Science of System Integration for CPS NSF CPS Program (Class 2010)

Vanderbilt University, ISIS; University of Notre Dame, EE/ME
University of Maryland, ISR; General Motors Research

Project Overview
Janos Sztipanovits

CPS PI Meeting
16-17 November, 2015
Arlington, VA

http://cps-vo.org/group/soi

Project Investigators

Faculty

- *Panos J. Antsaklis (ND)*
- *John S. Baras (UMD)*
- Getachew Befekadu (ND)
- Bill Goodwine (ND)
- Vijay Gupta (ND)
- Gabor Karsai (VU)
- *Xenofon Koutsoukos (VU)*
- *Janos Sztipanovits (VU)*
- *Shige Wang (GMR)*

Postdocs

- Mark Yampolskiy (VU)
- Peter Horvath (VU)
- Vladimir Ivanov (UMD)
- Shah-An Yang (UMD)
- Ion Matei (UMD)
- Yue Wang (ND)
- Surya Shravan (ND)

Project Motivation

- **System integration**: implemented/manufactured components are connected and system-level properties are verified/tested
  - High risk – many fundamental problems surface during system integration
  - Ad-hoc – 'making it work somehow' attitude
  - Expensive

Scientific Challenge: Foundations for Correct-by-Construction Design

Goal: extend the limits of "correct-by-construction" design:

- **in *broad sense*: model- based design process that leads to manufacturable CPS products with desired properties**

Challenges

Two major challenges in CPS to advance correct-by-construction:

1. Composition in heterogeneous domains

2. Multi-modeling with abstractions for modeling cross-domain interactions

SoI Project Goals

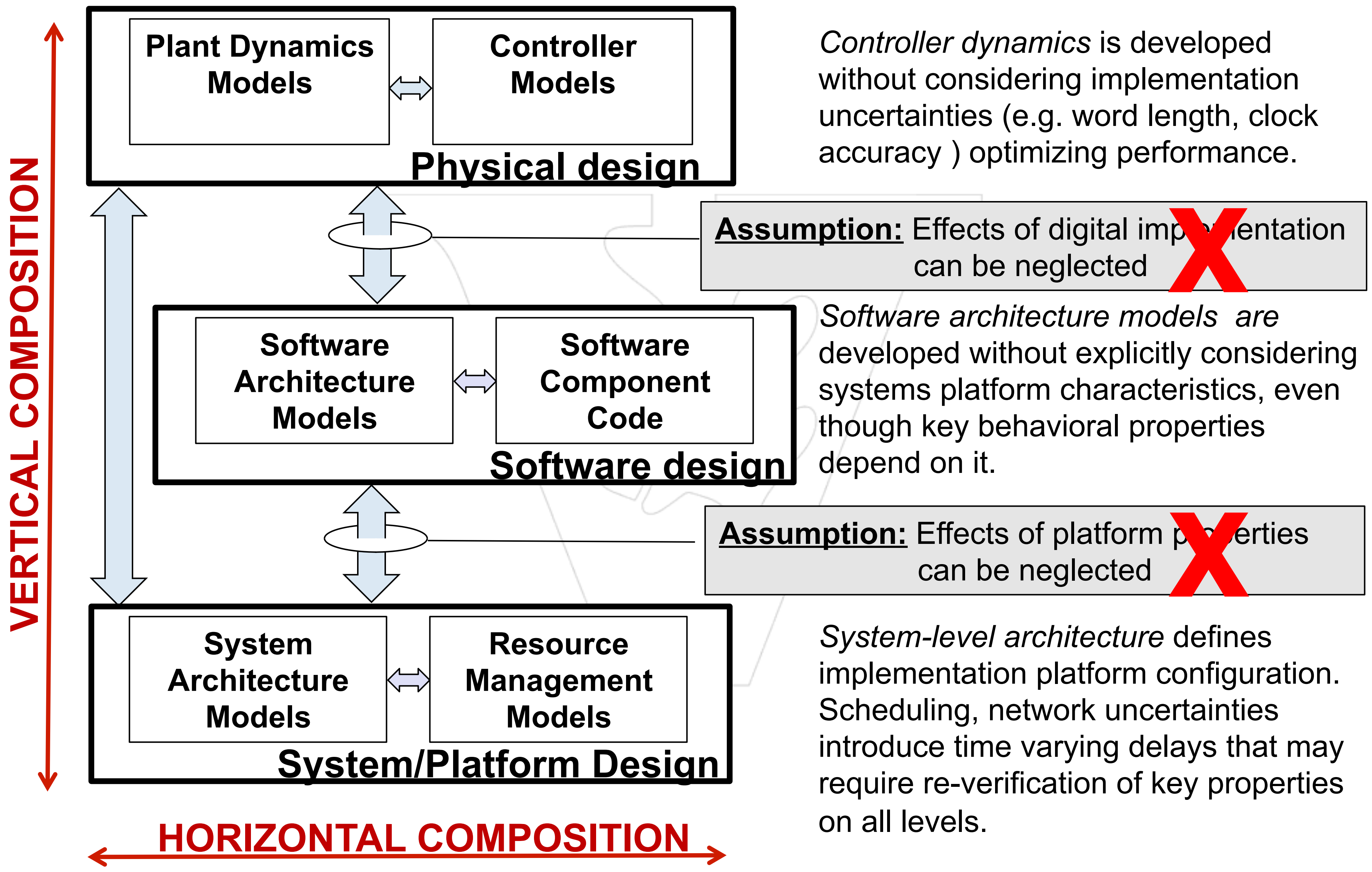
1. Investigate **composition and compositionality of heterogeneous systems** to achieve constructivity and predictability in CPS integration
2. Construct **tool chains for CPS design** based on semantically rigorous methods to define and compose of heterogeneous modeling languages.
3. Experimental validation of the ideas in **automotive and other applications.**
4. **Education methods**.

