

International Panel:
CPS Research Agenda
a personal perspective

Karl-Erik Årzén
Lund University
Lund, SWEDEN

CPS is like the World Series
in Baseball

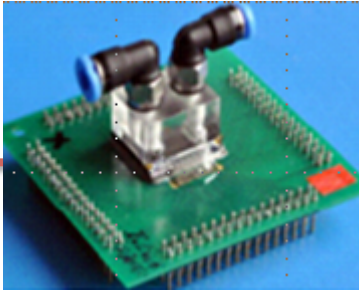


A Swedish Perspective

- No separate research programmes on CPS yet
- Swedish funding agencies will not react until the European funding agencies do
- Promoted areas:
 - Life Sciences
 - High Energy Physics
 - ESS (European Spallation Source)
 - MAX IV (synchrotron)
- The entire ICT area has problem motivating its existence as a separate area



CPS is everywhere and at all scales



ERICSSON

**Liquid-cooled
3D MPSoC**

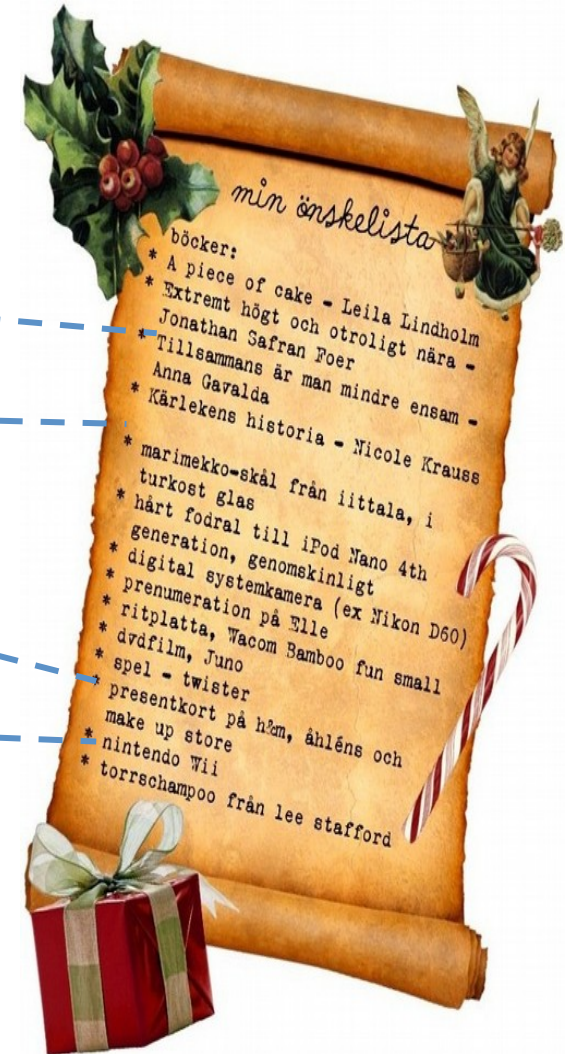
Data center

**50B interconnected
LTE devices by 2017**

CPS Research Agenda

- Model-Driven Engineering
- Modeling Tools
- Uncertainty Management
- Education

-
- Emergent behavior
 - Distributed analysis and synthesis of controllers



Model-Driven Engineering

- The vision
- Automated process from model to cyber-physical system
 - Functionally correct
 - Dependable
 - Secure
 - Resource-efficient
 - Timeliness
 -

Model-Driven Engineering for Software

In the software domain:

- Software abstraction layers tailored for different analysis and design tasks
- Property-preserving model translations and refinement mechanisms
- "The model is the software"
 - Often realistic
 - Automatic Code Synthesis
- UML

Model-Driven Engineering for Systems

In the software + hardware domain:

- SysML + architecture modeling languages such as EAST-ADL, AADL, Modelisar, ...
- Allows limited hardware modeling
- Used for modeling the hardware in which the software executes
- Not used for modeling the physical world that the system interact with
 - Too limited behavior models
- Not for CPS

