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

WP1.2: Design AIISM - Learning Activities

Co-funded by the Tempus Programme of the European Union

544490-TEMPUS-1-2013-1-ES-TEMPUS-JPCR

Authors: Albaladejo, J., Capella, J.V., Perles, A

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

WP1.2: Design AIISM - Learning Activities

Contract Number: 544490-TEMPUS-1-2013-1-ES-TEMPUS-JPCR

Starting date: 01/12/2013 Ending date: 30/11/2016

Deliverable Number: 1.2

Title of the Deliverable: Design AIISM - Learning Activities

Task/WP related to the Deliverable: Design of the AIISM-PBL methodology – Learning Activities

Type (Internal or Restricted or Public): Public

Author(s): Albaladejo, J., Capella, J.V., Perles, A

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 :	+34 963877579
E-mail :	husein@upv.es
Project WEB site address :	https://www.medis-tempus.eu

Context

WP 1	Design of the AIISM-PBL methodology
WPLeader	Universitat Politècnica deValència (UPV)
Task 1.2	Design AIISM - Learning Activities
Task Leader	UPV
Dependencies	USTUTT, TUSofia, MDU, UP

Author(s)	Albaladejo, J., Capella, J.V., Perles, A
Contributor(s)	Martínez, J.M., Hassan, H., Domínguez, C.,

History

Version	Date	Author	Comments
1.0	31/01/2014	UPV Team	Initial draft
2.0	14/02/2014	UPV Team	Final version

Table of Contents

1 (Executive summary

This section of workpackage WP 1.2 details the learning activities of the module of Industrial computers for the Advanced Industrial Informatics Specialization Module (AIISM).

The AIISM is conveniently structured in different activities of progressive complexity culminating in a final mini-project where the students will use the new knowledge acquired to develop a complete industrial control application. 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 Industrial Computers 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 acquires knowledge regarding 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.

It is necessary to establish for each activity a prerequisites list in order to provide the background and required skills in order to take profit of the activity.

3 Pre-requisites

These are the pre-requisites for this module:

- Basic C programming
- Basic digital electronics and algebra
- Basic analog electronics

4 The liquids tank model

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

The process consists on a liquid tank that must be regulated at a reference temperature 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 instrumented 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) show a diagram of the different element of the tank model.

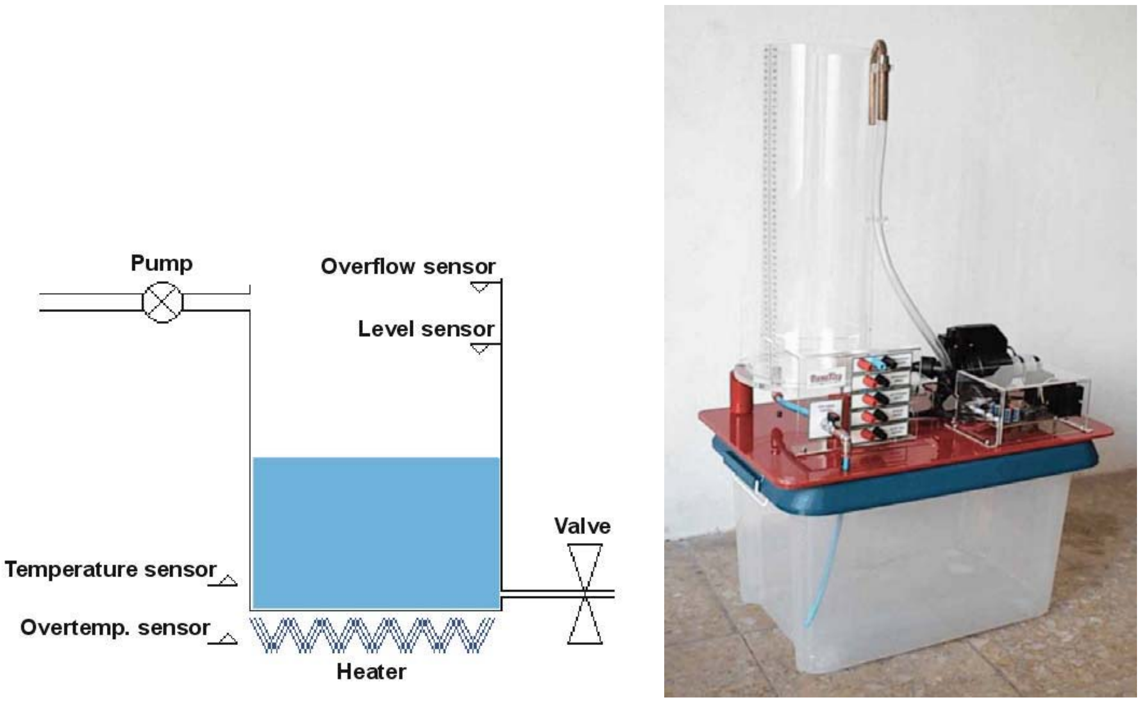

FIGURE 1-A DIAGRAM OF THE SCALE MODEL. 1-B REAL TANK DEVELOPED MODEL

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

Sensors and actuators of the process are connected to a given Data Adquisition Card (DAQ) and follow this description.

