Project Acronym: MEDIS

Project Title: A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics Systems

Contract Number:

Starting date: **Ending date:**

Deliverable Number: 1.2

Title of the Deliverable: Design of the AIISM-PBL methodology - Learning Activities

Task/WP related to the Deliverable: Design of the AIISM-PBL methodology – Structure of the AIISM courses

Type (Internal or Restricted or Public): internal

Author(s): Ivanov, I. E., Markov, E. K., Mladenov, V. M., Petrakieva, S. K.

Partner(s) Contributing:

Contractual Date of Delivery to the CEC: 31-03-2014

Actual Date of Delivery to the CEC: 31-03-2014

<u>Project Co-ordinator</u>

Company name :	Universitat Politecnica de Valencia (UPV)
Name of representative :	Houcine Hassan
Address :	Camino de Vera, s/n. 46022-Valencia (Spain)
Phone number :	+34 96 387 7578
Fax number :	
E-mail :	husein@upv.es
Project WEB site address :	

Context

WP 1	
WPLeader	Universitat Politècnica de València (UPV)
Task 1.1	
Task Leader	TUS
Dependencies	UPV, USTUTT, MDU, UP
Starting date	
Release date	

Author(s)	Ivanov, I. E., Markov, E. K., Mladenov, V. M., Petrakieva, S. K.
Contributor(s)	
Reviewers	

History

Version	**Date**	**Author**	**Comments**
1.0	2014/02/22	Ivanov, I. E., Markov, E. K., Mladenov, V. M., Petrakieva, S. K.	Initial version
1.1	2014/02/26	Ivanov, I. E., Markov, E. K., Mladenov, V. M., Petrakieva, S. K.	Revised version 1
1.2	2014/03/20	Ivanov, I. E., Markov, E. K., Mladenov, V. M., Petrakieva, S. K.	Revised version 2
1.3	2014/03/31	Ivanov, I. E., Markov, E. K., Mladenov, V. M., Petrakieva, S. K.	Final version

Table of Contents

1 Executive summary

This deliverable describes the learning activities proposed to implement the module “Process controllers and simulators” for the Advanced Industrial Informatics Specialization Module (AIISM).

The AIISM is conveniently structured in different activities of progressive complexity to facilitate the teams to develop their projects along the course. The learning sessions are organized in these activities: lectures, seminars, laboratories, mini-project and tutorship.

In this deliverable, the structure, learning activities, prerequisites and schedule for “Process controllers and simulators” module are provided.

2 Introduction

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) [1] and different experiences [2-5]. The general aims of the approach are:

- To guarantee that the student has a knowledge about the fundamentals of the specialization.
- To encourage the students to work as part of a team in solving industrial problems.
- 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.
- Due to the rapid advances in this area, to develop the capacity to adapt to any new computer based systems that may appear in the future.

Taking into account this methodology, all the learning activities are driven around a reference project. The example project for this module will be the control of a liquids tank.

The AIISM is conveniently structured in different activities, with progressive complexity to facilitate the development of projects along the course.

In order for the student to fully profit from the learning activities, a list of prerequisites is provided in the form of background knowledge and required skills that the student is expected to fulfil before beginning the activity.

3 Pre-requisites

These are the pre-requisites for this module:

- Basic C programming
- Basic digital electronics
- Basic analog electronics
- Single-board computer architectures and programming
- Basic Boolean logic

4 The liquids tank model

The specification, design, implementation and validation of microcontroller based control systems are taught using a simplified continuous process: a liquids tank. The size and complexity of this process is adequate to support the explanation of the essential concepts.

The process consists of a liquid tank that must be regulated at a reference temperature and to provide the fluid through a valve. A heater is responsible for heating the liquid and a pump to supply liquid to the tank. Level and temperature sensors allow knowing the current amount of liquid and temperature in the tank.

To sense physical magnitudes, the tank is equipped with the following sensors:

- An analog temperature sensor
- An analog level sensor
- A digital overflow sensor
- A digital overheat sensor

To actuate on the physics of the tank, and close the control loop, the following actuators are provided:

- A digital heater
- A digital valve
- An analog driver for the pump

Figure 1(a) shows a diagram of the different elements of the tank model.