Model-Driven Engineering for CPS

- The model is **not** the system
- Approximations rather than abstractions
 - E.g. reduced models, linearized models, truncated models,
- Properties not necessarily maintained
- Models have a limited validity range
- Models expire
 - Requirements change
 - The model and the reality deviate with time due to e.g., aging, wear,

Golomb on Modeling

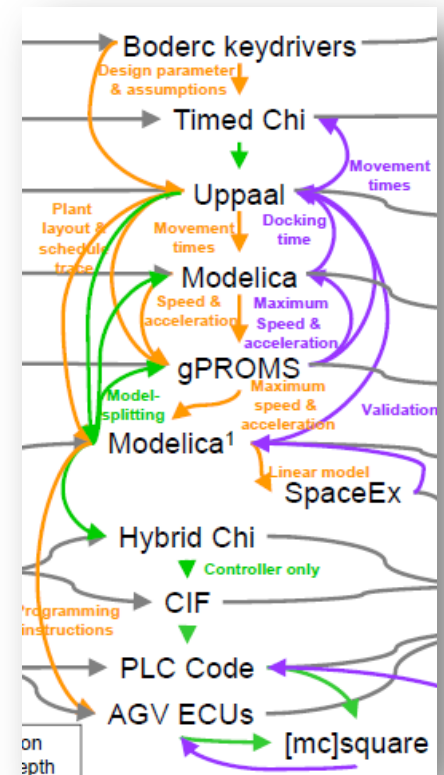
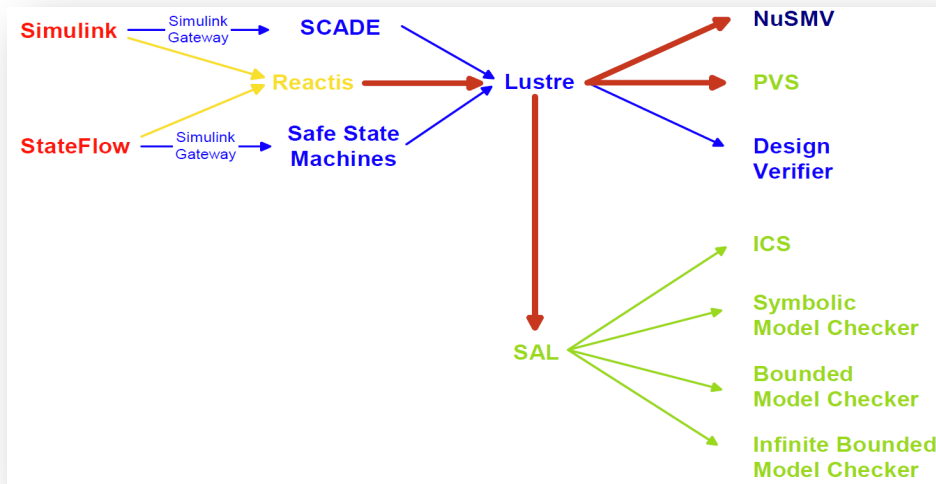
- "Mathematical Models: Uses and Limitations" – Simulation, Apr 70
- Don't apply a model until you understand the simplifying assumptions on which it is based and can test their applicability.
- Distinguish at all times between the model and the real world. **You will never strike oil by drilling through the map!**
- The purpose of notation and terminology should be to enhance insight and facilitate computation – not to impress or confuse the uninitiated



Solomon Wolf Golomb
(1932) *mathematician and engineer and a professor of electrical engineering at the University of Southern California.*