Challenge 1: Composition in Heterogeneous Domains

Composition is achieved for selected properties {P} under the following two conditions:

- Compositionality: system-level properties $\{P\}_S$ can be computed from the properties $\{P\}_{Ci}$ of the components
  - Cyber frameworks: e.g. BIP (Sifakis, 2005); Ptolemy-2 (Lee, 2003)
  - Physical frameworks: e.g. Behavioral approach (Willems, 2007); Port-Hamiltonian Approach (Duindam et.al. 2009)
  - Heterogeneous framework: e.g. Passivity-based design (this project)
- Composability: components preserve their properties $\{P\}_C$ in the composed systems
  - Physical components: requires proving that a component C in system S in all environments remains in the valid region of its state space.
  - Cyber components: requires understanding implications of resource sharing across components

Integration Across Abstraction Layers: Much Unsolved Problems



Controller dynamics is developed without considering implementation uncertainties (e.g. word length, clock accuracy) optimizing performance.

Assumption: Effects of digital implementation can be neglected

Software architecture models are developed without explicitly considering systems platform characteristics, even though key behavioral properties depend on it.

Assumption: Effects of platform properties can be neglected

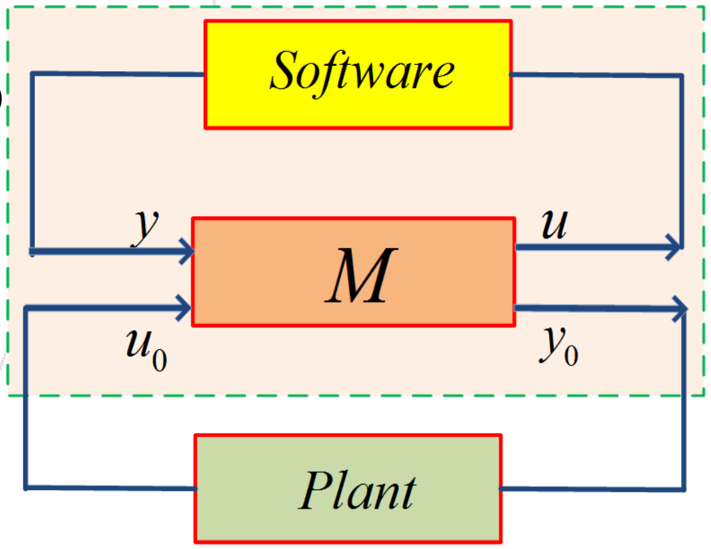
System-level architecture defines implementation platform configuration. Scheduling, network uncertainties introduce time varying delays that may require re-verification of key properties on all levels.

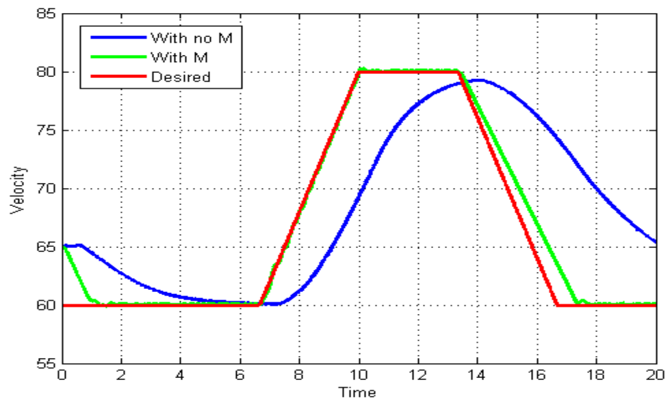
Contributions to Challenge 1: Decoupling

- Dissipativity and Passivity to decouple implementational side effects (Antsaklis, Gupta)
- Utilizing symmetry to increase robustness (Goodwin, Baras)
- Applications of decoupling to drastically decrease complexity of verifying cross-cutting properties, such as stability. (Koutsoukos)
  - One of our major goals was to demonstrate the gain in automotive domain

Key Results: Passivation Method from Control to Software

- When the control algorithm is implemented on networked computing platforms, time-varying delays may not be avoided.
- The transformation matrix M can be used to reduce the delay effects and improve the closed-loop system performance.
- By appropriate choices of the matrix M, desired passivity indices for the closed-loop system can be guaranteed
- Carsim + Simulink
  - M can be found by minimizing the tracking error via non-gradient optimization method.



