SENSOR NETWORKS PROJECTS IN AN INTERDISCIPLINARY CAPSTONE COURSE

María Feldgen[1], Osvaldo Clúa[2]

Abstract *— Engineers must be prepared to approach a wide range of interdisciplinary problems that blur the traditional boundaries within the computer science curriculum. To help our students develop a broader view of the computer science discipline in automation and engineering, senior Informatics Engineering students at University of Buenos Aires, are required to take a capstone industrial applications workshop. This course helps students learn soft and experimental skills that are not introduced in previous laboratories and also provides an interdisciplinary experience that combines design, analysis, implementation, and testing. Students working in a professional like environment design and deploy a wide range of industrial systems for our university research laboratories. In this paper, we describe the experiences and outcomes using sensor networks. The deployment of the sensor networks motes provides strong student motivation for learning, offers comprehensive and valuable educational experiences, enhances student performance, and is well suited for a semester course.*

Index Terms — Capstone course, Design, Interdisciplinary Education, Wireless Sensor Networks.

INTRODUCTION

Whereas until recently most engineers have been trained in one very specific discipline, today's product designers need a more interdisciplinary education and knowledge in systems design [1]. Engineering capstone courses in our institution have included instructional components that emphasize interdisciplinary teamwork, the relationship of teamwork to engineering and writing and speaking to various audiences about a project. Students from several engineering domains use an integrated and concurrent design approach. The focus of these courses has been to encourage students using the framework that teaming provides to leverage their existing technical expertise in design and development. These are mandatory courses for students finishing their studies of mechanical, electrical, industrial, electronics or informatics (computer science) engineering.

The structure of capstone design courses varies significantly among different engineering departments. While nearly senior level design courses have the same basic purpose of providing students with experiential learning activities that satisfy Accreditation Boards in Engineering and Technology requirements [2] [3]. The capstone courses in computer science (CS) engineering include projects that have diverse hardware components. Projects include the interaction of computer software with the associated hardware. Projects of the other engineering fields share the same hardware and problems.

Teams of 3 or 4 students work on a unique "real compelling application " that results in the development of a software artifact for a sponsor, interacting with experts, engineers, vendors and students of other engineering fields. The project sponsors are external to the School of Engineering or representing a particular research facility within the School. Many of these research facilities lacks software applications for their own developed devices, controllers or mini plants or needs simulation tools or prototypes to model different scenarios for their teaching, research experiences, and their own capstone courses.

In the experience described in this paper, computer engineering and electronics (control) engineering students enrolled in different capstone courses shared the lectures, laboratory experiences and projects on process control and instrumentation. The projects sponsor was the Automatic Control Laboratory of our School of Engineering. The projects required management, design and implementation of a sensor network with virtual and real devices (motes) using a real time operating system. C code was generated automatically using MATLAB tools (Simulink) [4]. EE students' duty was to develop the plant and control systems in Simulink and advise CS students in control issues. CS student teams design and implemented prototypes and procedures to convert the automatic generated code for the motes.

Teams must produce well-written documents detailing the design and the prototyping experience. They must plan and manage the project time. In this paper we give an overview of our capstone course experience, it nature and their importance to engineering education. We give details about our capstone design course objectives and structure. We give details of the projects, involving specific software tools and development of applications for sensor networks and outcomes. We conclude with a summary of the motivational aspects that made these projects successful,

WIRELESS SENSOR NETWORKS

The possibilities and challenges offered by wireless sensor networks in control applications both in theory and in practice are widely recognized and many research teams and

1 María Feldgen, Facultad de Ingeniería, Universidad de Buenos Aires, Paseo Colón 850, Buenos Aires, Argentina, mfeldgen@ieee.org
2 Osvaldo Clúa, Facultad de Ingeniería, Universidad de Buenos Aires, Paseo Colón 850, Buenos Aires, Argentina, oclúa@ieee.org

companies are active in the design and implementation of units that encompass these attributes of sensing, processing, communication and actuation. Devices of this kind, which are created either as prototypes or as commercial products, are generally referred to as “motes”. A mote is an autonomous, compact device, a sensor unit that also has the capability of processing and communicating wirelessly. Despite the autonomy they present, the big strength of motes is that they can form networks and co-operate according to various models and architectures. These networks, known as wireless sensor networks, have been the focus of considerable research efforts in the areas of communications (protocols, routing, coding, error correction etc), electronics (energy efficiency, miniaturization) and control (networked control system, theory and applications) [5].

