A First Course on Cyber Physical Systems

Walid Taha*† Robert Cartwright*† Roland Philippsen* Yingfu Zeng†

*Halmstad University

†Rice University

***Abstract*—Effective and creative CPS development requires expertise in disparate fields that have traditionally been taught in distinct disciplines. At the same time, students seeking a CPS education generally come from diverse educational backgrounds. 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 focused on three three key questions: What are the core elements of CPS? How can these core concepts be integrated in the CPS design process? What types of modeling tools can assist in the design of cyber-physical systems? Experience from the first two offerings of the course is promising, and we discuss the lessons learned. All materials including lecture notes and software used for the course are openly available online.**

I. Introduction

CPS innovation requires mastery of concepts and skills that are traditionally assigned to distinct disciplines. At the same time, students seeking a CPS education come from diverse educational backgrounds with differing expertise. This situation makes it difficult to articulate to students why CPS design is technically challenging and how specialized tools can help CPS developers overcome the technical problems involved in creating new systems.

This paper describes a first course designed to address these challenges. The course focuses on answering 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 formal statement of examinable educational outcomes. The Bologna Process is a series of agreements between European countries aimed at ensuring that the educational systems in various countries are comparable. All materials developed for our course 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 the interactive modeling and simulation environment 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 was the first formal description of the contents of the course. This formal description relies 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 broader CPS community, the educational outcomes and goals of the course can be 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 analyze and critique development processes; and
- Develop an awareness of the scientific, engineering, and social aspects of CPS.

The course seeks to realize these educational objectives by focusing 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 critical importance of research and education in CPS ("What is CPS?");
2) Modeling Physical Systems;
3) Hybrid Systems;
4) Control;
5) Modeling Computational Systems;
6) Communications;
7) Representative Case Study: A Single-Link Robot;
8) Multi-agent Systems.

The sequencing of these chapters is determined primarily by dependence. With 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 preceding case study. The case study immerses students in a particular class of physical systems and helps them successfully complete the term project, a major milestone in the course.

To accommodate students from diverse backgrounds, each chapter is a largely self-contained introduction to the designated topic. At the same time, several examples and themes are shared across chapters. For example, the concept of prototype equations (first and second order differential equations) is introduced in the first chapter and revisited in almost every chapter as mathematical characterizations of different concepts and phenomena that arise in CPS. As a result, differential equations are introduced in the first week of the course and students practice using them throughout the term. Similarly, issues relating to energy and delay are introduced and revisited in different chapters during the course. Naturally, composing a coherent set of lecture notes for such a diverse set of topics requires the construction of a common framework of terminology and concepts, which helps accentuate the differences in focus among the various disciplines involved in CPS.

The material and approach used in the lecture notes has been driven by ongoing interdisciplinary collaborations with colleagues at various institutions. Most of the material in the introduction is based on joint work and discussions with colleagues from industry (Schlumberger, AB Volvo) and on discussions with numerous researchers from the NSF CPS community. The chapters on Modeling Physical Systems, Hybrid Systems, and Control have been strongly 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 our understanding of 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 Tony Larsson, a professor of real-time and embedded systems at Halmstad University 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

As noted above, the lecture notes are divided into eight chapters. We are in the process of expanding the notes from outlines to expository narratives of approximately fifteen pages each, and most of the chapters are now in this form. We expect to continue to expand and develop the lecture notes as long as we teach the course.

The modeling and simulation tool used in the course is Acumen [2], [7], an interactive environment for distributed dynamic hybrid systems. It is used in the course project, laboratory sessions, and some of the homework assignments. Acumen is centered around a small textual language. The textual language consists of constructs that correspond to concepts from either continuous or discrete mathematics. A design premise is that building a tool around a parsimonious textual core can have several important benefits. First, it can provide a bound on the intrinsic complexity of the concepts that the students must master during the course. This bound can help 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, this formalism can serve as a common framework for articulating the concepts being taught. A tool that is easy to master can help empower students to become nascent CPS developers and encourage them to experiment with putative designs. In essence, the tool is designed to inspire students to become CPS hackers. A tool that makes it easy to construct 3D animations of CPS code can transform modeling and simulation into a creative game, making CPS modelling and simulation much more intuitive. Hacking and playing with mathematics (such as hybrid differential equations) should be fun!