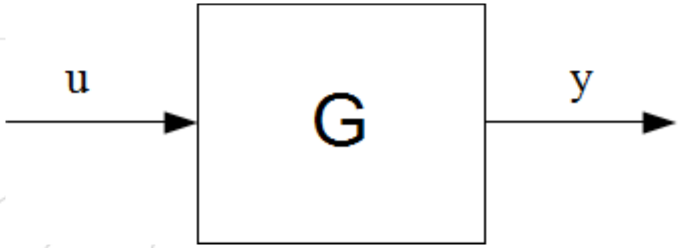


$$M = \begin{bmatrix} 0.25 & 0.19 \\ 12.84 & 5.41 \end{bmatrix}$$

Key Results: Passivity/Dissipativity in Discrete Time, Hybrid and DES Systems

- Passivity can also be defined for discrete-time systems. Consider a nonlinear discrete time system

$$x(k+1) = f(x(k), u(k))$$
$$y(k) = h(x(k), u(k)).$$



- This system is *passive* if there exists a continuous storage function *V(x)* ≥ *0* such that

$$\sum_{k=k_1}^{k_2} u^T(k)y(k) + V(x(k_1)) \geq V(x(k_2))$$

for all k_1, k_2 and all inputs *u(k)* ϵ *U*.

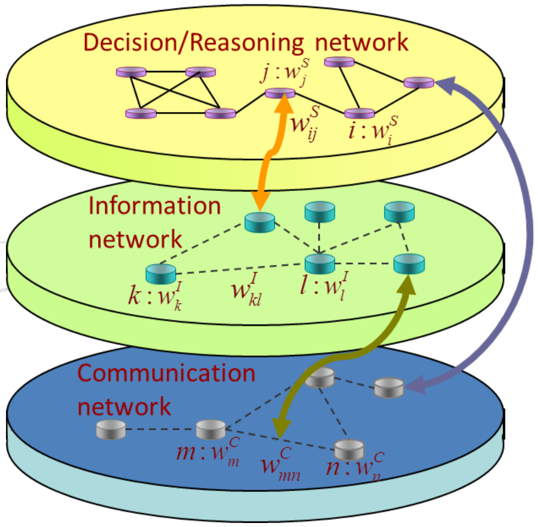
- DES abstractions of continuous systems. Define granularity to preserve passivity.
  [Sajja, Gupta and Antsaklis, ISIS-2014-005]
-

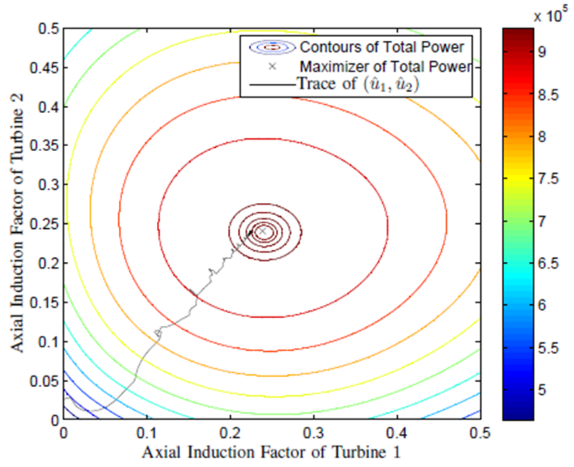
Key Results: Symmetry Methods for CPS

- Goodwin et.al. extensively studied methods for proving *invariance* of various property for *symmetric CPS*
  - A symmetric system has repeated instances of identical components.
  - Equivalence class of symmetric systems.
  - Desirable properties remain as more components are added to the system.
  - If components fail and the system reconfigures in a symmetric manner, the property persists.
  - Reduced-models preserve qualities such as stability
- Main Results
  - If a symmetric system is stable with additional proprties related to a Lyapunov function, then any equivalent symmetric system is also stable.
  - Results for approximately symmetric systems.
  - Bifurcation of optimal solutions for symmetric ssytems.
  - Reduced-order modeling using fractional-order differential equations for multi-agent systems.

Key Results: Symmetry Methods for CPS

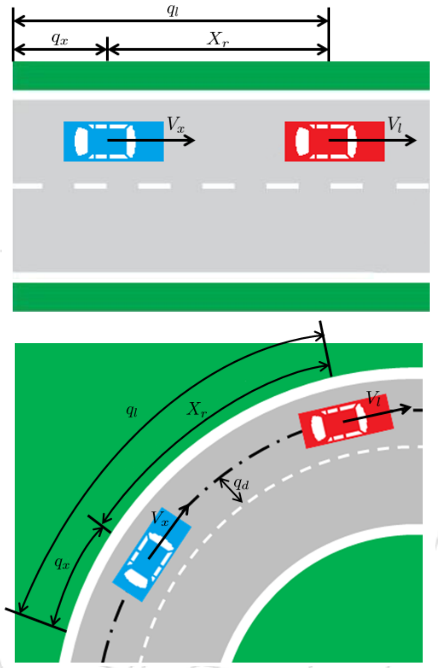
- Baras, 2012-2014, developed fundamental model for *networked CPS* involving multiple interacting hypergraphs
- Baras and Menon, 2013-2014, developed fundamental results for networked CPS demonstrating the interactions between the control and communication graphs: agents learn what is best for the team
  - Delineated contributions of direct and indirect (i.e. via action effects) communications -- Interaction and communication graphs.
  - No functional form of utilities required – only measured utilities!
- Simple distributed algorithm achieves system optimal. Applied to wind farms management. Related to extremum seeking controls.