A wide variety of applications and systems with very different requirements and characteristics are using WSN. As a consequence, it is becoming increasingly difficult to discuss typical requirements regarding hardware issues and software support. It is a multidisciplinary research area where close collaboration between users, application domain experts, hardware designers, and software developers is needed to implement efficient systems [6]. Programming sensor networks is currently a cumbersome and error-prone task since it requires programming individual sensor nodes, using low-level programming languages, interfacing to the hardware and the network, only supported by primitive real time operating system abstractions [7].

System design and implementation on such a platform introduces a series of issues if the WSN is used for control applications. These problems are suited to combine teams across disciplines, to provide a broader experience to the student.

Hands-on experiments of a WSN give a great opportunity for students to understand the state-of-the-art technology, its concepts, applications in modern telecommunication and information and automation systems. We believe this knowledge is already essential for engineers from CS as well as for EE. However, implementation of only a part of functions using well-designed evaluation platforms is not generally appropriate to fulfill students’ motivation and achievement. Standard WSN solutions, such as ZigBee [8], were designed with the idea, that primary user concern is just about applications only and that a deep knowledge of WSN is unnecessary. This is unsuitable for some education purposes.

CAPSTONE COURSE

Our course, *Industrial Applications Workshop*, includes a capstone project that offers a real-world experience in solving engineering problems and a small amount of additional material, but the major focus is on a significant team project including design, implementation and project management of a distributed system across diverse hardware devices and computers. Problem-Based Learning (PBL) methodologies are applied in order to provide this kind of learning environment [9][10] and, interaction and cooperation with engineers and students from other engineering fields.

The perspective of our capstone course, using Tim Rhodus et al. definition, is the “Real World" Preparatory Experience Perspective” [11]. It encourages students to focus on their immediate post-college future and to "sum-up" their experiences. This type of capstone course includes experiential learning activities that have “authentic involvement”. The authentic involvement activities expose the student to real situations with totally open-ended projects, although the faculty may influence the selection of the situations and set performance criteria to assure that positive learning objectives are met [12]. Authentic involvements use outside clients or sponsors.

Some of our research facilities have special hardware, own developed devices, controllers, and mini plants without adequate software products. Many of the experiments and learning experiences have special deployment and a variety of configuration and programming activities. Model examples and prototypes with adequate guidelines are helpful for students and researchers with a basic background in programming. These labs are our sponsors.

Our previous capstone course experiences were done with common bus architectures as industrial networks [13], building libraries for an ABB robotic arm [14] and controlling mobile robots in labyrinths and simulations using neural networks [15].

In this paper we describe the experience where CS students interacts with EE (control) students and researchers of the Control Lab of our School of Engineering using XBee sensors with MATLAB.

Projects selection and preparation

During the first week, faculty members and students meet with the sponsors of the project. The topics of the projects are critical for team motivation

The topic of the project is critical for team motivation because no team arises without a performance challenge that is meaningful for those involved [16]]. Faculty members of the capstone course and the labs have identified the projects that have the following motivational and capstone project characteristics:

- The project involves completion of a finished product and it will be a valuable tool for the sponsor’s lab.
- The project is of sufficient complexity and scope to be intellectually and academically worthwhile. It has some acceptable risks and some unexpected obstacles, but with high possibility of success for the team.
- The project must be achievable within time constraints of the course and is not critic for the sponsor.
- It supports the integration of and reflection on the knowledge that students have gained during previous stages of their education.

- The sponsor's lab, external organizations, and the Department of Computer Science provide the resources and technical assistance.
- The sponsor provides additional technical support (a course) to ensure that the proposed projects are appropriately scoped. Students attend the course to acquire a background in the other disciplines related to the project. These courses are capstone courses of other engineering programs. Students of different engineering fields share lectures, the labs and the project. In addition, the sponsor attends students' team interview meetings, reviews team proposals, provides prompt feedback, and attends team final product presentation. He/she invites hardware suppliers and experts as technical advisors.

In interdisciplinary projects, the success of a team is determined by their ability to overcome communication and organizational problems, not technical ones. Each discipline relies upon a specific culture and language to establish a methodology for problem definition and solution. To prepare CS students for these challenges and the environment they will likely face in industry, a significant amount of time is devoted to background preparation in the sponsor's engineering field and "unguided exploratory work" interacting with the students' teams of the sponsor's course.

