GRAND CHALLENGES
Education and Cross-Cutting Challenges In Cyber-Physical Systems

Jonathan Sprinkle, University of Arizona

October 24, 2008

Abstract

Information systems to control vehicles in the transportation domain are necessarily cyber-physical. This is true for controllers that are scheduling the arrival, departure, or switching of vehicles, as well as the on-vehicle controllers handling braking, steering, or autonomy. This paper extends a previous position paper centered on technical challenges in the automotive domain, to consider new cross-layer abstractions that will satisfy the challenging requirements of future Cyber-Physical Systems, and also provides thoughts on required educational curricula and programs to prepare the workforce to address CPS challenges in transportation systems. The paper's discussion centers on what grand challenges intersect all of the focus transportation sectors in this workshop. To conclude, there is some discussion on how fundamental understanding can be address in undergraduate—and graduate—curricula.

1 Introduction

Componentization (the treatment of a system as made up of various components) has the interesting recursive property that one designer's system may be a component in another designer's system. The use of modeling at various layers of abstraction permits us to think of an automobile or an aircraft as a system integrated by a manufacturer. Air traffic control, however, treats an individual aircraft as an atomic abstraction, and logic at a red light intersection is built in terms of vehicles as its atomic counting unit. The cyber-physical nature of these atomic components at a high level requires that any system controlling them must take into account physical limitations and constraints of those platforms. The timing of the red light must give sufficient time for a vehicle to stop based on the speed limit of the roadway it governs, and air traffic controllers must allow vehicles to maintain lift.

These straw-man examples (long solved through policies of design, but not necessarily solved generally in an automated way for arbitrarily scaled systems) belie the fundamental issue of cyber-physical systems: the interdependence of the physical platform with the algorithms and constraints of the software.

There is a major focus on the componentization of automotive software, and there must be some focus on how such components can be evaluated as units, as well as parts of a system, without assembling and driving the vehicle as the first step. For many such electronic components, an I/O test may be a sufficient initial test. However, components that deliver a CPS capability (e.g., automatic parking), the input/output set is too large to exercise fully. Further, there may be subtleties involving vehicle chassis that make software or algorithms work for one vehicle, and not another.

2 Grand Challenges for Transportation Cyber-Physical Systems

2.1 Notes on Automotive Grand Challenges

Despite the media success of the DARPA Urban Challenge, and the ability of many teams to complete the competition, there are a variety of significant outstanding problems not addressed by the competitors (even those who successfully finished). In fact, one could argue that the DARPA Urban Challenge does not even qualify as a Grand Challenge, because of the number of rules and regulations necessary to compete with other vehicles as a competition. These competition rules constrain designs, and do not provide incentives for out of the box thinking.

I do not pretend to be capable of organizing the DARPA challenges better, or producing better solutions than those who won or completed the competition. However, I see a few challenges on the vehicle level that are still left largely unsolved.

- Autonomous driving in *recent* suburban zones: driving to a destination using suburban maps that are out of date. The vehicle would not have access to high-resolution GPS waypoints that define the roadway, but would need to rely on alternative methods of discovering the road.
- Autonomous driving using commodity sensors: the combined value of all sensors on the autonomous vehicle cannot exceed the vehicle's MSRP.
- Autonomous driving using commodity software: the software used to drive the car must be developed with only common constraints of a vehicle (e.g., four wheels, front-tire steering).

These are *very* specific research questions, and perhaps do not qualify as grand challenges for the same reason that DARPA challenges could be described as "Autonomous driving through waypoint following and [moving] obstacle avoidance." However, when sub-meter accuracy is unavailable in GPS, and ranging sensors costing over $50,000 are unavailable, the algorithms required to solve even the waypoint following problem are drastically different—if they even exist.

2.2 Generalization to Transportation Grand Challenges

How can these automotive challenges be somewhat generalized? Let's examine some analogues in aviation.

- Automated air traffic control;
- Personal flying vehicles; and
- Verifiable software for aerial vehicles.

2.3 Cyber-Physical Systems: Transportation Grand Challenges

So, what are the fundamental research analogues? One position is:

- Self-resolving transport (routing, control, communication);
- Low-cost transport automation solutions; and
- Verifiable software for these high-confidence transportation systems.

What makes these grand challenges even more interesting is their potential effect on energy consumption. Depending on the time requirements, some methods can provide more energy efficient transportation of people and goods. The reuse of existing rail-lines to permit more traffic can reduce energy consumption per capita, while not requiring more capital investment by the government.

3 Educational Curricula and Programs

CPS curricula must be presented differently from traditional computer science curricula. Issues commonly abstracted by classical computer science—importantly, timing—are important for CPSs.

3.1 Functional Composition as Software Engineering

At the University of Arizona, senior software engineers are developing automotive software for high-level algorithms. Focus is placed on producing software that performs functionally, and provides sufficient tests to show that all requirements are met. Such focus instructs students that for each line of code there exists a requirement, and for each requirement there is a test.

3.2 Four-year Physical System Integration

Changes to the motivations for programming must be changed for future automotive designers. Introducing a physical motivation (rather than data abstraction) for software will help students fundamentally understand the "driving" forces: system requirements. This is a departure from standard abstraction instruction (sorting, trees, etc.). However, integrating computing as a tool to perform engineering tasks helps students understand that the cyber-physical parts of the system have computational relationships: either in simulation, control, or observation.

3.3 Graduate Curricula

Without a firm understanding of the issues of physical control, software modeling and development, and issues of distributed physical systems, advanced graduates will be unable to work authoritatively in the automotive domain. Curricular work in domain-specific modeling is key to this understanding—either development of customized domains, and the associated code synthesis that emerges, or in existing domain tools such as MATLAB's Real Time Workshop. Further, students will need testbeds upon which to work. The emerging requirement by DoD to automate a large portion of its vehicle fleet is a good opportunity for DoD to discover many of the fundamental results necessary, as well as produce a large assortment of researchers for DoD labs. Additional collaboration between universities and automobile manufacturers is also possible, but will not scale to the number of universities necessary to make a graduate population sustainable.

Biographical and Contact Information

Dr. Jonathan Sprinkle (sprinkle@ECE.Arizona.Edu) is an Assistant Professor of Electrical and Computer Engineering at the University of Arizona. Until June 2007, he was the Executive Director of the Center for Hybrid and Embedded Software Systems at the University of California, Berkeley. His research is in the area of intelligent autonomous systems, including UAVs, hybrid systems, and underwater vehicles. Building blocks for this are in domain-specific modeling, metamodeling, and generative programming. During 2006-07, Dr. Sprinkle was the co-Team Leader of the Sydney-Berkeley Driving Team, a collaborative entry into the DARPA Urban Challenge with partners Sydney University, University of Technology, Sydney, and National ICT Australia (NICTA). He can be reached at +1 (520) 626-0737, or at http://www.ece.arizona.edu/∼sprinkjm/