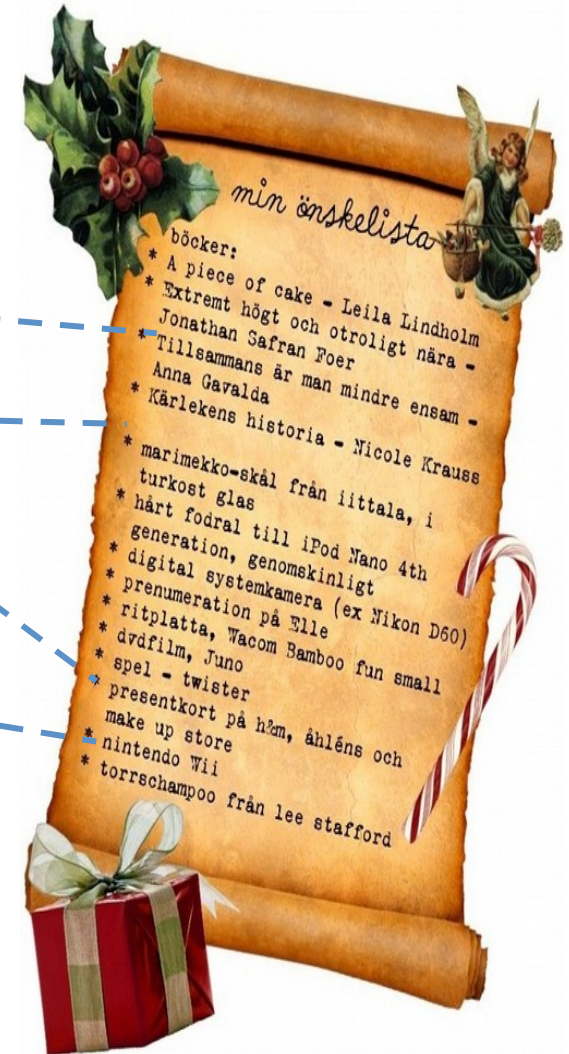
Model-Driven Engineering for CPS

- Most applications of formal methods only concern the discrete-event or discrete-time parts of the CPS (the "controller" part)
- Long and complicated tool chains



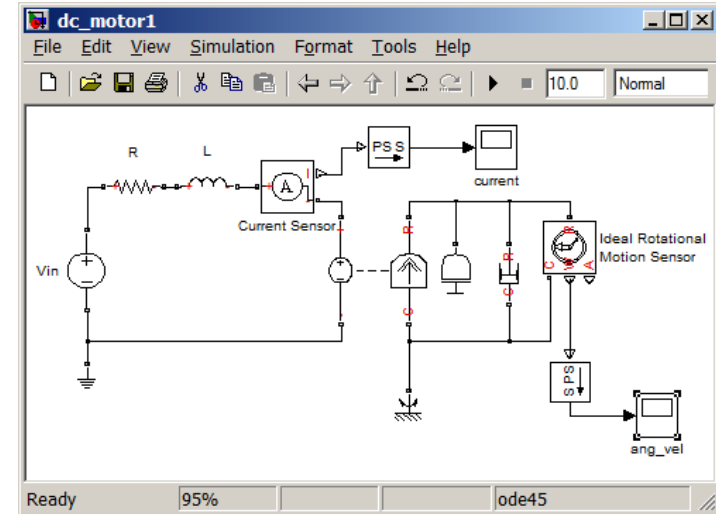
CPS Research Agenda

- Model-Driven Engineering
- Modeling Tools
- Uncertainty Management
- Education





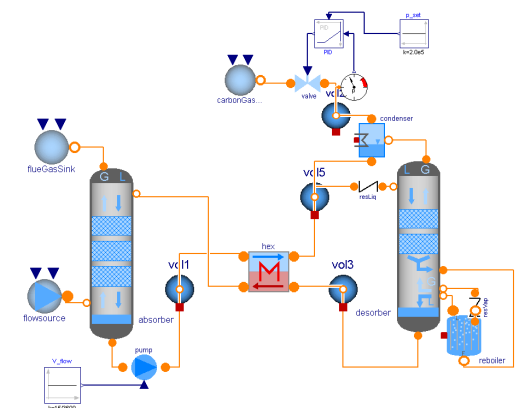
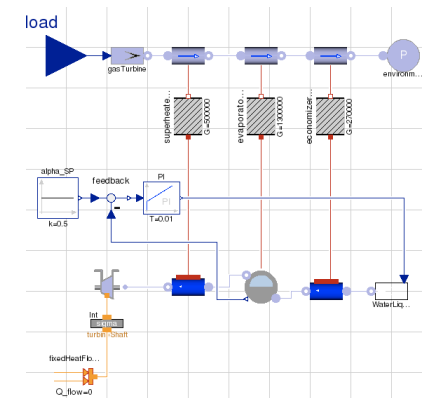
Modeling Tools

