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

Review Meeting
23 April, 2013
ISIS-Vanderbilt, Nashville, TN

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)
- Yuan Xue (VU)

Postdocs

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

Project Motivation

- **System integration**: implemented 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
  - *Fundamental problem – limited composability and compositionality in heterogeneous systems lead to lack of constructivity in system design*

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**

- in n*arrow sense*: use architectures (design invariants) that guarantee certain properties

Correct-by-Construction in CPS

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

1. Multi-modeling with abstractions for modeling cross-domain interactions
2. Composition in heterogeneous domains
3. Validation and Verification

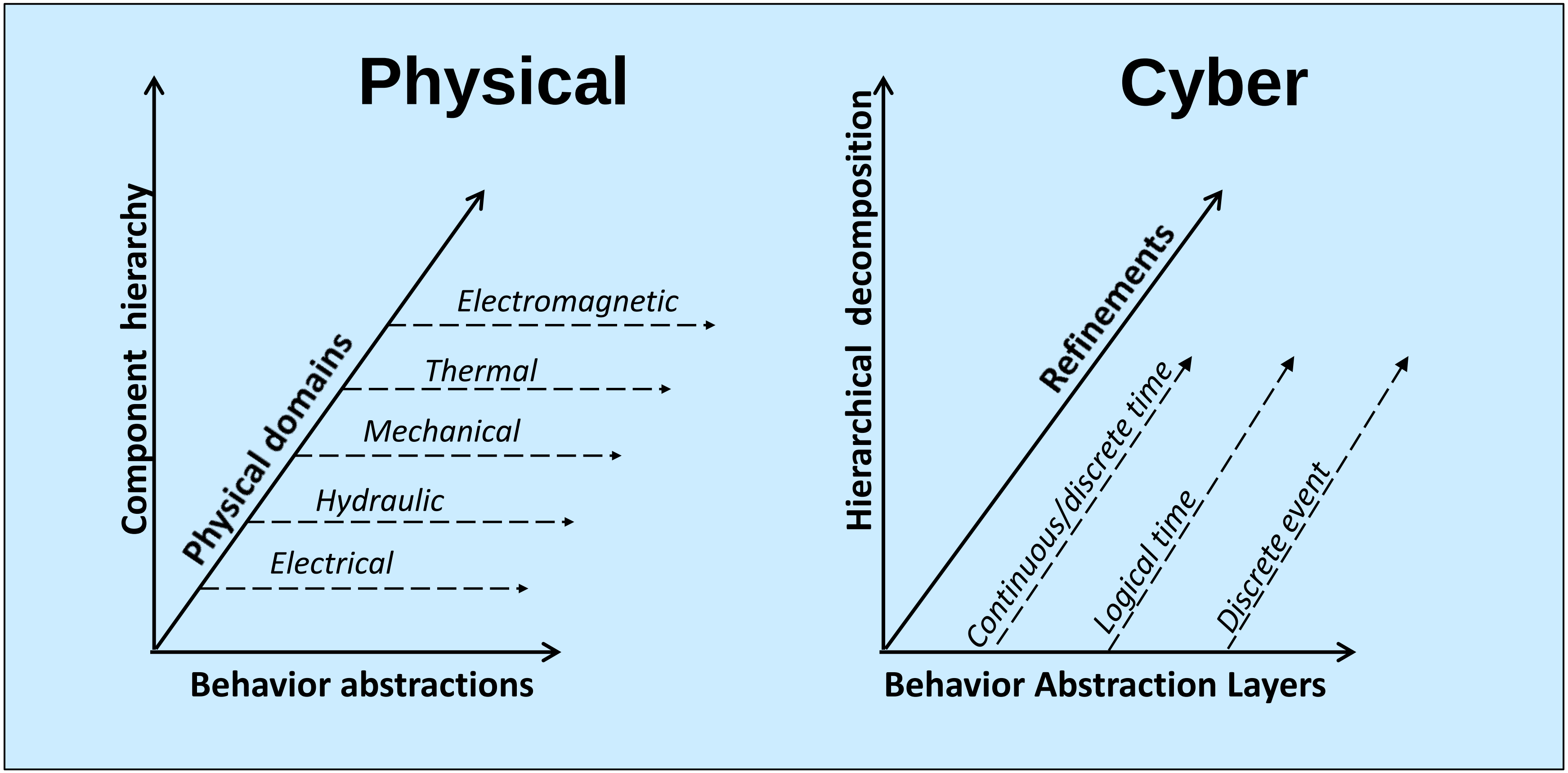
Challenge 1: Abstractions for Cross-domain Modeling