The course project is 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 consists of building controllers under increasingly more realistic scenarios. In addition, the project involves a series of tournaments where players (controllers) developed by different teams compete directly in a simulated environment. The cooperative and competitive organization of the project is intended to drive home the importance of testing as well as to give students experience with developing a complete system that must perform a complex, dynamic task whose specifications are defined only in high-level terms that may be hard to fully specify.

Several factors make robot design a good example 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),
- Raises non-trivial, open-ended control problems,
- Introduces embedded and real-time computation requirements, and
- Motivates analyzing issues of communication, knowledge, belief, and intent.

Having the students develop the project in a high level modeling and simulation environment seems to have 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;
- Demonstrates 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) in comparison with constructing physical systems; and
- Reaffirms 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. Lessons Learned

In this section, we review and analyze the feedback from students in the first and second offerings of the course at Halmstad University. In addition to the formal student evaluations required for all courses at our University, the informal feedback received during exercises and labs proved very valuable.

In the first offering of the course some informal feedback was collected from students during the last lab. Most of the feedback focused on Acumen and the project, and included statements such as "it was fun to run' the design" and "support for 3D visualization was very useful." Explicit requests for "more intermediate exercises." indicated room for improvement, and when asked if we should make the project more interesting some replied "the current project is difficult enough." A few students suggested simplifying the project, such as reducing the Ping Pong game from 3D to 2D. There were several suggestions for improving the Acumen syntax and Integrated Development Environment (IDE), as well as a request for a reference manual.

We were encouraged by the positive feedback on using simulation with 3D visualization. Rather than retreating from 3D in the second offering of the course, we made a concerted effort to accurately explain the 3D model and its interface. We also implemented most of the suggestions for improving the Acumen syntax and its IDE. We streamlined the exercises, but remain aware that there is still room for more intermediate exercises.

The formal student evaluations for the first offering of the course revealed an encouraging level of student satisfaction with the course as a whole. Eight students completed these reviews. Most students stated that the course required about 15-25 hours of studying per week, but a sizable number said it required less than 15 hours, which we consider to be a bit light. Student satisfaction with course materials (at the time this consisted mainly of pointers to external reading materials) varied significantly. A free-form comment from one student stated that there was "too much reading on something easy". Satisfaction with examination forms (which include homeworks and projects) ranged from intermediate to very satisfied (top of the scale). All students either agreed that course objectives were completely met (top of the scale) or had no opinion (2 students). Most students were satisfied or very satisfied (top of the scale) with lectures, exercises, laboratory sessions, seminars, and assignments – with one student indicating an intermediate level of satisfaction with the lectures and exercises, respectively. Satisfaction with the information received about the course was intermediate to very satisfied (top of the scale). On a scale from 1 to 6 with 6 being the highest, six students reported a level of satisfaction of 5, with one student reporting a level of 4 and the other of 6.

The second course offering is still ongoing (but nearly complete) as this paper is being written. At the end of the seventh lecture, we asked students whether they would like to offer their feedback for the purposes of reporting in this paper. They were told the paper had been accepted. The instructor suggested that they first formulate briefly items to list under "What worked" and "What did not work". After items were listed, we sorted them in terms of how strongly students felt about them (#1 = most important, #2 = second most important, and #3 = other). Afterwards, the students were allowed to update the notes about these results. The following is a snapshot of the outcome:

Things that worked:

#1: Lecturing style: whiteboard with no slides, conversational, interactive

#2: Open access to all course material "was cool". Google docs for notes and scribe notes (done collaboratively, in pairs, in real-time)

#3: The course introduction was inspirational and motivating, the assignments enjoyable. Getting people to work together and help each other (on the project, homework assignments, and preparing for the final) was great and "made learning social." The supplementary reading and videos were fun. Students like the fact that the class takes a "comprehensive approach to the subject."

Things that did not work:

#1: There is a gap between lectures, which are high-level and intended to motivate and engage students, and the course assignments. Students would like the homework problems to include a progression of easier problems that builds up to the level of difficulty of the problems in the existing assignments.

#2: Some open-ended problems included in the course assignment are too big and too hard to manage. The chapter on communications, which covers concepts of information, knowledge, belief, and truth, is too abstract, making it hard for students to infer a clear take-away message. It might help to devote more lecture time to discussing the course assignments and project.

A notable feature of the feedback on the second offering of the course was that it did not explicitly refer to either the project or Acumen. Both were taken for granted as integral parts of the course. The students' discomfort with the chapter on communications may be due to the fact that the corresponding lectures cover epistemic topics in CPS and introduce the new topic of agent-based modeling.