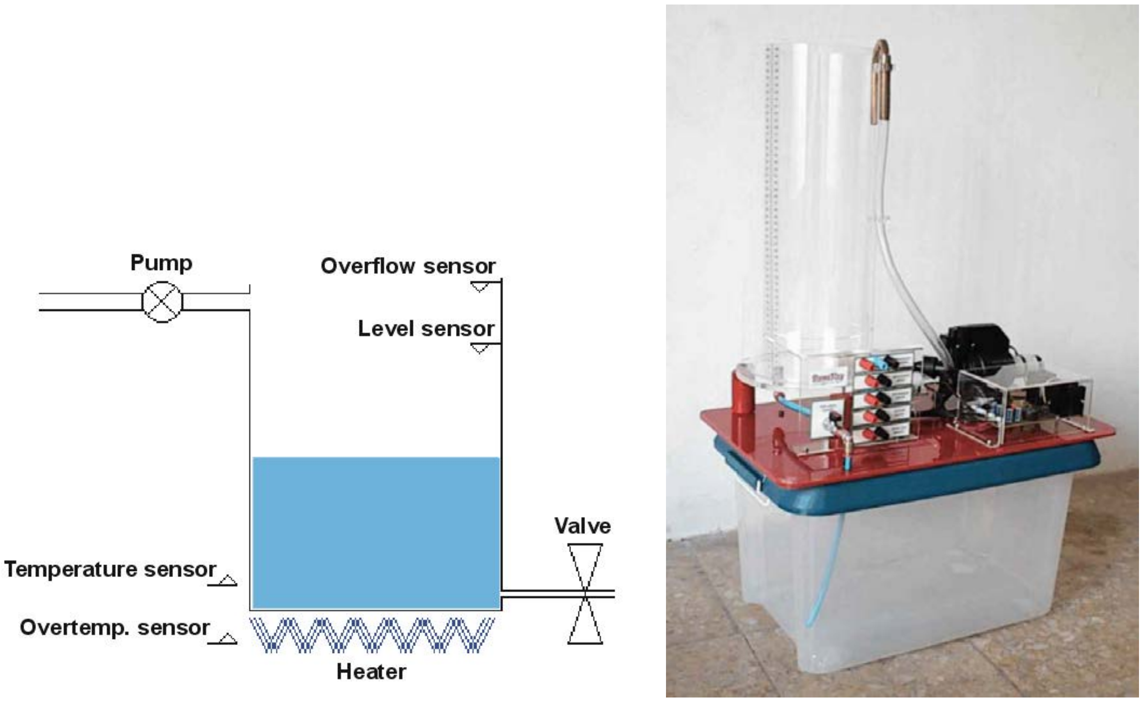

Figure 1-a Diagram of the scale model. 1-b Real tank developed model

The figure 1-b shows a real tank model that implements the characteristics of this proposal.

Sensors and actuators of the process are connected either directly to a microcontroller or by an electronic amplifier circuit and follow this description.

Signal	Direction	Type	Values/Range
Valve	Output	Digital	TTL, “0”->closed, “1”->open
Pump	Output	Analog	linear, 0V = 0%, 5V = 100% (pulse-width-modulated)
Heater	Output	Digital	TTL, “0” ->off, “1”->on
Overheat	Input	Digital	TTL, “0”->Overheat!
Overflow	Input	Digital	TTL, “0”->Overflow!
Liquid temperature	Analog	Input	Temperature Variable Resistor
Liquid level	Analog	Input	Variable Resistor

5 Module decomposition

Controller-based control systems are usually complex, but organized in a well-defined structure. This structure is oriented to industrial embedded control systems and state machines [7]. To develop a controller based system with sufficient guarantees of success it’s important to follow a validated methodology. It is necessary to keep the complexity of the problem limited in each phase of the project: planning, design, development, validation, deployment and exploitation.

The experience of the software developers’ community over the past 50 years has identified important issues to be taken into account in industrial informatics projects. The recommendations can be transferred to those in embedded-systems, but enriched by particularities like real-time aspects.