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

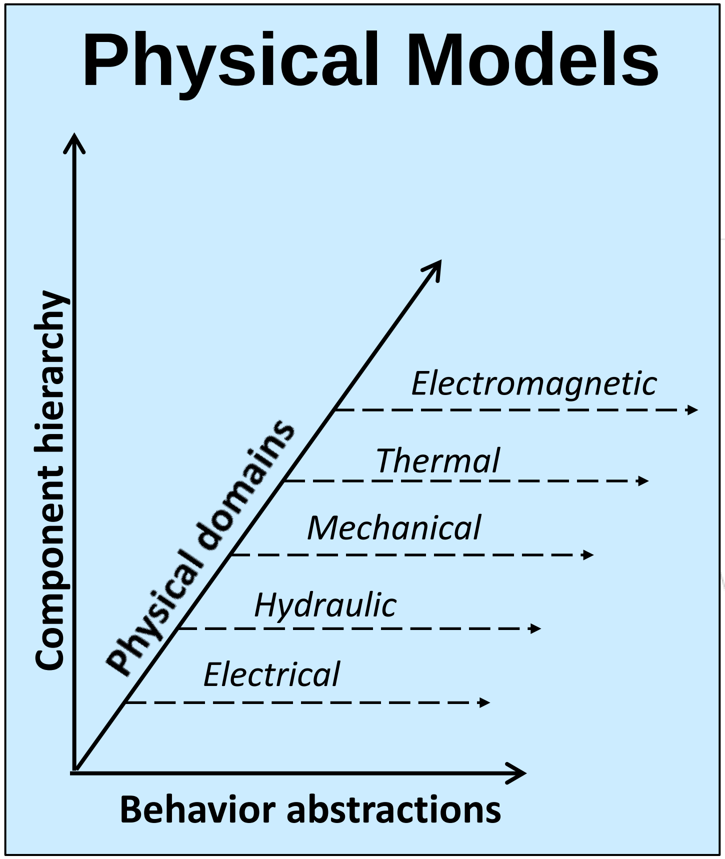
- CPS design flows typically separate physical domains, physical and computational domains, abstraction layers;
- use domain specific composition and verification theories and methods while neglecting cross-domain interactions and interdependences
- pay the price at system integration

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

Modeling Domains



Cross-domain Interactions: Physical



Understanding and representing modeling uncertainties is crucial

- physical (parametric)
- **conceptual (epistemic)**

Component hierarchy

- automated bottom-up composition of behaviors and structural (geometric) properties
- manually constructed lower fidelity physics-based models for assemblies
- automatically constructed non-physics based surrogate models