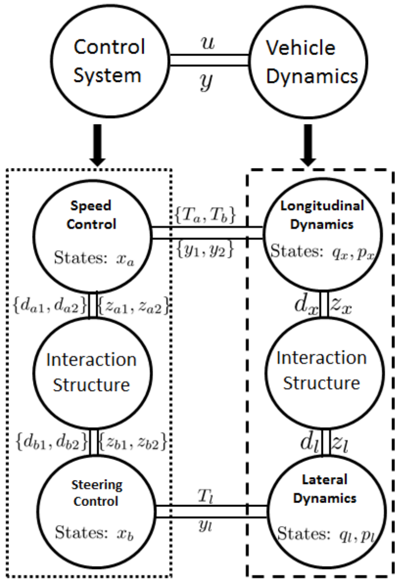


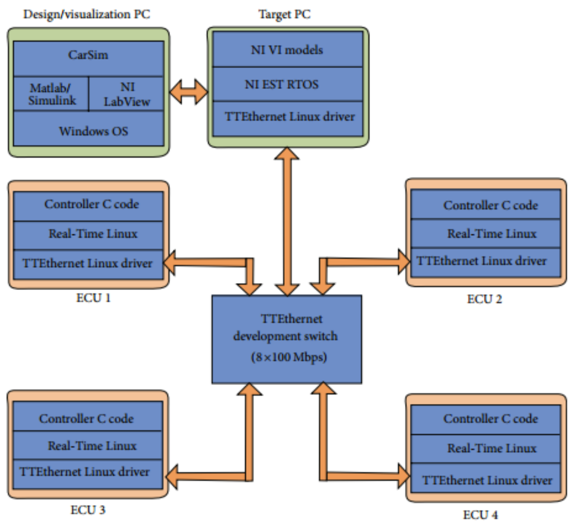


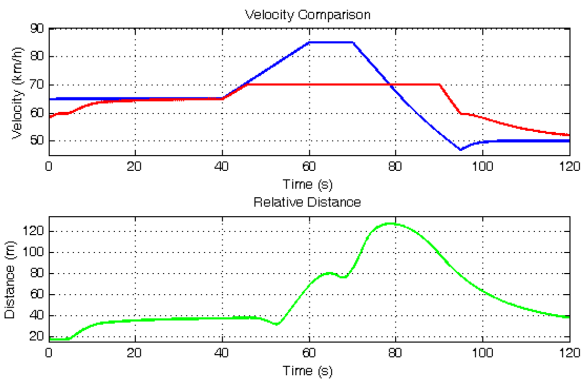
Key Results: Automotive Control Design Using Port-Hamiltonian Systems

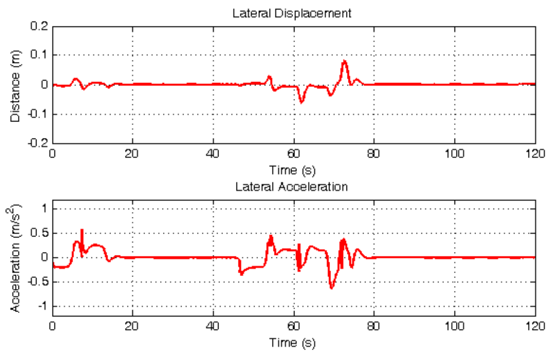
- Compositional control design based on passivity
  - Adaptive cruise controller
  - Lane keeping controller
- Stability and safety analysis
- Model-based control software design and deployment
- Demonstration and testing using HiL Simulation











Koutsoukos, Karsai, Wang

Challenge 2: Tool Chains for CPS Design

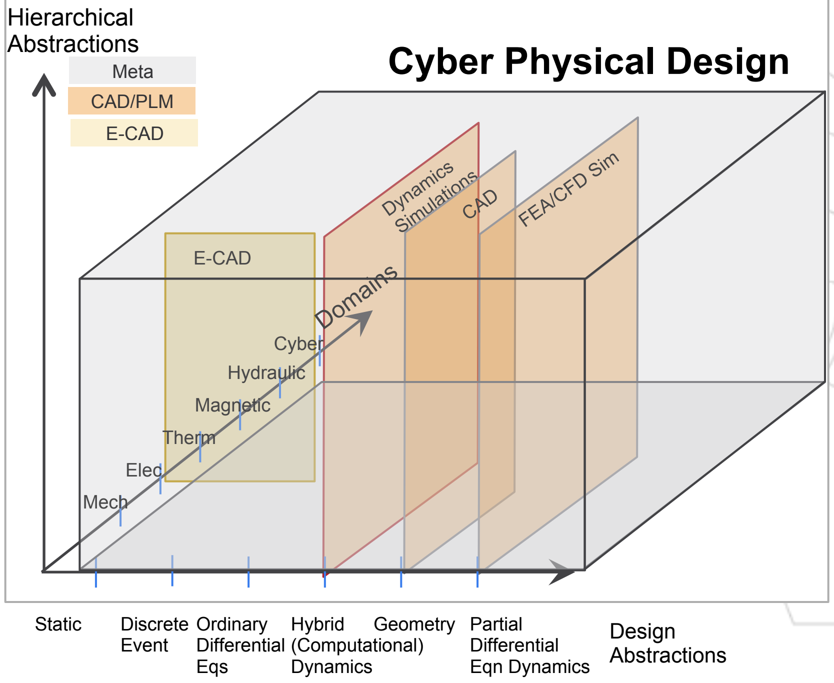
"Separation of concerns" is the state of practice in CPS and one of the primary approach to manage complexity:

