effort, which causes the feeling of satisfaction from the correspondence of the effort to the achieved results that amount at least 70% of the total volume of the proposed knowledge;

- the professor should encourage positive contagion of studying: even when working individually we adopt a lot from others; when there are students in the group who find the current learning process fun and joyful, and it is actively supported by the professor, then such an attitude is rapidly becoming common for the whole group of students;

- training should be practical: all theoretical provision should be put in their practical use during implementation of small projects and formation of students' definite skills of their use in future work; the professor should explain to students where and how to use the material studied at real plants, what future trends are possible and how the students can then improve their skills to meet these trends.

The APBL-cases have two versions, one for the students, and one for the teachers. The teachers' version has a full description of the situation and the essential information which the students have to gather from all the course units as well as the description of actions undertaken in certain situations for the solving of the specific problem or achievement of objectives in each possible variant. The usage techniques for the required hardware and software as well as the explanation of some notions, necessary formulae and indexes are also given here. In that way the teacher knows what the students have to discuss, in what order the decision-making should be done, what should be used as the basic hardware and software modules and what are the possible solutions of the case. An important part is the controlled object (CO) introduced at the very beginning of the case (only in the teachers' version) for which the students should by themself

form the volume of the analyzed information, the ways and accuracy of its obtaining, the adjustment devices and the ways of operating them. Before doing some practical actions the students should discuss and introduce the solutions of the problem in whole which were chosen within the group.

The students' version of a case suggests the studying of the required hardware and software one after the other. Every device required for the problem solution in general should be previously studied apart from the others. The aim of these small substeps is the studying of both the ware and the methods of its isolated testing which gives an opportunity to find out the reason for the failure to work of the big system consisting of a substantial quantity of different units. The next step is the gradual increment of the equipment and software up till the development of a whole system. With that, every new increment should be tested for the correct work of the procured complex of devices. An essential moment in the case solution is to teach the students how to work with technical documents. On the basis of general understanding of the hardware and software structure and their documentation, the students are to learn to quickly find the relevant information in technical specifications which consist of thousands of pages. The second important moment should be the elaboration of the students' skills for using the programmes examples and the ways of operating which are provided both by the hardware and software producers and the development environments producers.

The third important moment is to teach the students how to produce documentation for the received solution of a case, how to present both the intermediary results and the final solution as well as a reasonable argumentation of correctness and effectiveness of planned and produced actions.

The fourth moment which shows the strong distinction of APBL from PBL is the different role of a teacher. In APBL he is not a tutor but more of a stalker, i.e. a guide which helps to find at least one solution for the case. The modern development environments of hardware and software are quite complex products with wide functionality which are constantly upgraded and added. The whole studying of these wares all the more without any relation to a certain problem requires a lot of time and effort from the students. In APBL the students, first supervised by a professor, study one of the ways of getting from the initial task in general to an operating device. Later, the circle of the equipment in use and the amount of the necessary efforts from the students increases gradually for the unsupervised studying of the sub-solutions of a case. Thus, the students acquire not just the basic approach to the solving of the certain type of problems, but also learn about the development environments together with the skills of deepening the degree of knowledge of these environments.

IV.   IMPLEMENTATION OF APBL IN THE TEMPUS MEDIS PROJECT

In the TEMPUS MEDIS project the implementation of APBL

looks this way. The aim is to train the students in such a way which corresponds to the level of specialists fully proficient in the developing of modern industrial control systems. In order to do that the set of disciplines are put together and they will be studied at the same time and interconnectedly:

- The usage of the industrial computers together with the signal input and output facilities (to control a particular object which could be, for example, a water container with a water pump, an outflow valve, water level sensor, etc.)

- The usage of the modern microcontollers to build the control systems (a certain controller and a software development environment for it are taken for operation with the same object, for instance, the evaluation module STM32F429IDISCOVERY and development environment IAR or KEIL).

- The usage of standard field buses for the construction of distributed industrial control systems (RS232, RS485, I2C, CAN, MODBUS, etc.)

- The usage of a wireless communication channel for the development of the distributed industrial control systems (Bluetooth and Wi-Fi for the connection between tablets, between tablets and desktop computers, between tablets and microcontrollers, etc.)

- The modeling of the industrial control systems (in relation to those used in the previous subjects of hardware, software and controlled objects).