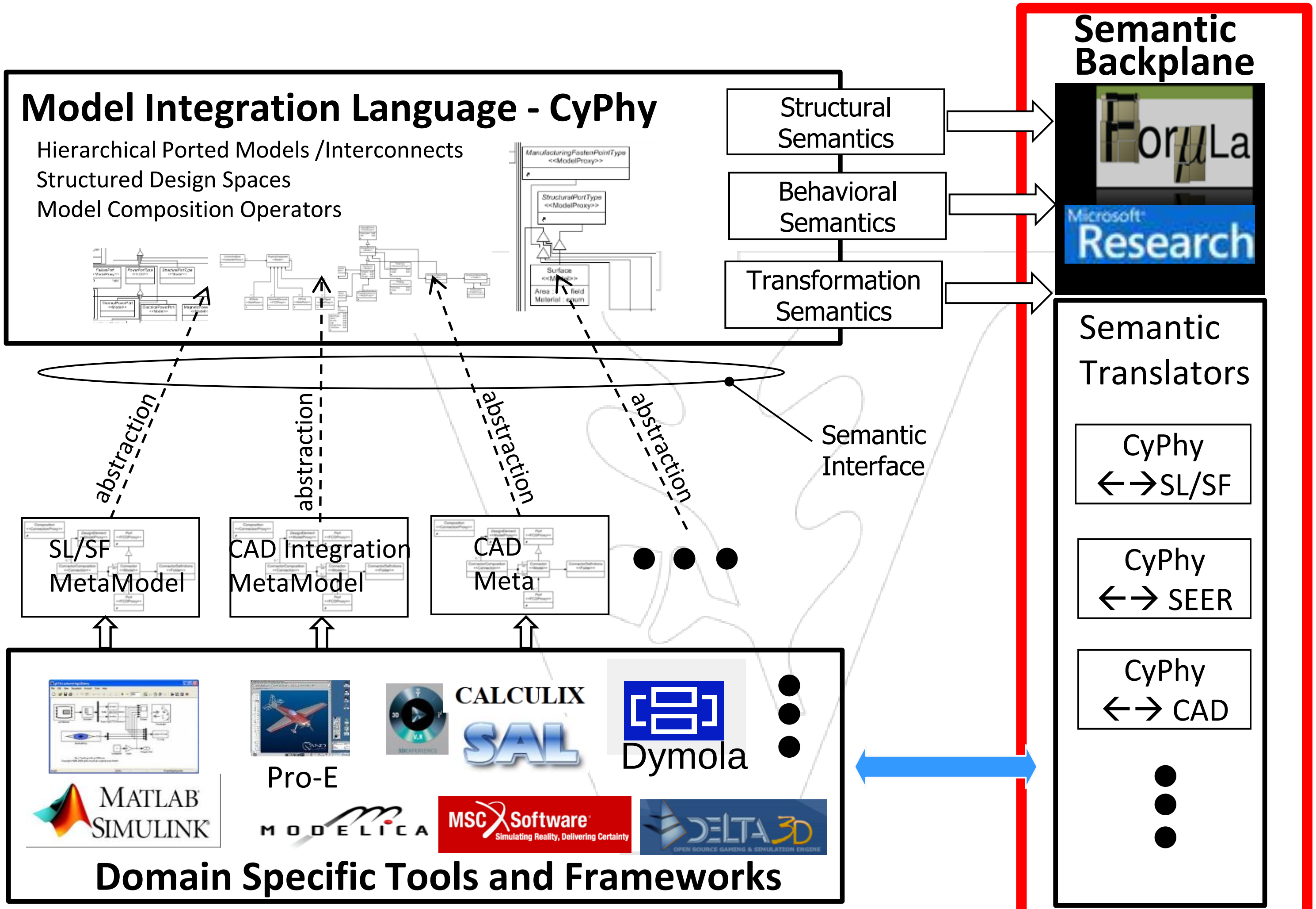
Physical domains

- cross-domain interaction
- enabling/disabling phenomena
- Modelica Standard Library

Behavior abstractions

- static characterization of behavioral properties
- lumped parameter dynamics (nonlinear, hybrid, piecewise lin., linear)
- Finite Element Models

