Great Fakes--Less Field Testing
The Importance of Abstraction in CPS

Jonathan Sprinkle
Grand vision for high-confidence cyber-physical systems?

- Distributed compositional systems doing time-dependent tasks

Automated Aerial Refueling
Decision-authority with human-in-the-loop autonomy
Joint work with Ding, Tomlin, Sastry
Grand vision for high-confidence cyber-physical systems?

- Distributed compositional systems doing time-dependent tasks
Grand vision for high-confidence cyber-physical systems?

- Distributed compositional systems doing time-dependent tasks
Grand vision for high-confidence cyber-physical systems?

- Distributed compositional systems doing time-dependent tasks
"Grand Challenges" to be accomplished in 5, 10 and 20 years?
"Grand Challenges" to be accomplished in 5, 10 and 20 years?

- **20 years**: Deployed and available societal CPSs, with plugins for:
  - Personal vehicle semi-autonomy (significantly safer roads)
  - Healthcare and assisted living devices are in the cell phone equivalent
  - Devices for energy-efficient homes
- **10 years**: Componentization of Cyber-Physical Platforms and Capabilities
  - “PCI” cards for your car’s infotainment and driver assistance
  - Robust algorithms for fall-detection, vitals-spikes, with special devices
  - Composable home automation devices
- **5 years**: Cyber-Physical Composition—Proof of verification
  - Model-based synthesizers can be proved and verified (*Cyber*-physical)
  - New algorithms for composition in cyber-*Physical*
  - Time-critical layers of software, to overlay on top of traditional platforms
Existing research results “ready for prime-time”, not yet adopted?

- Systems Modeling
- Create *model* of the system
- Perform: Analysis, Architecture exploration, Simulation
- Generate: Configuration, Code, Executables

*From the same models!*
Fundamental research and technology challenges

- Using implied physical models to generate software

- Discretizing software development into functional behaviors to reduce integration problems (only works if composition is fundamental!)

- Understanding impact of timing on system, and whether measurement timing, or information latency, is important

- Distributed compositional systems doing time-dependent tasks
Educational curriculum changes for the workforce of the future: with \textit{vi} operational semantics!!!

- **First year**
  - Teach C/Java/Python/Matlab/C++/FORTRAN programming
  - Focus on systems building, understanding models of computation and communication. Use programming to prove knowledge. Students use peer-code-reviews to grade each others’ work.

- **Second year**
  - Data structures or object-oriented programming, algorithms
  - Executable models, code synthesis. Foundations of composable software, including OO, synchronous languages, compilers. Students evaluate assignments (esp. software) written by previous terms’ students as regular course events.

- **Third year**
  - Principles of software engineering, advanced OO programming
  - Introduction to cyber-physical systems, verification, etc.. Students perform evaluation of a significant CPS through an established portal

- **Fourth year**
  - Capstone design
  - Capstone CPS with design. Understanding complexities of system verification, modeling and metamodeling of systems, especially in semantic anchoring
Case in point: Software Engineering Concepts (4th year)

Teams learn software engineering by understanding, and extending, a cyber-physical system: an autonomous ground vehicle

Projects:
* Laser-based multi-vehicle collision avoidance
* Threat Detection and Avoidance Modification
* Multiple Autonomous Robotic Vehicle Integrated Navigation
* 3D laser drivers and simulation

Nightly Builds by Each Team

![Dart Server on bracton.ece.arizona.edu](image)

<table>
<thead>
<tr>
<th>Available Dartboards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>teama</td>
</tr>
<tr>
<td>team0</td>
</tr>
<tr>
<td>batman</td>
</tr>
<tr>
<td>dgc-orca</td>
</tr>
<tr>
<td>gfolks</td>
</tr>
</tbody>
</table>

Pre-conditions:
* C++ experience? No.
* Linux experience? No.
* Education level: 4th year undergrad

Post-conditions:
* C++ capabilities? Yes.
* Linux experience? Yes.
* Nightly builds? Yes.
* System design? Yes.
* System investigation? Yes.
* Robust requirements changes.
Government, industry and academia working together

**Industry needs to**
- examine the horizon, and see where (if) there is a discrete jump in available personnel, or innovation, to meet the challenges
- **GIVE US OPEN EXPERIMENTAL PLATFORMS**
  - Make sharing vehicle bus data possible, if not actuation

**Government needs to**
- back its commitment to competitiveness
- improve peer review of interdisciplinary proposals, where key players are small---they’re all submitting, and can’t review!!

**Academia needs to**
- provide more students who understand CPS problems, and
- discern better what technologies are needed, and what technologies can be learned on the job
- better work across disciplines, and in-discipline
We are always looking for good graduate students.

http://ece.arizona.edu/