Model-Based Design for Resilient Cyber-Physical Systems

Xenofon Koutsoukos, Janos Sztipanovits, and Gabor Karsai

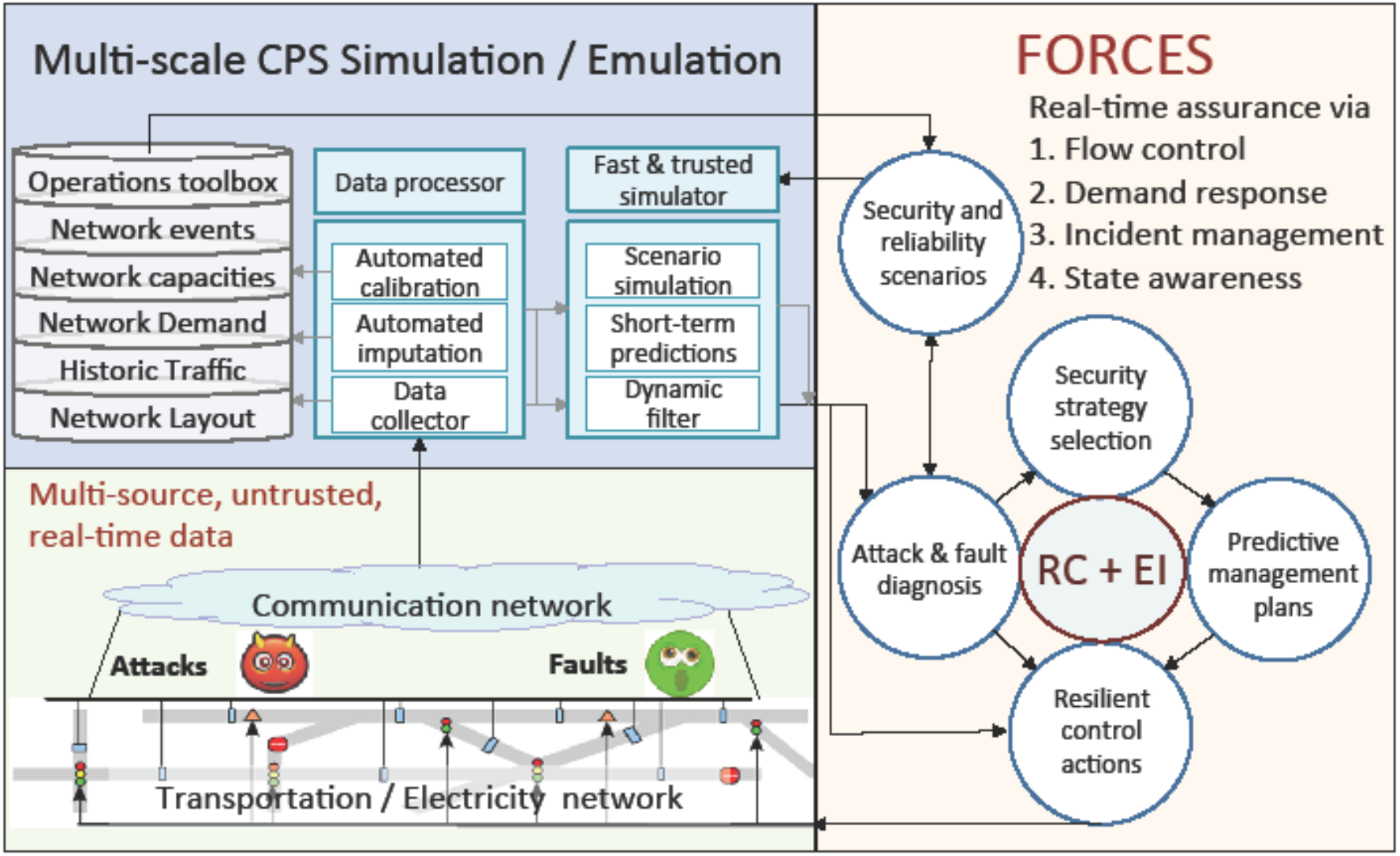
Institute for Software Integrated Systems
Vanderbilt University