The course has CS discipline-oriented complementary lectures, project management sessions, inspections and walkthroughs and discussion sessions.

Project development and assessment

Students are required to apply a wide range of analysis, fabrication, test, and field operation techniques and tools during the design and development process. Relevant to computing-related skills, these include the use of integrated development environments, a wide range of programming languages, UML specification tools, various operating systems, a range of microcontroller hardware platforms, industrial networks and board design/layout/fabrication tools.

Assessment of performance is based on product (60%) and process (40%). The product is assessed for its appropriateness as a solution to the sponsor brief, the functionality and features, the technical standards used and the accompanying documentation. The process is assessed on the project plan, risk analysis, how effectively this plan has been monitored and controlled, interviews, inspections and walkthroughs, and presentations. The final product presentation is a public launch with a marketing campaign (contrasting with other products or "IT solutions"), formal presentations, demonstrations of the product and a "free lunch" for all sponsor, instructors, and attendees.

SENSOR NETWORK PROJECTS

The Automatic Control Lab incorporated a set of XBee OEM RF modules, development boards and Atmel AVR microcontrollers for design and implementation of prototypes of motes to study different networked control problems, to analyze how the network in the closed loop affects the system performance, and to designing controllers that consider this effect to optimize system performance.

Students and researchers of these labs have a basic background in programming and proficient in the use of MATLAB and their tools Simulink, Real Time Workshop and Real Time Windows Target, and the different toolboxes. Simulink is widely used in control theory and digital signal processing for multidomain simulation and design. They do not have experience in motes programming.

Simulink

Simulink, developed by The MathWorks, is a commercial tool for modeling, simulating and analyzing multidomain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It is integrated with the MATLAB environment. Coupled with Real-Time Workshop, another product from Simulink can automatically generate C source code for real-time implementation of systems. The Real-Time Windows Target is software that allows running C code generated by Real-Time Workshop on a PC in real time. In this environment, a PC is the host for MATLAB, Simulink, and the Real-TimeWorkshop [4].

TinyOS and nesC

Motes are controlled and make them perform a certain way by software programs built on a real time operating system. TinyOS is the traditional real time and open-source component-based embedded operating system developed for the motes. TinyOS provides a set of software components that allows applications to interact with the processor, network transceiver and the sensors. Applications written for this operating system are compiled and statically linked with nesC code and loaded on the flash memory of the microcontroller for execution. The main goal is efficient application execution in resource-constrained devices [17].

The nesC language is for programming structured component-based applications for sensor network embedded systems. It has a C-like syntax. nesC applications are built out of components with well-defined, bidirectional interfaces. NesC defines a concurrency model, based on tasks and hardware event handlers.

In addition, TOSSIM is a discrete event simulator for TinyOS sensor networks. Instead of compiling a TinyOS application for a mote, users can compile it into the TOSSIM framework, which runs on a PC. This allows users to debug, test, and analyze algorithms in a controlled and repeatable environment [18].

The projects

The sponsor of the project identified a labs need for the development of their experiences: It is hard for people using

Simulink to understand the TinyOS model. The objective was that students and researcher could continue developing their control experiences in Simulink and transfers them to TinyOS without a deep understanding of the underlying technology.

Based on that complex engineering problems need to be thought of in terms of different abstraction levels. The concept of abstraction is defined as the simplification of a complex problem by breaking it down into manageable components.

The goal of the project was to create this type of abstractions, hiding the motes/MATLAB interaction and instructing people how to "cut and paste" Simulink C code as nesC code in provided mote prototypes. The design and development of the systems must be documented as manuals for further developments.

Two projects emerged. **TinyModels**: A project developing general control models to migrate Simulink C code to TinyOS. **TankSoft**: A project developing specific experiences for control of an interconnected tanks system. Both projects distributed the sensing/control software between a mote (simulated by a PC running TOSSIM) and a computer running MATLAB/Simulink. In both cases, using the TCP/IP stack of TinyOS and the libraries for monitoring or operation.

Every project was in charge of a team. Teams collaborated each other, EE students, labs researchers and faculty acted as experts in control.

The teams used the Unified Process (UP) methodology for the development of the project and the framework for project management IEEE1058.