Signal	**Direction**	**Type**	**Values/Range**
Valve	Output	Digital	TTL, “0”->closed, “1”->open
Pump	Output	Analog	linear, -5v = 0%, 5v = 100%
Heater	Output	Digital	TTL, “0” ->off, “1”->on
Overheat	Input	Digital	Ttl, “0”->Overheat!
Overflow	Input	Digital	Ttl, “0”->Overflow!
Liquid temperature	Analog	Input	linear, -5v = 0º, 5v = 100º
Liquid level	Analog	Input	linear, -5v = 0 liters, 5v = 8000 liters

For example, we can use a National Instruments USB-6008 DAQ because it is low copst and can handle these signal. Figure 2 shows the aspect of this card.

FIGURE 2 ASPECT OF A NATIONAL INSTRUMENTS USB-6008 DAQ

An example of final GUI for the proposed project is showed in figure 3.

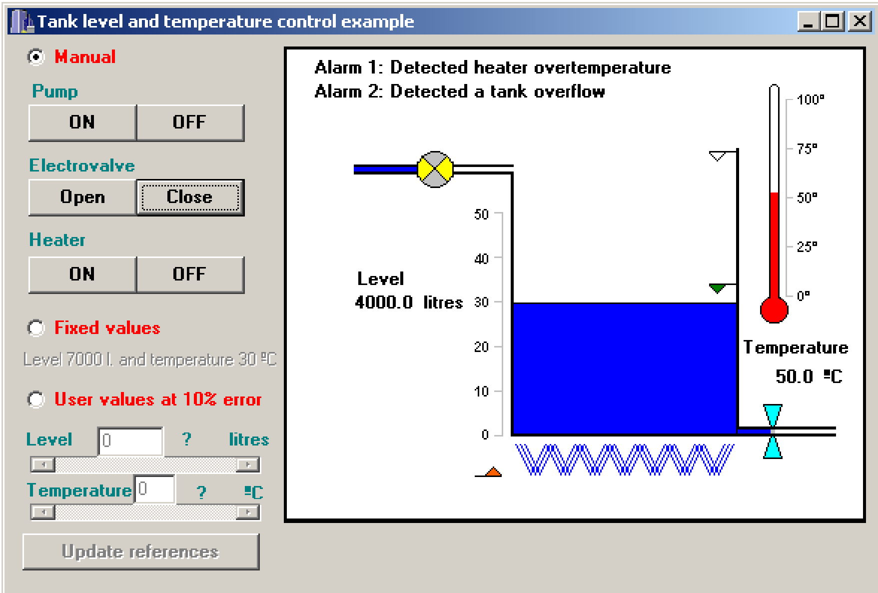

FIGURE 2. EXAMPLE OF USER INTERFACE FOR THE TANK PROJECT

5 Module decomposition

Industrial informatics systems are usually complex. To develop an industrial computer system with sufficient guarantees of success is important to follow a validated methodology. It is necessary to keep bounded the complexity of the problem in each of the phases 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 are similar to those proposed for problem solving in other areas, although adapted to the particularities of industrial informatics systems.

Mainly, the basic and permanent recommendation consists of dividing the complicated problems in smaller or simpler pieces that can be addressed separately, based on the basics principles of the structured programming.

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.

We can use 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.

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. For example to formally design a state machine and then to easily code it using a “switch” sentence in C language.
- The modular decomposition based on divide and conquer algorithms has 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.

To apply this idea, we arrive to a decomposition of any industrial process using the diagram of figure 4. It is important to note that this decomposition is referred to a simplified problem focused to the learning needs of the students.

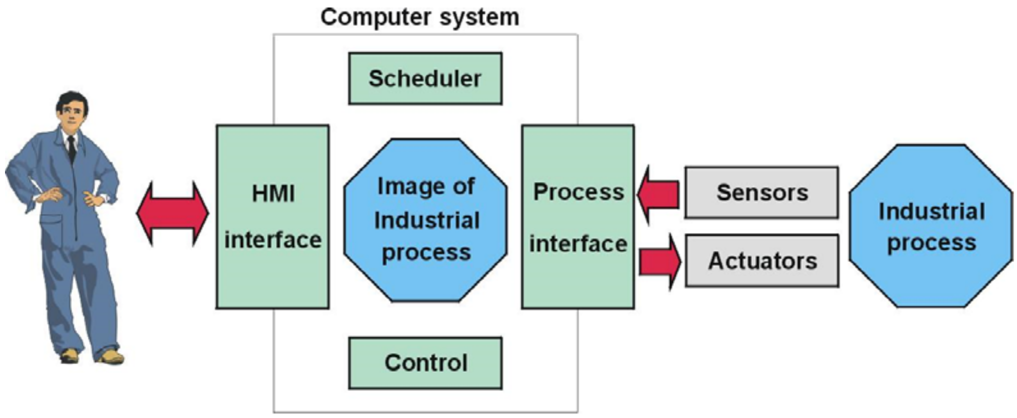

FIGURE 3. PROPOSED MODULAR DECOMPOSITION OF AN INDUSTRIAL CONTROL APPLICATION

6 Learning activities

Based in the previous proposals, it is necessary to provide an adequate set of chapters to group different topics. The following set of chapters is proposed:

1 Introduction to industrial informatics

2 Computer architecture

3 Project management

4 Software development

5 Process interface

6 Graphical user interface

7 Task scheduling

8 Regulation strategies

9 Integration and validation

Chapters 1 and 2 introduce basics about computer architecture and the applicability of computers to industry.