- Frameworks that allow seamless integration of tools
- Or multi-domain/multi-MoC tools
 - Ptolemy II
 - Simulink
 - S-functions allow extensions
 - SimScape → equation-based
 - SimEvents → discrete-event simulation
 - StateFlow → FSM
 - TrueTime → rt kernels + networks
- Well documented semantics
- Equation-based DAE languages have many advantages for the physical parts
 - Modelica, SimScape, Acumen, ..
 - However physical system modeling is difficult
 - Large threshold
 - Problems with high index, initializations, efficient code



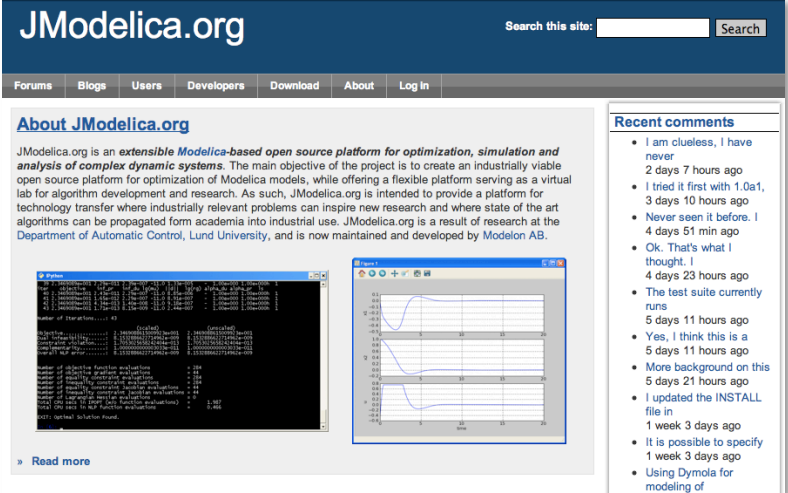
Modelica

- The most mature equation-based DAE language
- Several commercial tools, e.g. Dymola
- Two open source tools
 - OpenModelica 
 - JModelica 
- Discrete-time and discrete-event parts recently redesigned based on synchronous language ideas
 - Well-defined semantics based on clock inference
- Automatic code generation



JModelica

- Java + C + Python
- JastAdd
 - AspectJ
 - Reference attribute grammars
- Optimica
 - Language extension for representing optimization problems
- Lund University + Modelon



JModelica.org Search this site: Search

Forums Blogs Users Developers Download About Log In

About JModelica.org

JModelica.org is an *extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems*. The main objective of the project is to create an industrially viable open source platform for optimization of Modelica models, while offering a flexible platform serving as a virtual lab for algorithm development and research. As such, JModelica.org is intended to provide a platform for technology transfer where industrially relevant problems can inspire new research and where state of the art algorithms can be propagated from academia into industrial use. JModelica.org is a result of research at the Department of Automatic Control, Lund University, and is now maintained and developed by Modelon AB.