The other problem is system simulation in both aspects – computer simulation and semi-natural simulation (using real controller and hardware simulating the object). This aspect is rather important because it saves money, time, reduces the level of risk and can be used in many additional aspects.

Since people are the key element in the project organization and their attention capacities are limited, it is necessary to make an abstraction process, keeping apart the irrelevant parts of a problem and focusing on the important things. This is also ideal for the students work.

The module proposes some basic design methods and programming techniques that allow this abstraction. The design and development recommendations proposed are based on concepts such as: top/down design approaches, modeling the problem before start coding, modular decomposition of the program, and minimizing interfaces between the modules. In addition the modern program-generation technic will be described.

These are proposals of interest to ensure the success of the project:

- To mix different design approaches has advantages if tailored specifically to each part of the project. For example, top-down approach is recommended in the design of the tasks. The main control task can be decomposed in a set of more specific collaborating tasks: “observe the actual situation”, “desire a new situation”, “decide the action”, “act” and “report to user”. The bottom-up approach is recommended however for developing some of the basic utilities and services, such as the input / output operations to sense and manipulates the external process, or the reading / writing operations on the blackboard system.

- Formal modeling during the design phase has advantages, because it allows a systematic codification from the model and then it is also easy to verify the program using the model.
- Component approach for state-machine design and implementation will give the flexibility to implement different elements of control system’s functionality separately and to validate and verify each element and the composed solution, as well.
- The modular decomposition based on divide and conquers algorithms have advantages too, because it allows address a complex problem into simpler parts by abstraction and concealment. You can delegate the development of each part to different team members, and it accelerates the edit-compile cycle cause of the smaller translation units.
- The minimization of interfaces between the modules, through the “common data module”, has advantages, because it improves the consistency of information, avoiding duplication of variables, and simplifies the use of the variables by using a common interface. Additionally a centralized policy management and protection of the variables can be developed with the use of the blackboard access functions.

6 Learning activities

The learning activities are based on a PBL approach. This section provides a first approach of a timeline for the controller based module. The timeline and detailed contents of the learning activities will be fine-tuned in the following Work-Package 2, according to pilot experiences and the curricula of the partner country universities.

The course has been scheduled assuming duration of 15 weeks, with **4** hours of direct teacher student interaction per week.

Another basic of an engineer is the correct management and documentation 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.

Chapter	**Type**	**Topic**	**Week**														
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	**Introduction**																	
1	Lecture		x														
1	Seminar		x														
1	Lab		x														
1	Mini-project		x														
2	**Architecture of Computer Control Systems**																	
2	Lecture			x													
2	Seminar			x													
2	Lab			x													
2	Mini-project			x													
3	**Organization and structure of computers for control purposes**																	
3	Lecture				x												
3	Seminar				x												
3	Lab				x												
3	Mini-project				x												

4	**Basic control algorithms**																
4	Lecture					x											
4	Seminar					x											
4	Lab					x											
4	Mini-project					x											
5	**Real-Time software environment**																
5	Lecture						x										
5	Seminar						x										
5	Lab						x										
5	Mini-project						x										
5	**Real-Time software environment**																
5	Lecture							x									
5	Seminar							x									
5	Lab							x									
5	Mini-project							x									
6	**SCADA**																
6	Lecture								x								
6	Seminar								x								
6	Lab								x								
6	Mini-project								x								
6	**SCADA**																
6	Lecture									x							
6	Seminar									x							
6	Lab									x							
6	Mini-project									x							
7	**Simulators – general theory**																
7	Lecture										x						
7	Seminar										x						
7	Lab										x						
7	Mini-project										x						
7	**Simulators – general theory**																
7	Lecture											x					
7	Seminar											x					
7	Lab											x					
7	Mini-project											x					
8	**Simulators – practical aspects**																
8	Lecture												x				
8	Seminar												x				
8	Lab												x				
8	Mini-project												x				
8	**Simulators – practical aspects**																
8	Lecture													x			
8	Seminar													x			
8	Lab													x			
8	Mini-project													x			
9	**Simulation of distributed objects and control systems**																
9	Lecture														x		
9	Seminar														x		
9	Lab														x		
9	Mini-project														x		
10	**Simulators validation**																
10	Lecture															x	

10	Seminar															x	
10	Lab															x	
10	Mini-project															x	
11	**Real-Time system improvement using simulation environment**																
11	Lecture																x
11	Seminar																x
11	Lab																x
11	Mini-project																x

As it has been stated, this is a provisional proposal that will be developed in subsequent WPs.

