# Software Model Checking for Embedded Systems

SAnToS Laboratory, Kansas State University, USA  
LASER, University of Massachusetts, USA

<table>
<thead>
<tr>
<th>Principal Investigators</th>
<th>Staff</th>
<th>Students</th>
</tr>
</thead>
</table>
| Matt Dwyer             | Todd Wallentine  
                          | Stephen Siegel  
                          | Heather Conboy |
| John Hatcliff          | Robby  |
| George Avrunin         | William Deng  
                          | Oksana Tkachuk |
|                        | Jamie Cobleigh |

**Support**  
US Army Research Office  
National Science Foundation  
US Air Force Research Laboratory
At previous meetings ...

... we presented interesting technical developments made on our project in the preceding year

In support of the goals of this year’s meeting we will do something a bit different

- Explain our current vision for supporting software development for embedded systems
- Describe analysis tool support that is applicable at multiple points in that vision
- Describe an instance of the vision instantiated in a real development setting
Goals

Much of this is "non-controversial" for this audience

- High-levels of assurance
  - Functional and non-functional aspects
- Less human, more machine intensive
  - Reduce development cost/time
  - Leverage human expertise
- Evidence/artifacts that witness quality
  - Not enough to have a tool say “ok”
Features of our Vision

- Early and varied semantic modeling
  - structural modeling is useful as well
- Analysis driven feedback and refinement
- Synthesize code wherever possible
- Aspects of an agile process
  - continuous delivery of working artifacts
  - team development (human & machine)
- Exploit "domain information" throughout
  - ultimately meta-tools may be useful, but it's too premature for that
Development Flow

User’s informal requirements

Requirements Model

Query checker, Visualization tools

Consistency, Completeness, ... checker
Development Flow

User’s informal requirements → Model-specific analysis

Functional Model → Performance Model → Functional Model

Inter-model consistency, completeness, ... checking
Development Flow

- Functional Model
- Functional Model'
- Performance Model
- Conformance checker(s)
- Design Model
Development Flow

Functional Model

Functional Model'

Performance Model

Structural Design Model

Synchronization Policy Spec

Abstract Behavioral Model

Quality of Service Spec

...
Development Flow

- Functional Model
- Functional Model'
- Performance Model
- Structural Design Model
- Synchronization Policy Spec
Development Flow
Development Flow

- Structural Design Model
- Synchronization Policy Spec
- Abstract Behavioral Model
- Quality of Service Spec

...
Development Flow

- Structural Design Model
- Synchronization Policy Spec
- Abstract Behavioral Model
- Quality of Service Spec
- ...
This Vision

- Has been influenced by intensive interactions with engineers building real-time mission-critical avionics systems
  - It may be too heavily biased ...
- To make progress skip the toy problems, dive head-first into real domains
  - Painful
  - Useful (access to problems, sell methods)
- We believe we’ve learned several important lessons already ...
Lessons

- Adapt methods to developers
  - Ease of use, leverage domain abstractions
- Use layered, incremental methods
  - Low entry barrier, early and focused feedback
- Focus technology on the hard part
  - Synchronization, timing, global properties
- Synthesize as much code as possible
  - Developer buyin, reduce code-level reasoning
The rest of the talk ...

- **Bogor**
  - Tool support for analysis of behavioral software artifacts via model checking

- **Cadena**
  - An example development flow for distributed real-time embedded avionics software

- **Poster/Demo**
  - For both Bogor and Cadena this afternoon