Teams conducted interviews to begin with the requirements workflow to outline the experiences to prototype. During the interviews, preplanned questions are asked and subsequent questions are posed in response of the client answers. After an interview was concluded, teams prepared a written report for the interviewee.

The teams evaluated and presented different alternatives to choose to the sponsor. The decision was to design and build toolboxes for MATLAB/Simulink. The toolboxes include a set of control panels for the interaction between devices with TinyOS and the MATLAB/Simulink environment and a library for TCP/IP communications. The original development framework of the lab was preserved.

The prototypes were designed, implemented, tested, and documented using the same steps (1 and 2) as used usually in the lab experiences. They add additional steps to turn the Simulink model into TinyOS sensor network:

1. EE students develop a controller model in Simulink. The model is validated.
2. C code is generated and compiled with Simulink's Real Time Workshop. The model run under control of the Real Time Windows Target to test it.
3. The C code was migrated to nesC in the TinyOS prototype to run on a device (TOSSIM PC in this case, because the "real" motes were under construction).
4. The interfaces to interact with the device with TinyOS are programmed using ad-hoc

TinyModels:

The goal of this project was to execute in TinyOS general controller models designed in Simulink. TinyModels includes a control panel for this type of models with a user access control system with privileges, an interface to process the outcomes in MATLAB and an ad-hoc protocol for TCP/IP communications (Figure 1).

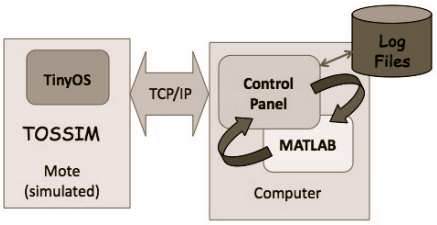

FIGURE 1
TINYMODELS SYSTEM

An example of the prototype models is a PID control loop with anti-windup system included (Figure 2 and 3).

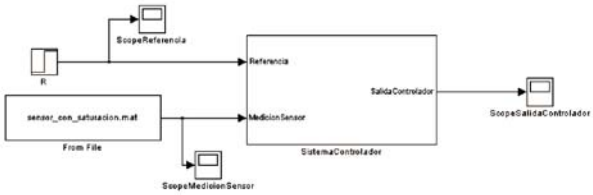

FIGURE 2
PDI CONTROL LOOP WITH ANTI-WINDUP IN SIMULINK

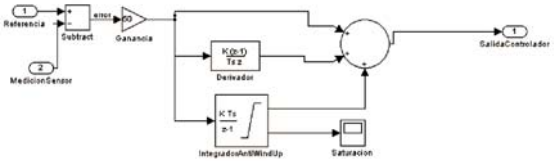

FIGURE 3
CONTROLLER DETAILS IN SIMULINK

C code generated by the Real Time Workshop was partially extracted to create the mote's TinyOS application. TinyModels manages the mote, the communications, and the interaction with MATLAB. Using the motes control panel, the user can modify some parameters of the PID controllers of the models in execution. In addition, the panel provides security features, user profiles, communication management, storing and retrieving of data and executions.

TankSoft:

The goal of this project was to build a sensor network prototype for a traditional set of experiences of the lab controlling an interconnected tanks system equipped with a variety of sensors and actuators. The software running in the motes filter noise in sensors input data. The motes communicate with a monitoring PC running MATLAB (Figure 4).

The system filters noise at the source of the input data and send the information preprocessed to the application. It is the first time that the lab uses motes for digital control and it enables a new set of experiments to study the side effects

of sensor measurement imperfections, redundant motes to overcome sensor failures, technological limitations, spatial, and temporal coverage problems, energy restrictions, etc.

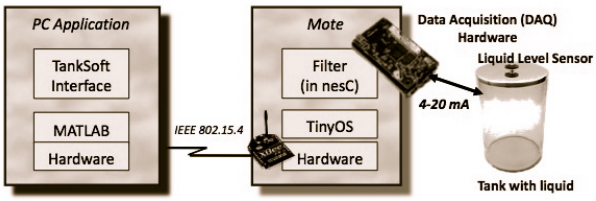

FIGURE 4
TANKSOFT ARCHITECTURE

A Java implemented user interface store the variables in the MATLAB workspace. Therefore it is possible to configure filters, measurements vector and simulations integrated into the traditional MATLAB framework. The interface is shown in Figure 5.