For each subject the classes are held in the form of lectures, seminars, laboratory researches and mini-projects:

Lectures: the teacher sets out the basic theory related to the planned mini-projects, their importance for practical application, the full range of the hardware and software which could be used in the projects, and the design technology and development environment. Apart from the basic method of a mini-project implementation through the given course, the alternative variants are definitely listed which are studied as the basic ones at the other subject units of other courses.

Seminars: are carried out in the form of discussion within the students' team (of 4 students) at the core of which is the current state of mini-projects, the presentation of the accomplished steps with the analysis of the usage peculiarities of the chosen hardware and software, the planning of the next step for the mini-projects implementations.

Laboratory researches: the basic for this course hardware and software are studied which are necessary for the mini-projects implementation. The students work in pairs and acquire basic set of skills related to the topics presented in the lectures. The tasks in the laboratory researches resolve particular and clear-set problems; they have guidelines, are fully documented and their complexity gradually increases. For instance, the following tasks could be used: create a project in the design environment IAR for a microcontroller, input a discrete signal from a sensor, output a discrete signal to an actuator, input an analog signal from a sensor, output a signal from the pulse-width modulation for the DC motor control, display the control process on the screen of a microprocessor, etc. The teacher helps the students in the

usage of the design environments for the progress from the problem definition to the working device. The teacher should monitor the students' practical training by answering their questions and guiding them using hints enough to make them find some of the solutions themselves. The typical mistakes, problems, and interesting approaches found during the laboratory research should be discussed with the whole group.

Mini-projects: the students integrally use their knowledge and skills which they have acquired during the lectures, laboratory researchers and seminars for the full-scale development of the physical process controller. At every next stage the hardware and software are added which are being studied at the parallel courses. For instance, at the first step the object control is done with the help of the microcontroller software with the display of the controllable parameters on its screen. At the second stage, an industrial computer is connected to the used equipment through the MODBUS and on its monitor the control processes are displayed. At the third stage, the second microcontroller is connected to the first controller through the CAN bus which receives data from the additional sensors. At the fourth stage, a wireless channel is added between one of the microcontrollers and the tablet, and the control of the physical process simultaneously from the industrial computer and tablet is implemented. At every stage, the controller configuration which will be implemented on this stage is preliminary modulated together with the object of the control for the test of the work correctness of the used regulators in all the possible function modes of the controlled object.

It should be pointed out that lectures and laboratory researches are clearly related to their subjects. Mini-projects and seminars have the form of interdisciplinary classes. After the mini-project the students are to be capable of integrating the instruments and procedures practiced during the course units in order to develop non-complex medium-sized applications for control of the physical processes. They also should be able to document the developed hardware and software and give a presentation on the process of the mini-project implementation and its results.

As well as in classical PBL, one of the essential conditions for the APBL is the active cooperation among the students aimed at quicker and more diverse gain of the required knowledge and skills, and also the group decision-making, the ability to work in a team. An important advantage of APBL is the training of the development skills on the basis of a whole unit of courses. Moreover, the projects are implemented in the format which is as close as possible to the requirements of an industry, i.e. the following takes place:

- The general definition of the problem;
- The team work on the project;
- The choice and justification of the most effective problem solution;
- The choice of the most effective hardware and software for the problem solution;
- The project review;
- The practical realization of the project;
- Commissioning works;
- Acceptance tests for all the function modes;
- Full documenting of the implemented project;
- The presentation of the project.

An important advantage of APBL as compared to the traditional learning is the creation of the positive motivation to study through the feeling of an achievement of a practically meaningful result which is in-demand in real life. The students become more confident, they acquire a portfolio of the problems they are capable of solving on a high level. This, in its turn, leads to the sense of accomplishment from the learning process, higher self-esteem and acknowledgement from others.

V. CONCLUSION

The TEMPUS MEDIS project is aimed at the integration of APBL into the modern industrial control systems specialists training. However, the APBL itself allows to increase the effectiveness of the education in any activity areas due to the gradually increasing requirements to the higher education institutions graduates. These may be both the practical side of certain industrial types of activity and the scientific researches in some areas, creative projects, informational (introductory and tentative) projects, etc. The main principle implemented here is that the knowledge and skills are gained not only for some future use, but for a specific type of activity. And the level of these knowledge and skills should provide the carrying out of this activity on the highest technological level not just now but in the future as well, as the university graduates will know not only how to make something on a high level, but how to keep to this level at all times.