[Read more](#)

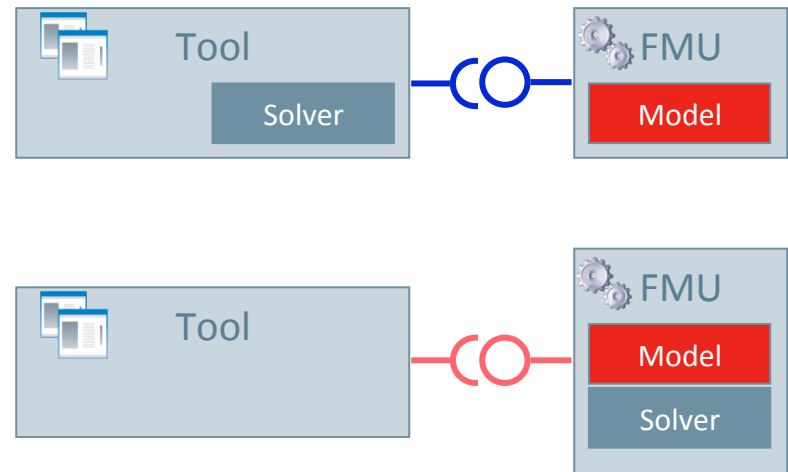
Recent comments

- I am clueless, I have never
2 days 7 hours ago
- I tried it first with 1.0a1,
3 days 10 hours ago
- Never seen it before. I
4 days 51 min ago
- Ok. That's what I
thought. I
4 days 23 hours ago
- The test suite currently
runs
5 days 11 hours ago
- Yes, I think this is a
5 days 11 hours ago
- More background on this
5 days 21 hours ago
- I updated the INSTALL
file in
1 week 3 days ago
- It is possible to specify
1 week 3 days ago
- Using Dymola for
modeling of

```
optimization VDP_Opt(objective=cost(finalTime),
                    startTime=0,
                    finalTime(free=true, initialGuess=1))
VDP vdp(u(free=true, initialGuess=0.0));
Real cost (start=0);
equation
der(cost) = 1;
constraint
vdp.x1(finalTime) = 0;
vdp.x2(finalTime) = 0;
vdp.u >= -1; vdp.u <= 1;
end VDP_Opt;
```

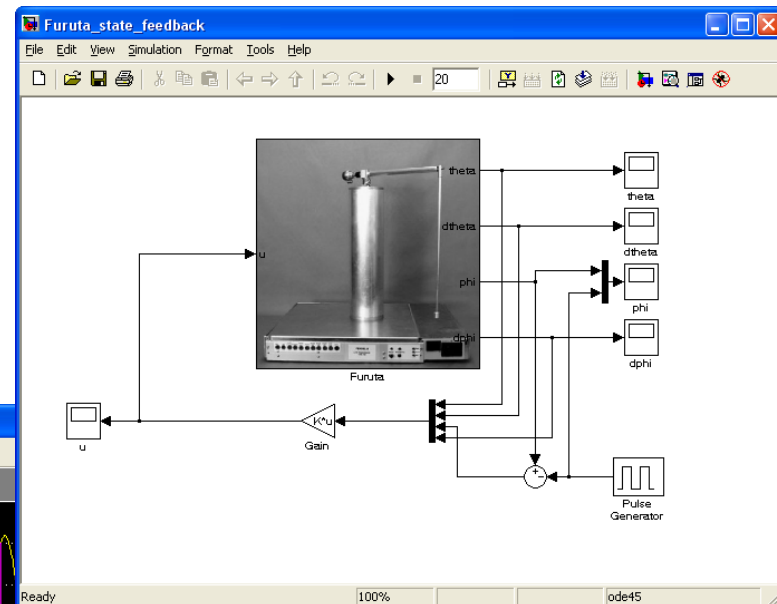
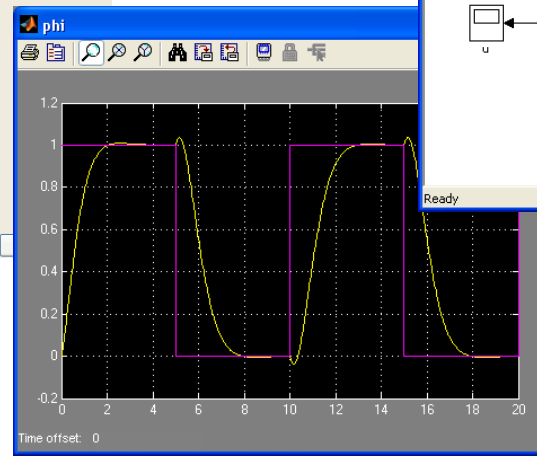
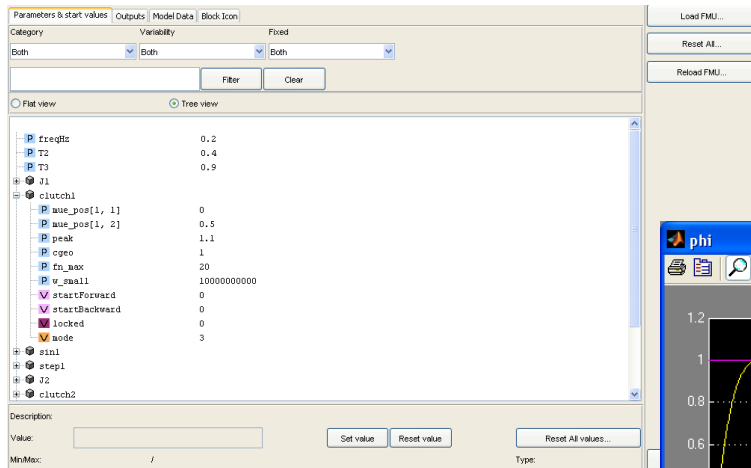
Functional Mock-Up Interface (FMI)