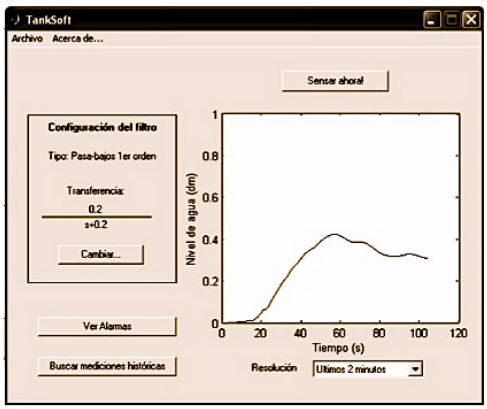

FIGURE 5
TANKSOFT USER INTERFACE

The filters were implemented with Simulink, the C code was generated using the Real Time Workshop and partially extracted and converted to nesC. This code was implemented in the motes with TinyOS.

CONCLUDING REMARKS

Highly motivated student teams can exceed sponsors and faculty expectations. One of the key attributes of this motivation is that the students identify with the project objectives. If the project topic is compelling and the projects outcomes are valuable for the sponsor's lab, students commit time and effort. Obstacles are treated as challenges to overcome and the projects objective becomes more important than the course grade. Wireless Sensor Networks is this compelling problem. It is very complex with side effects, but it is a new technology with a lot to research to do.

The people of the lab are proud users of the project final products, sharing with the teams and instructors the satisfaction of a positive outcome. The most relevant indicator of their motivation and commitment with the project was that team's main purpose involved discovering a new technology, developing, and using their natural talents and expertise to help other people to deal with sensor networks. This is the kind of satisfaction depends on how valuable is the outcome of the project for a real client.

REFERENCES

[1] Siegwart, R, "Grasping the Interdisciplinarity of Mechatronics", *IEEE Robotics and Automation Magazine*, Vol 8 No.2, June 2001, pp 27-34.

[2] ABET Engineering Accreditation Commission, "Criteria for Accrediting Engineering Programs," November 2002, http://www.abet.org/

[3] CONEAU; Acreditación de carreras de Ingeniería. http://www.**coneau**.edu.ar/

[4] Simulink – Simulation and Model based Design, http://www.mathworks.com/products/simulink/

[5] Arampatzis,T., Lygeros, J., Manesis, S. "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks", Proceedings of the 13th Mediterranean Conference on Control and Automation, Limassol, Cyprus, June 2005, pp. 719-724.

[6] Römer, K., Mattern, F., "The Design Space of Wireless Sensor Networks", IEEE Wireless Communications, Vol.11, Nº 6, Dec. 2004, pp. 54-61.

[7] Römer, K., "Programming Paradigms and Middleware for Sensor Networks," GI/ITG WORKSHOP ON SENSOR NETWORKS, 2004, págs. 49--54.

[8] "ZigBee Alliance.", http://www.zigbee.org/

[9] Bridges, E. and Hallinger, P. Implementing Problem Based Learning in Leadership Development, ERIC Clearinghouse on Education Management, 1995.

[10] Samford University, Problem-based Learning http://www.samford.edu/pbl/background.html

[11] Rhodus, T., Hoskins, J., "Toward a Philosophy for Capstone Courses in Horticulture" http://hcs.osu.edu/hcs/EM/docs/capstone.html

[12] Harrisberger, L., R. Heydinger, J. Seeley, and M. Talburtt, Experiential Learning in Engineering Education, American Society for Engineering Education, 1976.

[13] Feldgen, M., Clúa O., Ferreira, F., " An Interdisciplinary Teaming Laboratory on Industrial Networks ", *Proceedings of the 32th ASEE /IEEE Frontiers in Education (FIE2002)*, Boston, MA, USA, 2002.

[14] Feldgen,M., Clúa, O., Anigstein, M.: "CE, EE and ME Students using a Robotic-Arm as an Integration Tool". GCETE 2005, Santos, Brazil. March 2005.

[15] Feldgen,M., Clúa, O.,"Capstone Courses Where Students Interact With Experts". ICECE 2007, Monguaguá, Brazil. March 2007.

[16] Katzenbach, J. R., Smith, D. K., The Wisdom of Teams: Creating the High-Performance Organization, HarperBusiness, 1994.

[17] Haenselmann, T. An FDL'ed Textbook on Sensor Networks, University of Mannheim, Mannheim, Germany, 2007.

[18] TinyOS, http://www.**tinyos**.net