- CPS design flows typically separate physical domains, physical and computational domains, abstraction layers;
- use domain specific composition and verification theories and methods while neglecting cross-domain interactions and interdependences
- pay the price at system integration

Challenge: multi-physics, multi-abstraction and integrated cyber-physical design flows that incorporates modeling cross-domain interactions

CPS Design Domains and Tools

Heterogeneous Domains & Abstractions: **Model Integration**



Heterogeneous Tools & Asset Libraries: **Tool Integration**

NGSPICE, DYMOLA, MATLAB SIMULINK, SIMULIA ABAQUS, MODELICA, creo™ A PTC Product, Adams

Integrated Engineering Tools

Key Results: Horizontal Integration Platforms for CPS Design Automation

Model Integration Platform

- Information architecture
  - DSMLs
  - Taxonomies
  - Semantics
- Model Integration Lng.s

Tool Integration Platform

- Tools as Components
  - Data Interface Std.s
  - Semantic If. Std.s
- Transformations

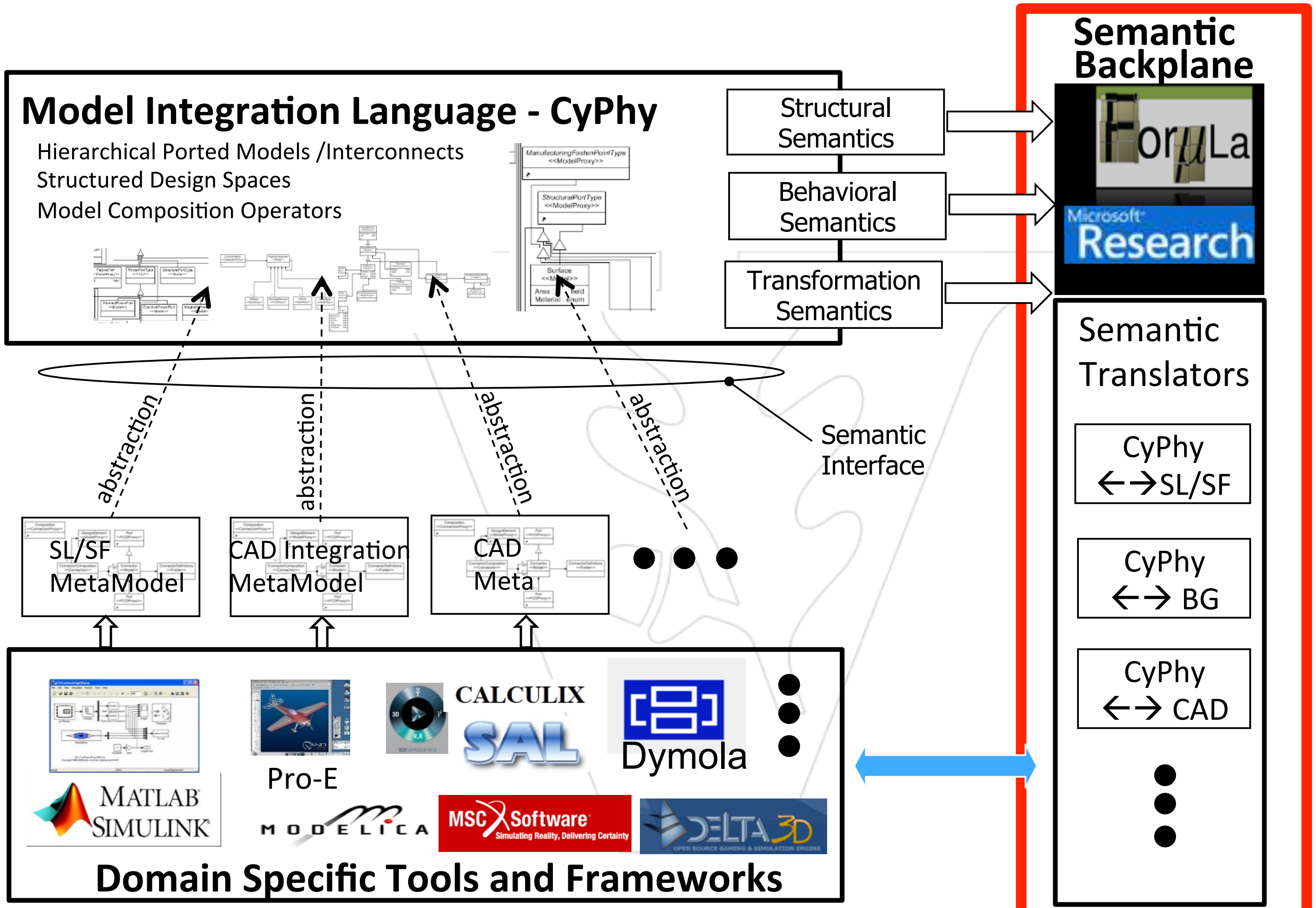
Analysis and Execution Integration Platform