Another basic of an engineer is the correct management of a project. This is the objective of the chapter 3 that is spread along the course. This is also an horizontal content of the module, so it is spread along the course and in a position where student understands its implications.

Chapter 4 develops skills on C programming to be applied on the application creation. This is a horizontal requirement of the module.

Chapter 5 deals with the connection of the computer to the real world, the so called “process interface”. This is set tends to motivate the student because he/she sees the interaction with physical reality.

Taking into consideration that the actual student’s generation is accustomed to stunning visual user interfaces. Chapter 6 is in place for introducing another motivating set of activities related to this aspect.

At this point, it is necessary to start coordinating actions inside the application. So chapter 7 introduces the very basics around task coordination/scheduling.

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 chapter 8.

A serious project of industrial informatics needs an investing on testing of echa piece and integration. This important task is in chapter 9.

This proposal should be considered as a first approach, and the definitive distribution must be adjusted according pilot experiences and the curricula of the Universities.

7 Scheduling

In this section, we propose the scheduling distribution of chapters in weeks, including some highlights about he contents.

Week	Chapter	Type	Topic
	1	INTRODUCTION	
1	1	Lecture	Introduction to industrial informatics
1	1	Lab	Development environment - Programming the "Hello World"
1	1	Seminar	C programming (1) - Basic resources
1	1	Miniproject	Presentation of the problem to solve
2	1	Lecture	Structure and design of industrial informatics systems
2	1	Lab	Event oriented programming
2	1	Seminar	C programming (2) - Programming tools
2	1	Miniproject	Analysis of the project requirements
	2	COMPUTER	
3	2	Lecture	Computer architecture
3	2	Lab	Using libraries in C
3	2	Seminar	C programming (3) - Libraries
3	2	Miniproject	Project formal specification
	3	PROJECT PLANNING	
4	3	Lecture	Project management (1)
4	3	Lab	Tools for project management
4	3	Seminar	Discussing cases of project management systems
4	3	Miniproject	Project planning
	4	PROGRAMMING + DATA	
5	4	Lecture	Modular programming
5	4	Lab	Modular programming in C
5	4	Seminar	Modular programming resources
5	4	Miniproject	Modular decomposition of the program
6	4	Lecture	Data representation and sharing
6	4	Lab	Data sharing between C modules
6	4	Seminar	Chosing the appropiate data representation
6	4	Miniproject	Implementation of the shared data module
	5	PROCESS INTERFACE	
7	5	Lecture	Process interface (1) - Introduction and digital input
7	5	Lab	Digital input
7	5	Seminar	DAQ card (1) - Introduction and digital input
7	5	Miniproject	Implementation of the process interface module (1) - DI
8	5	Lecture	Process interface (2) - Digital output
8	5	Lab	Digital output
8	5	Seminar	DAQ card (2) - Digital output
8	5	Miniproject	Implementation of the process interface module (2) - DO
9	5	Lecture	Process interface (3) - Analog input and output
9	5	Lab	Analog input and output
9	5	Seminar	DAQ card (3) - Analog input and output
9	5	Miniproject	Implementation of the process interface module (3) - AIO
	6	USER INTERFACE	
10	6	Lecture	Graphical user interface (1) - Introduction
10	6	Lab	Programming GUI controls
10	6	Seminar	Graphical user interface for the industry (1) - Basic
10	6	Miniproject	Implementation of the user interface module (1) - Basic
11	6	Lecture	Graphical user interface (2) - Advanced resources
11	6	Lab	Programming a GUI for an industrial application
11	6	Seminar	Graphical user interface for the industry (2) - Advanced
11	6	Miniproject	Implementation of the user interface module (2) - Advanced
	7	TASKS	
12	7	Lecture	Task scheduling
12	7	Lab	Basic scheduler
12	7	Seminar	Scheduling strategies
12	7	Miniproject	Implementation of the task scheduler module
	8	REGULATION	
13	8	Lecture	Foundations and continuous control
13	8	Lab	Programming regulation strategies (1) cc
13	8	Seminar	Control strategies (1) cc
13	8	Miniproject	Implementation of the regulator module (1) cc
14	8	Lecture	Event-driven control
14	8	Lab	Programming regulation strategies (2) edc
14	8	Seminar	Control strategies (2) edc
14	8	Miniproject	Implementation of the regulator module (2) edc
	10	PROJECT (2) ENDING	
15	10	Lecture	Project documentation and presentation
15	10	Lab	Tools for project documentation
15	10	Seminar	Project documentation strategies
15	10	Miniproject	Documentation and presentation of the project

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)

.