A First Course on Cyber Physical Systems

Walid Taha*† Robert Cartwright*† Roland Philippsen*
*Halmstad University
†Rice University

***Abstract*—A key challenge in designing a CPS curriculum is the fact that effective and creative CPS development requires expertise in disparate fields and that have traditionally been taught in distinct disciplines. Another challenge is educating students from diverse backgrounds on why CPS innovation is technically hard and on what technical tools can help them overcome the underlying sources of difficulty. In this paper we report on our recent experience developing and teaching a course on CPS. The course can be seen as a detailed proposal for addressing these challenges. This proposal addresses questions such as what constitutes core elements of CPS, how these core concepts can be integrated and related in an educational context, and what types of modeling tools can assist in teaching CPS. All materials including lecture notes and software used for the course are openly available online.**

I. Introduction

Teaching CPS raises two major challenges. First, CPS innovation requires mastery of concepts and skills that are traditionally assigned to distinct disciplines. Second, students who study CPS come from diverse backgrounds with differing expertise which makes it difficult to articulate why CPS design is technically hard and how specialized tools can help CPS developers overcome the technical problems involved in creating new systems.

We describe here a first course on CPS designed to address these challenges. The course focuses on answering the following three questions:

- What are the core elements of CPS?
- How are these core concepts integrated in the CPS design process?
- What kinds of modeling tools can help students create innovative cyber-physical systems?

Since this course was developed at Halmstad University in Sweden, compliance with the Bologna Process [3] mandated developing a formal syllabus [1] including a statement of educational outcomes. All course materials are freely available under appropriate open licenses. The living (continually evolving) lecture notes are available online [6] under a Creative Commons license and a portable distribution of modeling and simulation tool used in the course is available online [2] under a GPL license.

II. Educational Outcomes

The formal syllabus [1] for the course identifies the expected educational outcomes. This syllabus accurately describes the content of the course, but it is rather technical because of space limitations and its reliance on the vocabulary and terminology used to describe other components of the masters programs in Embedded and Intelligent Systems (EIS) at Halmstad University. From the perspective of the CPS community, the educational outcomes and goals of the course are more clearly described as helping students:

- Recognize the scope and scale of the potential impact of CPS innovation;
- Understand why many of tomorrow's innovations will be in CPS;
- Develop lifelong, sustainable skills and sensibilities for the analysis and design of innovative cyber-physical systems, including
  - back-of-the-envelope estimation,
  - familiarity with the fundamental sources of complexity in CPS design (such as system size, nature of continuous dynamics, discrete state size, different types of uncertainty), and
  - facility with virtual experimentation;
- Gain experience with the mathematical modeling and simulation of hybrid systems and the issues that arise when building and validating of such models;
- Assimilate a conceptual model of the CPS development process and master the terminology and communication skills required to reflect on, analyze, and critique development processes; and
- Develop an awareness of the different aspects of CPS, including scientific, engineering, and social aspects.

The course works toward realizing these educational outcomes through a technical focus on the topics described in the following section.

III. Core Elements and Topics of CPS

The course follows lecture notes [6] consisting of eight chapters covered at a rate of two lecture hours and two lab hours per week during an eight week term. Each chapter focuses on a topic that we view as a core element of CPS:

1) The present and pressing need for research and education in CPS ("What is CPS?")
2) Modeling Physical Systems;
3) Hybrid Systems;
4) Control;
5) Modeling Computational Systems;
6) Communications;
7) Case Study: A Single-Link Robot;
8) Multi-agent Systems.

The sequencing of the chapters is determined primarily by dependence. With only one exception, the ideas introduced in each chapter are used and explicitly reinforced in ensuing chapters. The exception is that the last chapter does not depend on the case study. The case study immerses students in a particular class of physical systems and helps them successfully complete the final milestone in a term project.

Each chapter is a largely self-contained introduction to each topic. At the same time, several examples and themes are shared across chapters. For example, a notion of *prototype equations* is introduced from the first chapter. These simple (first and second order) differential equations are revisited in almost every chapter as models and demonstrators of different concepts and phenomena. So, differential equations are introduced from the first week and students get a chance to work with them throughout the semester. Similarly, issues relating to energy and delay are revisited in different chapters. Naturally, creating notes that would enable a treatment of such a diverse set of topics requires an alignment of terminology and concepts, which also helps accentuate the differences in focus between topics.