- Model exchange standard
 - Open source
 - Non-proprietary
 - cp. S-functions
- Model exchange
- Co-simulation



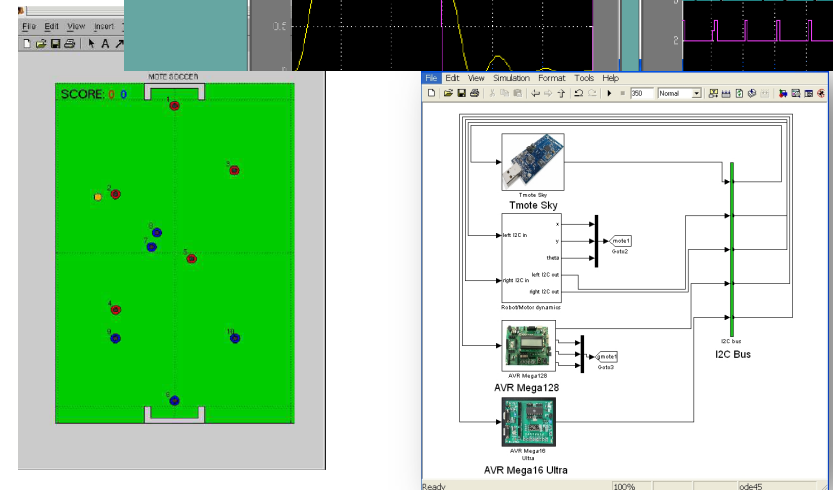
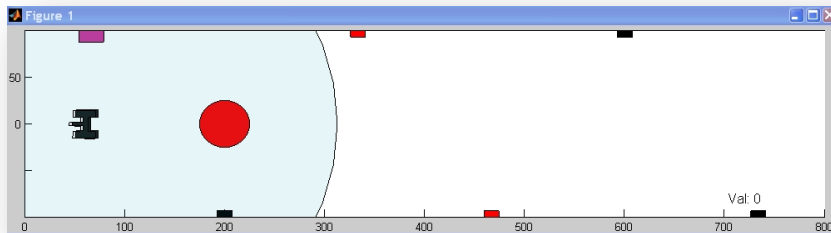
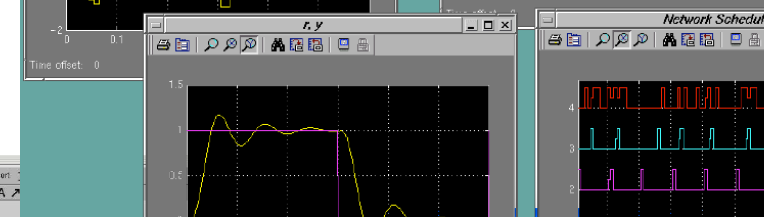
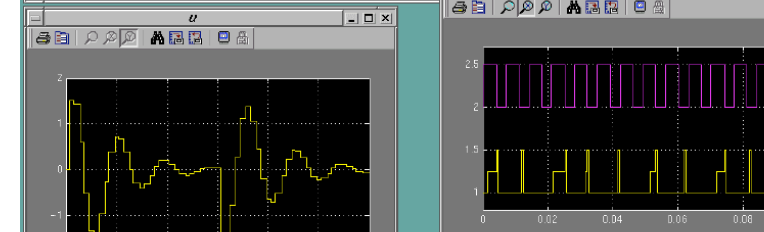
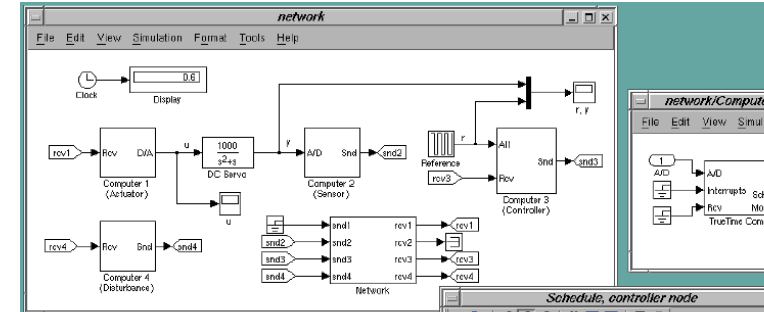
Simulation of Modelica models in Simulink