One of the crucial factors for the implementation of lifelong learning of specialists is to solve the problem of continuing education of the professors. This concerns both the subject areas and teaching methodologies. Professionals in every field need to constantly upgrade their qualifications to be "in shape". And this is especially important for those whose profession concerns teaching others

VI. [ACKNOWLEDGMENT

REFERENCES

[1] A. I. Kravchenko, "Life-long learning: flexibility and growth", *Distance learning Centre "Elitarium"* [Online]. Available: www.alitarium.ru/2010/09/29/neprepryvnoe_obrazovanie.html

[2] L. Staut. "Learning efficiency: seven fundamentals"," *Distance learning Centre "Elitarium"* [Online]. Available: www.elitarium.ru/2007/04/27/jeffektivnost_obuchenija.html

[3] Rhem J. "Problem-Based Learning: An Introduction", Presented at Conference "The National Teaching & Learning Forum" December 1998, Vol. 8 No. 1. Available: http://www.ntlf.com/html/pi/9812/pbl_1.htm

[4] A.V. Iskrenko, T.A. Polton. "Problem-based learning: features of teaching methods in GB (the example of St.George University of London, Great Britain)", *Scientific journal of Belgorod State University. Series: History. Political science. Economics. Computer science.*- Issue №8.-Vol.10.-2008.-pp. 214-218

[5] Joint European Tempus project "A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics Systems" (MEDIS) 544490-TEMPUS-1-2013-1-ES-TEMPUS-JPCR.

[6] H. Hassan, J. Martínez, C. Domínguez, A. Perles, J. Albaladejo " Innovative Methodology to Improve the Quality of Electronic Engineering Formation through Teaching Industrial Computer Engineering". IEEE Transactions on Education Vol: 47, No 4, pp: 446-452, 2004.

[7] Hernando, M.; Galan, R.; Navarro, I.; Rodriguez-Losada, D.; , "Ten Years of Cybertech: The Educational Benefits of Bullfighting Robotics," IEEE Transactions on Education, vol.54, no.4, pp.569-575, Nov. 2011.

[8] J. R. Savery and T. M. Duffy, "Problem Based Learning: An Instructional Model and its Constructivist Framework," in Constructivist Learning Environments: Case Studies in Instructional Design, B. Wilson, Ed. Englewood Cliffs, NJ: Educational Technology Publications, 1996.

[9] H. Hassan, C. Dominguez, J.-M. Martinez, A. Perles, and J. Albaladejo, "Remote laboratory architecture for the validation of industrial control applications," IEEE Trans. Ind. Electron., vol. 54, no. 6, pp. 3094–3102, Dec. 2007.

[10] H. Hassan, J. Martínez, C. Domínguez, A. Perles, J. Albaladejo, JV. Capella. Integrated Multicourse Project Based Learning in Electronic Engineering, International Journal of Engineering Education, Vol. 24, Nº 3, pp: 581-591, 2008

Oleg Galchonkov received the Dipl. Ing. degree from the Odessa Polytechnic Institute in 1977 and received Candidate of Science degree (Ph.D) in Electrical Engineering from Leningrad Electrotechnical Institute (St. Petersburg, USSR), in 1987.

He spent 33 years in industry, with Design Organization "Diskret", Golden Telecom, MMDS Ukraine before joining Institute of Computer Systems ONPU in 2010, as an assistant professor . His current research focus is on adaptive antenna processing, digital signal processing, adaptive noise compensation, embedded microcontroller control systems, design and development of advanced industrial informatics and control systems.
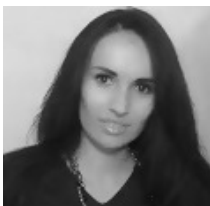
Natalia Loziienko was born in Odessa, Ukraine in 1984. She received the B.S. degree in economics from the Odessa National Polytechnic University, in 2005 and M.S. degree in management of foreign trade activity, from the Odessa National Polytechnic University, in 2007. She is currently pursuing the Ph.D. degree in economics at Odessa National Polytechnic University.

From 2008 to 2011, she was an research assistant and teacher at Management of foreign trade and innovation activity Department at Odessa National Polytechnic University. She is the co-author of one book, more than 10 scientific articles. Her research and teaching interest includes marketing and advertising, as well as innovation management in the field of education and science.

Mrs. Loziienko occupied the position of senior specialist in the field of management of international projects in the Marketing and Innovation Politics Department of Odessa National Polytechnic University since 2007.