Answer to Challenge 1: Model Integration Languages



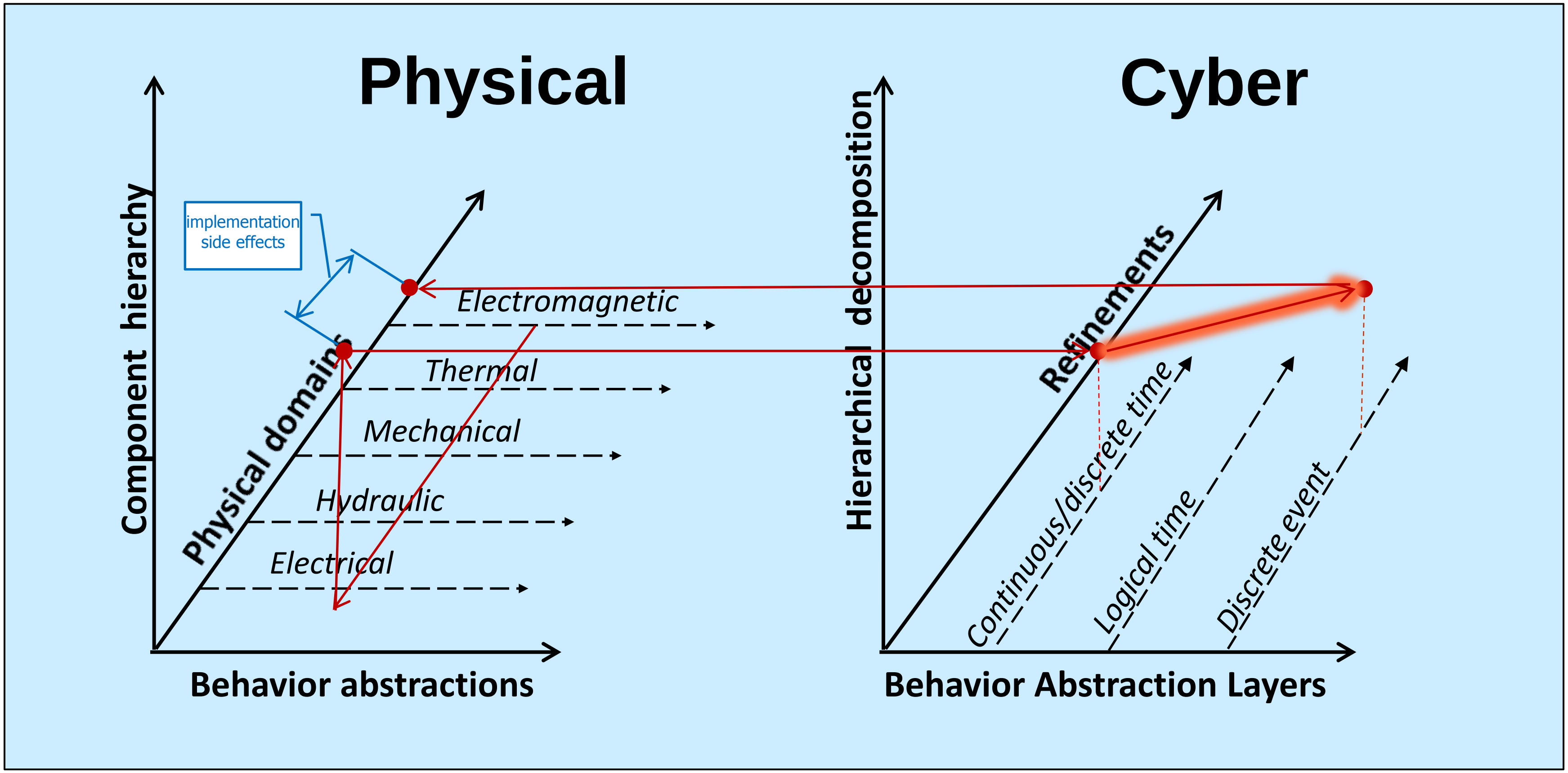
Foundation for MILs: Formal, composable semantics and semantic interfaces

Challenge 2: 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

Cross-Domain Interactions: Cyber-Physical



Answer to Challenge 2: Decoupling

- Dissipativity, Passivity and Symmetry
- Understanding of network effects in "cyber" implementation of continuous dynamics:
  - Structure – graph models
  - Asynchrony and uncertainty
  - Wireless effects

Challenge 3: Validation and Verification

In model-based design verified properties are properties of the models

- Significant scalability problems even in relatively simple (but real) systems
- Scalable verification requires strong restrictions on modeling abstractions (e.g. linear hybrid dynamics, reduced-order systems) and NOT high data fidelity
- Dealing with modeling uncertainty is essential:
  - probabilistic uncertainty
  - epistemic uncertainty

Model validation is crucial for physical components

Contribution to Challenge 3: Cross-domain Tradeoff

- Decoupling drastically decreases complexity of verifying cross-cutting properties, such as stability.
  - One of our major goals to demonstrate the gain in automotive domain

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 (demo)**

Project Plan

Year	Milestone	Success Criteria
Year 1	*End-to-end model-based system integration.* Baseline modeling and system integration tool chains are demonstrated in both testbeds with models in all design layers and automated integration on real platform.	Integrated systems are fully functional. Tool chains are customized to testbeds.
Year 2	*Demonstration of impact of decoupling on system integration in the automotive and networked control testbed.* Theory is validated by measuring resilience of stability against changes in controller implementation (scheduling, clock rate, load changes, comm. delays). Demonstration of integration of deadlock/invariance analysis in tool chain.	Experimental proof that stability of the physical platforms are preserved under adverse implementation changes.
Year 3	*a.) Demonstration of maintaining stability, safety and performance requirements* while partially reconfiguring control architectures (features) in testbeds. *b.) Demonstration of tool chain reconfiguration* without losing semantic integrity	Experimental proof of safety guarantees and performance optimization under decoupling. Semantics-based integration of tools.
Year 4	*a.) Demonstration of maintaining stability, safety and performance requirements* while executing platform architecture change in both testbeds. *b.) Demonstration of rapid integration of new tool components* required by the platform change	Experimental proof that platform reconfiguration and change can be completed without changing controller architecture.
Year 5	*Demonstration of incremental and continuous system integration process* for evolving vehicle architectures and the feasibility and practicality of virtual system integration in both testbeds	Experimental proof that the model-generated (simulated) behavior and physical system behavior correlates in stability, safety and performance metrics