- Component Exchange
- Design Data Mgmt
- SW as a Service model
- Security model

Collaboration Platform

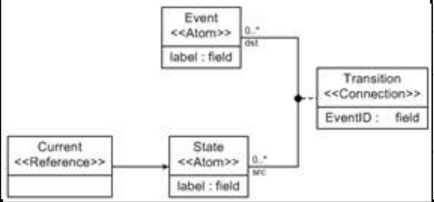
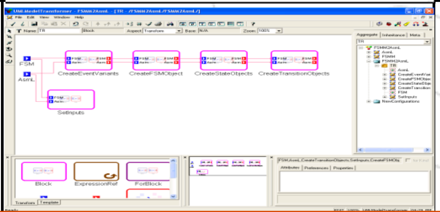
Horizontal Integration Platforms cut across traditionally isolated design domains.

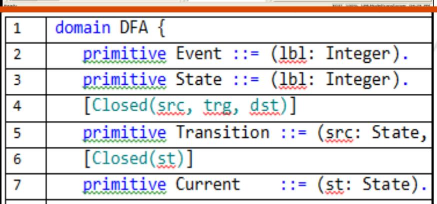
Key Results: Model Integration Languages

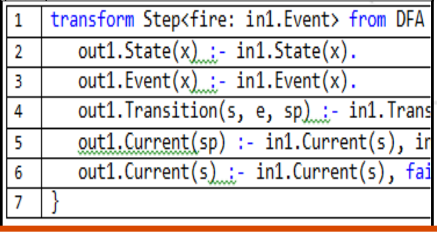


Foundation for MILs: Formal, composable semantics and semantic interfaces

Key Results: Semantic Backplane

Functions	(Meta)Models	Languages	Tools	Role
Metamodeling	 Event <<Atom>> label : field 0..* dst Transition <<Connection>> EventID : field Current <<Reference>> State <<Atom>> label : field 0..* src	MetaGME	• GME • MetaGME-2-Formula	• DSML spec. • Constraint Checking • Metaprog.
Transformation Modeling		UMTL	• GReAT • UDM	• Transf. spec. • Compiling spec to transformer

Formal Metamodeling	`domain DFA {`
	`    primitive Event ::= (lbl: Integer).`
	`    primitive State ::= (lbl: Integer).`
	`    [Closed(src, trg, dst)]`
	`    primitive Transition ::= (src: State,`
	`    [Closed(st)]`
	`    primitive Current     ::= (st: State).`
		Formula (Jackson, MSR)	• Domain Comp.
			• Trace Gen.
				• Metamod. checking
				• Example gen.
				• Semantic units

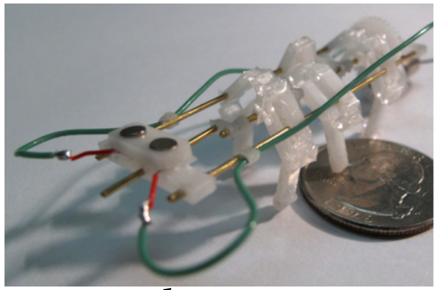
Formal Transformation Modeling	`transform Step<fire: in1.Event> from DFA`
	`    out1.State(x) :- in1.State(x).`
	`    out1.Event(x) :- in1.Event(x).`
	`    out1.Transition(s, e, sp) :- in1.Trans`
	`    out1.Current(sp) :- in1.Current(s), in`
	`    out1.Current(s) :- in1.Current(s), fai`
	`}`
			• Semantic Anchoring	• Semantics for complex DSMLs
				• Compositon

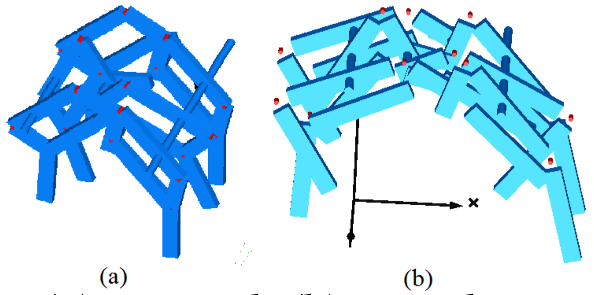
Experimental Research Component

- **Objectives**
  - Validation of results
  - Measuring progress using challenge problems
  - Contribution to education mission of the CPS Program
- Automotive Open Experimental Platform for Integration of Control Software in cooperation with GM Research
- End-to-end model based control software synthesis and integration
- Distributed virtual prototyping environments (C2WT) with FMU-CS
- Demonstrations and pilots
  - Zero-energy building
  - Smart grid
  - Robotics
  - Drivetrains
  - Manufacturing
  - Electronics

Experimental Research Component: Cyber & Physical Co-design

- Baras and Zhou, 2013, developed design of microrobots as CPS, employing rigorous **model integration** and **co-design** of cyber and physical parts: geometry, material, control law