The material and approach used in the lecture notes was influenced directly by ongoing interdisciplinary collaborations. Most of the material in the introduction is based on joint projects and discussions with colleagues from industry (Schlumberger, AB Volvo) and on discussions with numerous colleagues from the NSF CPS community. The approach taken and the emphasis in the chapters on modeling physical systems, hybrid systems, and control are influenced by collaborations with two professors of Mechanical Engineering (Marcia O'Malley from Rice University and Aaron Ames from Texas A&M University in the context of an ongoing NSF CPS project on Robot Design). They played a direct role in shaping the material on classical mechanics, variational principles, Lagrangian modeling, and the use of hybrid systems in modeling physical systems. Designating multi-agent systems as a core topic of CPS was inspired by the work of a professor of real-time and embedded systems (Tony Larsson) and of several other professors at the CERES and CAISR research centres at Halmstad University.

IV. Open Lecture Notes, Modeling Tools, Term Project, and External Resources

The lecture notes follow the eight chapter structure described above. These notes are freely available online under a Creative Commons license. We are in the process of expanding the notes from outlines to expository narratives of approximately fifteen pages each. We will continue to expand and develop their content as long as we teach the course.

The modeling and simulation tool used in the course is Acumen [2], [7], an interactive tool for modeling and simulating distributed dynamic hybrid systems. This tool is used in the course project, labs, and some of the homework assignments. It is designed as an interactive environment centered around a small textual language for modeling and simulating cyber-physical systems. The textual language consists of constructs that correspond to key concepts from either continuous or discrete mathematics. Building a tool around a parsimonious textual core can have several important benefits. First, it provides a bound on the intrinsic complexity of the tool in terms of concepts that the students are either already familiar with or will learn during the course. This helps ensure that students from diverse backgrounds have a fair chance of learning and mastering the tool quickly. Second, once students learn the core formalism from some canonical examples, it serves as a transparent vehicle for explicating the technical content of the course. Using a tool that is easy to learn helps empower students to become nascent CPS developers and encourages them to experiment with putative designs. In essence, the tool is designed to encourage students to become CPS hackers. A tool that makes it almost trivial to construct 3D animations as things that "naturally fall out" of the core textual formalism can help turn modeling and simulation into a creative form of game, and makes CPS modelling and simulation much more intuitive. Hacking and playing with math (hybrid differential equations) should be fun! Acumen helps us transform formidable design challenges into engaging exercises.

The course project focuses on designing a controller for an autonomous, three dimensional robot that can play Ping Pong. To introduce students to problem decomposition and iterative refinement, the project involves building controllers under increasingly more realistic scenarios. In addition, the project involves a series of tournaments where player models developed by different teams compete directly in a simulated environment. The cooperative/competitive organization of the project is intended to drive home the importance of testing as well as giving students experience with being focused on designing a complete system that must perform a complex "dynamic task" whose specifications are either only defined in high-level terms or may even be hard to fully specify in the first place. There are several reasons why robotics is a good examples of a CPS challenge, including that it

- Involves intimate coupling between cyber and physical components,
- Requires using hybrid systems models and non-linear ODEs (even the simplest rigid body modeling of 2D dynamics gives rise to such characteristics),
- Gives rise to several non-trivial and open control problems,
- Introduces embedded and real-time computation requirements, and
- Requires consideration of issues of communication, belief, and intent.

We believe that having the students work in teams to carry out this project within the context of a modeling and simulation environment based on mathematical models has several benefits, including that it:

- Motivates the use of modeling and simulation techniques to study a design problem,
- Helps them experience the value and the limitation of analytical techniques,
- Drives home the importance of developing extensive test scenarios as well as systematic experimentation,
- Allows them to run many more virtual experiments than they can do with physical experiments,
- Facilitates experimental measurement (and to some extent, evaluation) when compared with physical experiments
- Emphasizes the value of collaboration, teamwork, and competition.

External resources are included as embedded web links in the text of the lecture notes. Many of the references point to Wikipedia articles containing deeper discussions of topics mentioned in the course. Other references link to online video recordings relevant to CPS such as Lawrence Lessig's Threats to a Freedom to Innovation and Edward Lee's Heterogenous Actor Models.

V. Status, Excluded Topics, and Plans

We are currently teaching the course for the second time as elective course in a masters program in Embedded and Intelligent Systems (EIS) at Halmstad University. Starting in Fall 2013, it will be a mandatory first-semester course for all students in the EIS masters program. Our ultimate goal is to move this course earlier in the curriculum, first to the advanced undergraduate level and later to the freshman or sophomore year.