7 Scheduling

This section provides a proposed scheduling distribution in weeks.

7.1. Introduction

Classification and characteristics of the Computer Control Systems (CCS): embedded (specialized) systems; control systems for industrial applications with standardized functions.

7.2. Architecture of Computer Control Systems

Functional organization of the modern hierarchical industrial control systems.
Types of computer control and data acquisition systems: data collection systems, supervisor control, direct numerical control, logical controllers, etc. Modern, decentralized and distributed control systems.

7.3. Organization and structure of computers for control purposes

Organization and structure of computers for the industrial controller and for embedded system. Analog and discrete I / O subsystems; analog and discrete control peripherals.
Organization of computational processes in CCS for continuous control. Concept of static and dynamic process scheduling.
Organization of computational processes in CCS interacting with discrete objects: implementation of synchronous, asynchronous and synchronous-asynchronous state machines.

7.4. Basic control algorithms

Controllers for analog objects - standard functions and algorithms; concept of configuration vs coding - specialized languages for continuous control system. Controllers for discrete objects - logical and sequential controllers.

7.5. Real-Time software environment - ×2

Real-Time operating systems - functions and subsystems; management processes (tasks). Scheduling in hard real-time constraints.

7.6. SCADA - ×2

Basic structure. Functions. Standards. Connection to the controllers. Interfaces.

7.7. Simulators – general theory - ×2

Software-in-the-Loop simulators. Hardware-in-the-Loop simulators. Agent-based simulators.

7.8. Simulators – practical aspects - ×2

Computer simulators including process periphery. Connecting the controller to the simulator. Setting-up the simulator. Induction of errors and special situations. Keeping the history of the process. Analysis encountered in real operation problems. Training of the personnel.

7.9. Simulation of distributed objects and control systems

Virtual monomachine approach. Component approach. Communication network influences – simulation.

7.10. Simulators validation

Validation using the "Configure/Reconfigure" approach.

7.11. Real-Time system improvement using simulation environment

Model improvements. Software improvements. Performance optimization.

8 References

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

[2] Albaladejo Meroño, José; Domínguez Montagud, Carlos Pascual; Escolar Bartet, Luis Tomás; Perles Ivars, Ángel Francisco; Martínez Rubio, Juan Miguel; Hassan Mohamed, Houcine; Capella Hernández, Juan Vicente. “A Hybrid Structured Methodology for Developing Computer-based Industrial Computer Systems” International Conference on Frontiers in Education (FECS). Las Vegas (EE.UU.) 2012

[3] Houcine Hassan Mohamed; Juan Vicente Capella Hernández; Ángel Perles Ivars; Juan Miguel Martínez Rubio; Carlos Domínguez Montagud; José Albaladejo Meroño. "An innovative proposal for the Industrial Informatics subject". International Conference on Engineering Education (ICEE). Valencia (Spain) July 2003.

[4] Houcine Hassan; Martinez Rubio, Juan Miguel; Dominguez Monteagud, Carlos; Perles Ivars, Angel; Albaladejo Meroño, Jose. "Innovative Methodology to Improve the Quality of Electronic Engineering Formation through Teaching Industrial Computer Engineering". IEEE Transactions on Education. (2004). 47-4, pp: 446-452

[5] Hassan, H.; Domínguez, C.; Martínez, J-M.; Perles, A.; Albaladejo, J.; Capella, J-V. "Integrated Multicourse Project-based Learning in Electronic Engineering". International Journal of Engineering Education, Volume 24, Number 3, March 2008 , pp. 581-591(11)

[6] Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, http://LeeSeshia.org, ISBN 978-0-557-70857-4, 2011.

[7] http://en.wikibooks.org/wiki/Embedded_Control_Systems_Design

[8] MathWorks, Generate and verify embedded code for prototyping or production, http://www.mathworks.com/embedded-code-generation/, [last accessed: 08.08.2014].