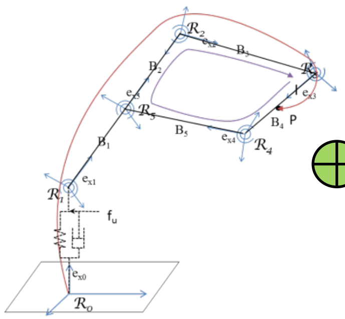

microrobot prototype



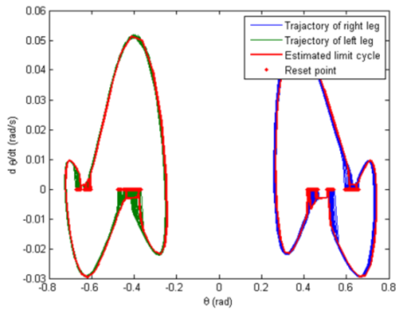
(a) original, (b) new, design

Resulted in superior performance, speed of motion, stability

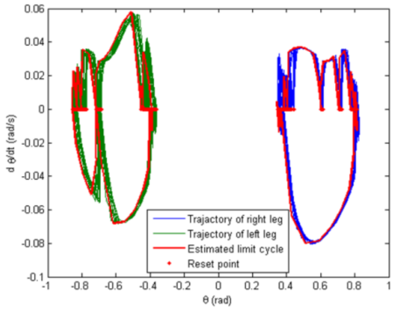
  - Integration of kinematics, dynamics, ground contact, power source, material properties, control law and limit cycle models
  - Leg geometry and material as design variables !