We were encouraged by the feedback on the things that worked. In addition, many of the things that did not work confirmed the instructors' impressions of the course and helped formulating concrete action items. The instructors gave three unplanned tutorial sessions to help students with assignments and project in response to this feedback already.

The feedback on course materials for the second offering of the course was better than for the first, presumably because we expanded the lecture notes from outline to expository form (about 15 pages per chapter). We also reduced the overall amount of reading and made a clear distinction between required reading (which became the lecture notes) and recommended reading.

Based on student feedback and our experience with the course so far, we believe that the course has succeeded in showing that:

1) Hybrid systems can be used effectively as a unifying framework for thinking about a wide range of aspects of cyber-physical systems;

2) Simulation is very helpful in bringing hybrid automata to life;
3) Animated simulation for virtual experimentation engages students;
4) Using a project that focuses on designing a robot and evaluating its performance in a simulated environment supporting 3D animation allows students from diverse educational backgrounds to experiment and innovate in the context of CPS;
5) Using a competitive game engages students; and
6) Support for 3D visualization facilitates CPS design and makes the project accessible to nearly all students.

The one experiment that was clearly not an obvious success in the first offering of the course was teaching Lagrangian dynamics. While the aim is to make another attempt at introducing this material in a later offering of the course, the diversity of the student body suggested that a broader treatment of different classes of physical systems may deserve higher priority.

VI. Status, Excluded Topics, and Plans

So far, the course has been taught as a required course in one of three tracks that students must select between in the masters program in Embedded and Intelligent Systems (EIS) at Halmstad University. Starting in Fall 2013, it will be a required 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, depending on a student's level of preparation before entering college.

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 an introduction to the wide range of computational and simulation tools supporting CPS design. This is not to say that these topics and tools are not important. Rather, we chose to exclude them because the core principles of CPS can be taught in the context of a single high-level tool (Acumen) and there are numerous excellent treatments of these topics and tools that students can pursue as needed on an independent basis after completing a first course.

In contrast, there are topics not included in our first course syllabus that we would like to add to our core CPS curriculum. For example, the section on Modeling Computational Systems focuses on justifying the 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 embedded and real-time software design as covered, for example, in the recent textbook of Lee and Seshia [5]. We are developing material introducing the basic concepts from Lee's Models of Computation into the chapter on Modeling Physical Systems to prepare students for subsequent exposure to this material in other courses. We expect that this addition 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 embody the concepts introduced in lectures can be challenging. Currently, vector algebra is introduced in the chapter on Modeling Physical Systems. We plan to introduce more material on geometric modeling in this chapter, and expect that this will help students distinguish modeling system geometry from modeling system dynamics and gain more facility with vector algebra.

In terms of tool support, we plan to extend Acumen in the near future to support expressing uncertainty in models and simulations. We also plan to develop a library of basic components. Such a library can enable students to experiment with building large systems more rapidly. An appropriately crafted library can serve as an effective tool for teaching the wide range of principles and skills needed for virtual experimentation in CPS design. At a more practical level, we are developing a socket-based interface that exposes 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 inclusion 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.

VII. Conclusions

There is significant agreement in the CPS community on the need for better CPS curricula. A major challenge in forging such a curriculum is accommodating a student population with diverse educational backgrounds. To address these issues, we have developed and offered a first course on CPS that relies on a simple high level modeling and simulation language embodied in an interactive environment supporting 3D visualization. This paper describes the course, and includes a statement of desired educational outcomes, a proposed selection of core CPS topics, an outline of the course organization and materials, and lessons learned from the first two installments of the course. The course employs openly available resources, including lecture notes, a modeling and simulation environment (Acumen), and links to other external resources. The course makes extensive use of virtual experimentation and visualization to enrich and accelerate the learning of the concepts introduced in lectures in a practical and cost-effective manner. Experience with the course has been positive. Students taking the class came from undergraduate programs around the world and appear to have benefited uniformly from the course. 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 the 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. Kay Berkling, Adam Duracz, Zain-ul-Abdin, Nicolina

Månsson, Tony Larsson, and Mohammad Mousavi kindly provided invaluable feedback on drafts of this paper. Jörgen Carlsson helped us in the formulation and revision of the formal syllabus. We are indebted to the students of the first and second offerings of the course at Halmstad for their candid feedback.

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.