- Import of compiled Modelica models (FMUs)
 - From Intelligent editor
 - From other FMI compliant tool such as Dymola or Simulation X



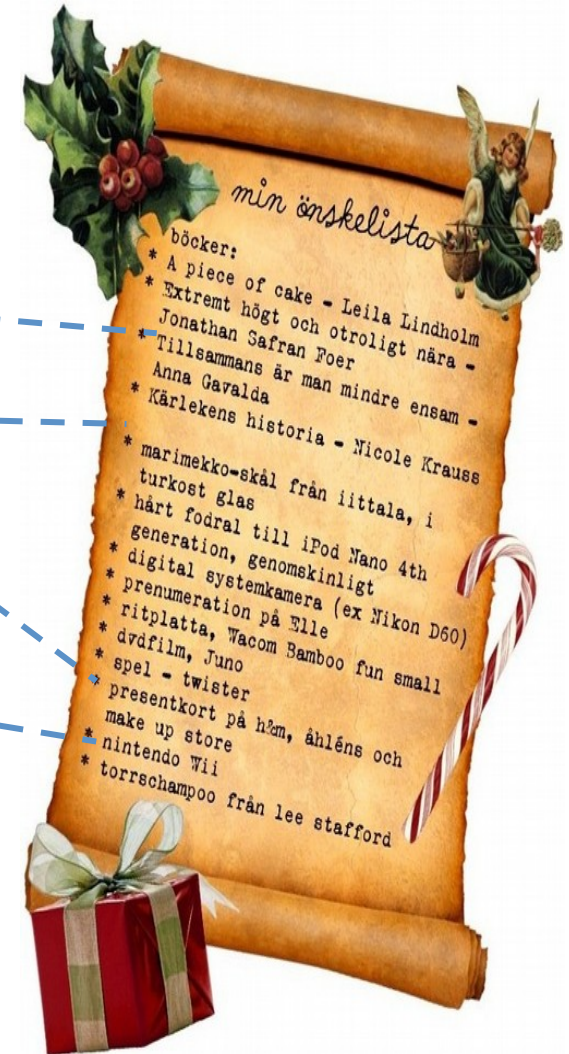
TrueTime

- Co-simulation of controller task execution, network transmissions, and continuous-time plant dynamics
- Simulink using S-functions
- Currently ported to FMI
 - Modelica
 - Joint project with Vanderbilt within DARPA AVM