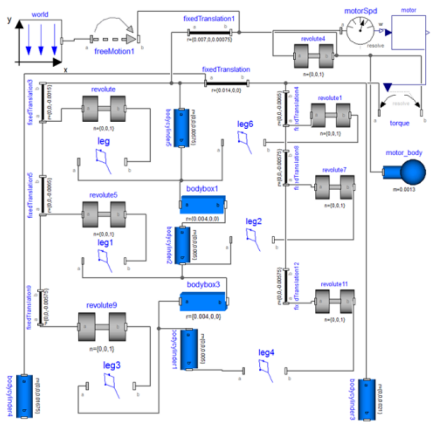
$\oplus$



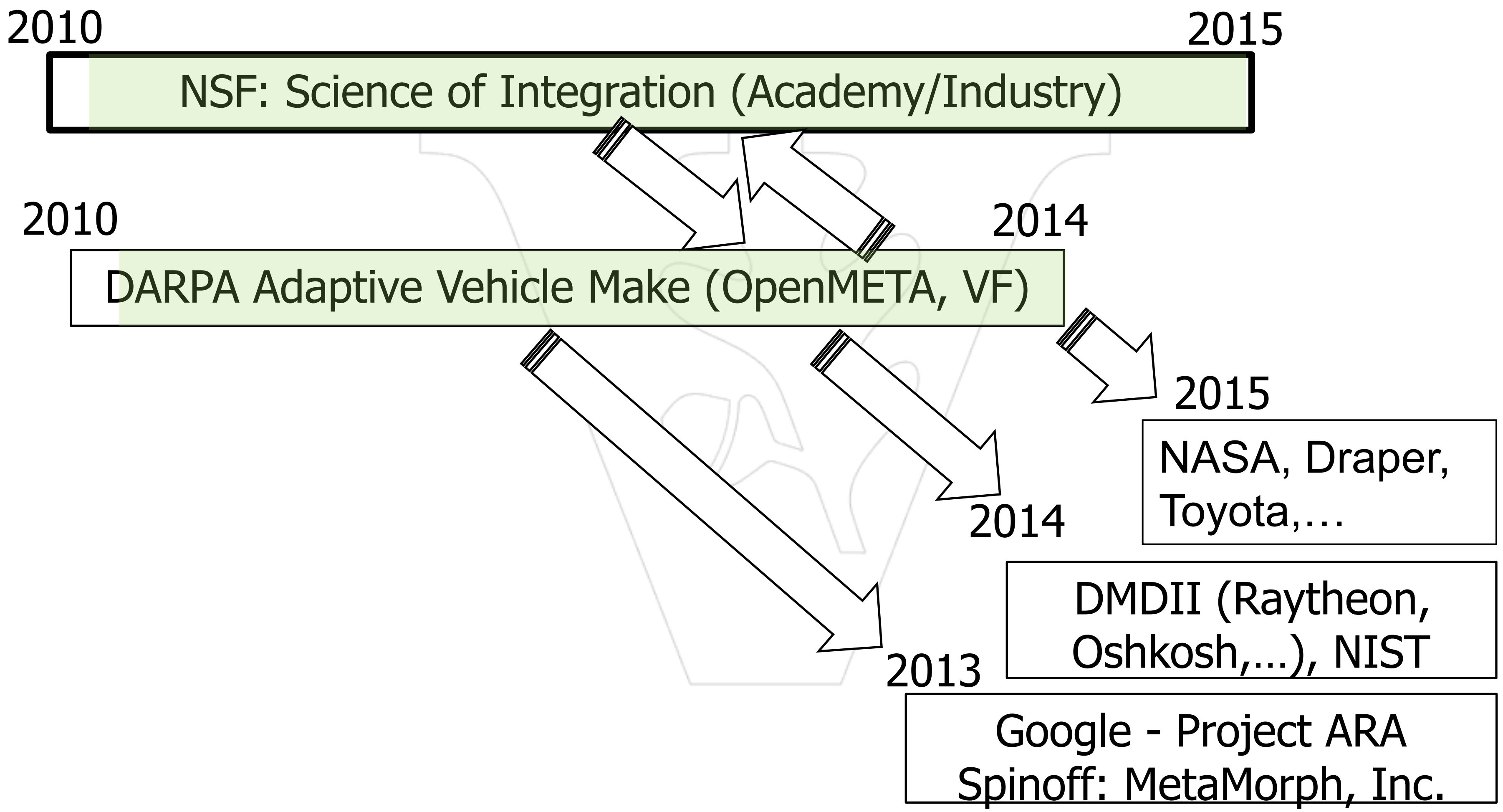
$\oplus$



$\oplus$

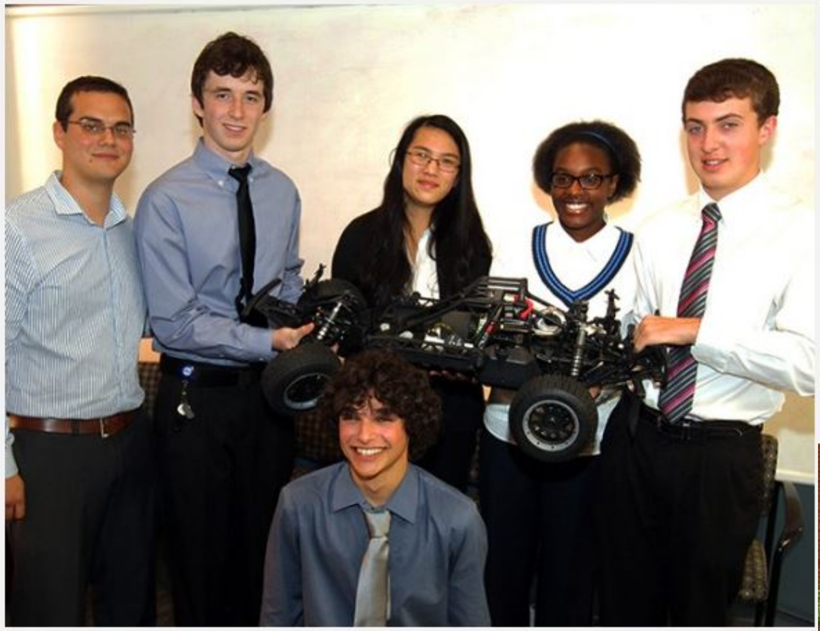


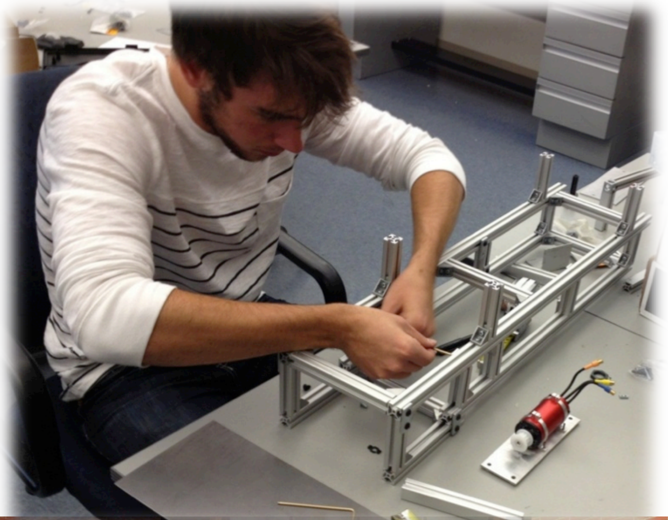
Transitioning Programs

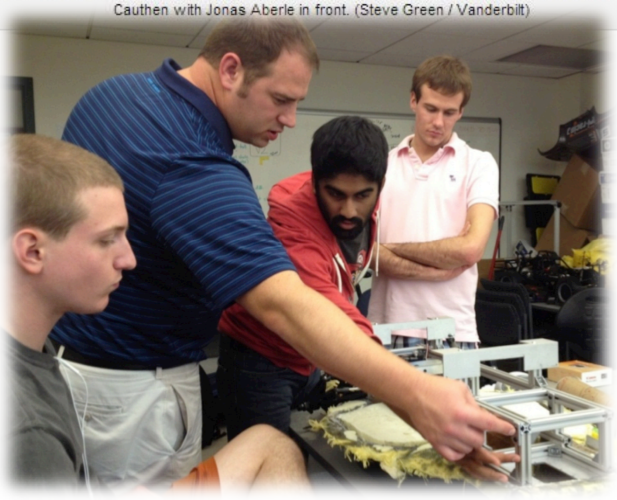

2010
2015
NSF: Science of Integration (Academy/Industry)
2010
2014
DARPA Adaptive Vehicle Make (OpenMETA, VF)
2015
NASA, Draper, Toyota,…
2014
DMDII (Raytheon, Oshkosh,…), NIST
2013
Google - Project ARA
Spinoff: MetaMorph, Inc.

Teaching CPS Design Via Practice: It Works!

High School Internship Team - 2013


Program coordinator, Brandon Knight, left, poses with interns Michael Eden, Sydney Bailes, Asha Elsberry and Lucas Cauthen with Jonas Aberle in front. (Steve Green / Vanderbilt)





Undergraduate Internship Team - 2013

Acknowledgement

PI-s gratefully acknowledge the support of the National Science Foundation under award # CNS-1035655.