Every syllabus makes implicit choices about what material to exclude from a course. In our case, we chose not to include linear systems (of ordinary differential equations), mathematical models based on complex numbers, and introduction to a wide range of computational and simulation tools for CPS. This is not to say that these topics and tools are not important. Rather, we chose to exclude them because there are numerous excellent treatments of these subjects that students can pursue as needed on an independent basis after completing this course.

In contrast, there are topics not included in our first course syllabus that we would like to include in our core CPS curriculum. For example, the section on modeling focuses on justifying fundamental physical characteristics of computational components, how they interact with the surrounding world, and the difficulties that arise when we try to use them to implement continuously modeled components (typically controllers). We are working on expanding this section to cover more concrete aspects of computational components and systems, such as the principles involved in the design of embedded and real-time software (as covered, for example, in the recent textbook of Lee and Seshia [5]). In particular, we are developing material to introduce basic concepts from Lee's Models of Computation into that chapter to prepare students for subsequent exposure to this material in other courses. Our hope is that this integration will both enrich the course and make Models of Computation accessible to a broader audience.

As is often the case with new courses, developing practice problems that drive home the concepts introduced in lectures and notes can be challenging. Currently, vector algebra is introduced in the second course. We plan to introduce more material on geometric modeling in the second chapter. This is expected to simultaneously help students with distinguishing modeling the geometry from modeling the dynamics as well as to provide students with more practice with vector algebra.

In terms of tool support, the main area in which we expect to develop Acumen in the near future is to develop support for expressing and dealing with uncertainty in lab exercises. We see also a need to develop a library of basic components: Such a library would enable students to experiment with building large systems more rapidly. An appropriately crafted library may prove to be an effective tool for teaching the wide range of principles and skills needed for virtual experimentation. We have started developing such a library. At a more practical level, we are developing a socket-based interface that can expose the entire state of the simulated system to other tools such as Ptolemy II, MATLAB, and LabView. We are also exploring the possibility of supporting the Functional Mockup Interface (FMI) [4] under development by Daimler AG and other companies and research institutes.

Finally, we note the mention of ethical education in the call for papers for this workshop. This is an area that we would like to explore with educators and students with relevant experience participating in the meeting.

VI. Conclusions

There is significant agreement in the CPS community on the need for improved CPS education. A common challenge in realizing such an education is to address students coming from a wide range of different backgrounds. To address these issues, we have developed and offered a first course on CPS. This paper presented this course, including a statement of desired educational outcomes, a proposed selection of core CPS topics, and outline of course organization and materials. The course makes use of several openly available resources, including lecture notes, a modeling and simulation environment, and links to other external resources. The course makes extensive use of virtual experimentation to enrich and accelerate the learning of the theoretical and conceptual content introduced in lectures in a practical and cost-effective manner. Experience with the course has been positive, and a decision has been made to make it a required, first-semester course in the Embedded and Intelligent Systems (EIS) masters program at Halmstad University.

Acknowledgements

The primary support for the development of this course comes from Halmstad University, the Swedish Knowledge Foundation's CERES centre at Halmstad University, and the US National Science Foundation CPS Robot Design Project. Adam Duracz, Zain-ul-Abdin, and Tony Larsson kindly provided helpful feedback on a draft of this paper.

References

[1] Cyber physical systems. Course Syllabus adopted by the School of Information Science, Computer and Electrical Engineering, Halmstad University, Sweden, 2011. http://bit.ly/DA8003.

[2] Acumen web site. http://www.acumen-language.org, 2012.

[3] Bologna process (Wikipedia). http://en.wikipedia.org/wiki/Bologna_Process, 2012.

[4] Torsten Blochwitz, Martin Otter, Johan Åkesson, Martin Arnold, Christoph Clauss, Hilding Elmqvist, Markus Friedrich, Andreas Junghanns, Jakob Mauss, Dietmar Neumerkel, Hans Olsson, and Antoine Viel. Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. In *9th International Modelica Conference*, Munich, Germany, September 2012.

[5] 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.

[6] Walid Taha. Lecture notes on cyber physical systems. http://bit.ly/CPS-lecture-notes, 2012.

[7] Walid Taha and Roland Philippsen. Modeling basic aspects of cyber-physical systems. In *Workshop on Domain-Specific Languages and Models for Robotic Systems (DSLRob)*, 2012.