CPS Research Agenda

- Model-Driven Engineering
- Modeling Tools
- Uncertainty Management
- Education



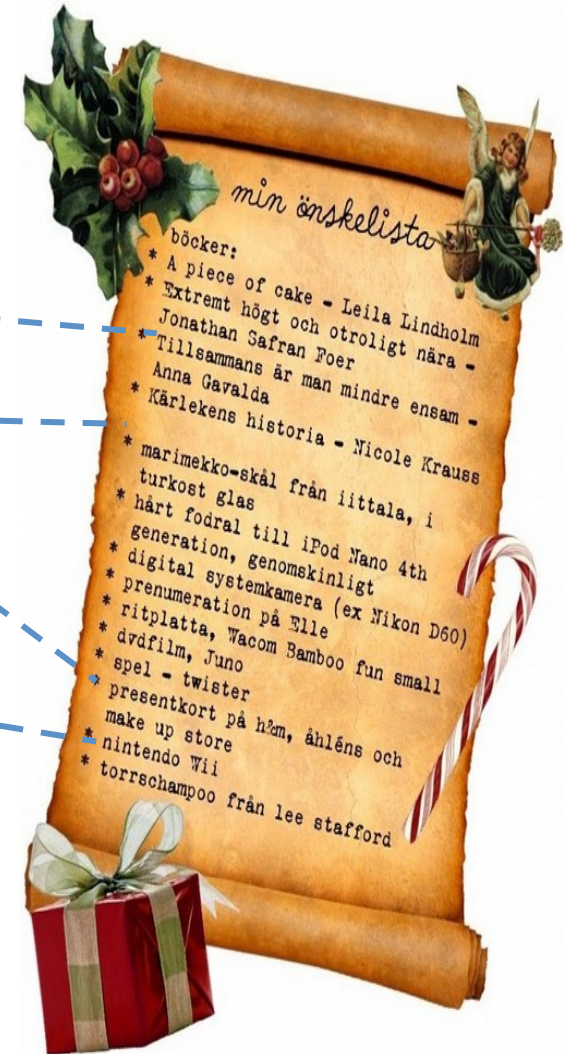
Manage uncertainty

- Resource-sharing in our implementation platforms
 - Cores sharing caches
 - Threads sharing cores
 - Applications sharing computers
 - Communication links sharing
 - media
- Inherent in the physical domain
- Modeling, analysis, simulation, verification,
- Robustness and resilience



CPS Research Agenda

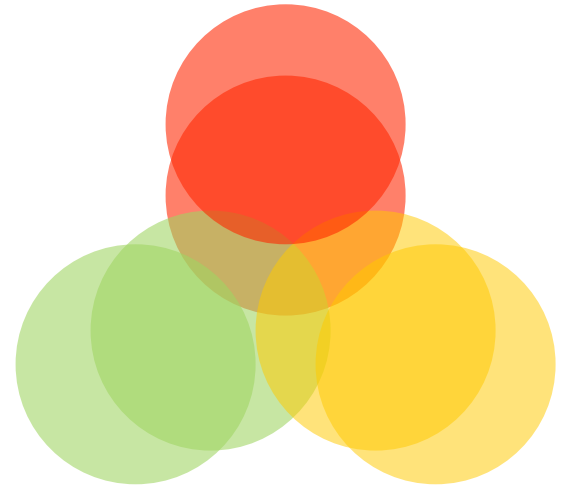
- Model-Driven Engineering
- Modeling Tools
- Uncertainty Management
- Education



Who should do the research?

CPS is in the Overlap

- Integration-Based Design
- Codesign
- Cross-layer design



- Current education system produce specialists
 - “know a lot about little”
- Only generalists won’t work
 - “know little about a lot”
- CPS need persons that know a lot in at least two disciplines
 - Can we produce these in reasonable time?

Managing Complexity