FORCES Kickoff Meeting
Washington, D.C., April 12, 2013

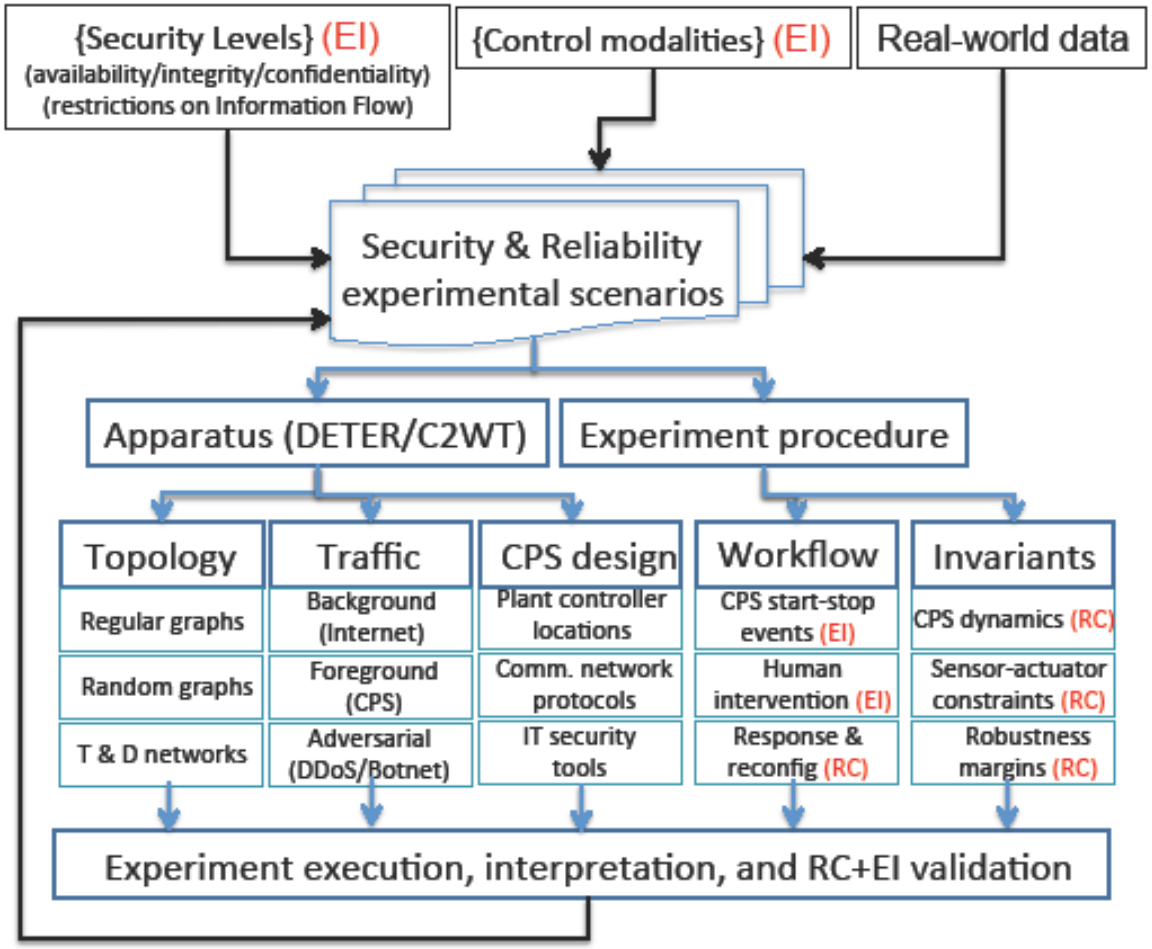
Overview

- **Objective:** Tools for multi-layer RC+EI integrated model-based design
  - Validation using multi-scale CPS simulation/emulation
  - Co-design and co-experimentation
- **Example:** Resilient cooperative control of networked systems
  - EI-aware RC design
- **Research Plan**
  - Multi-scale CPS simulation: Command and control wind tunnel (C2WT)
  - Multi-scale CPS emulation: DETERLab
  - Reference implementation using an open software infrastructure
  - FORCES Modeling Integrated Language (MIL)

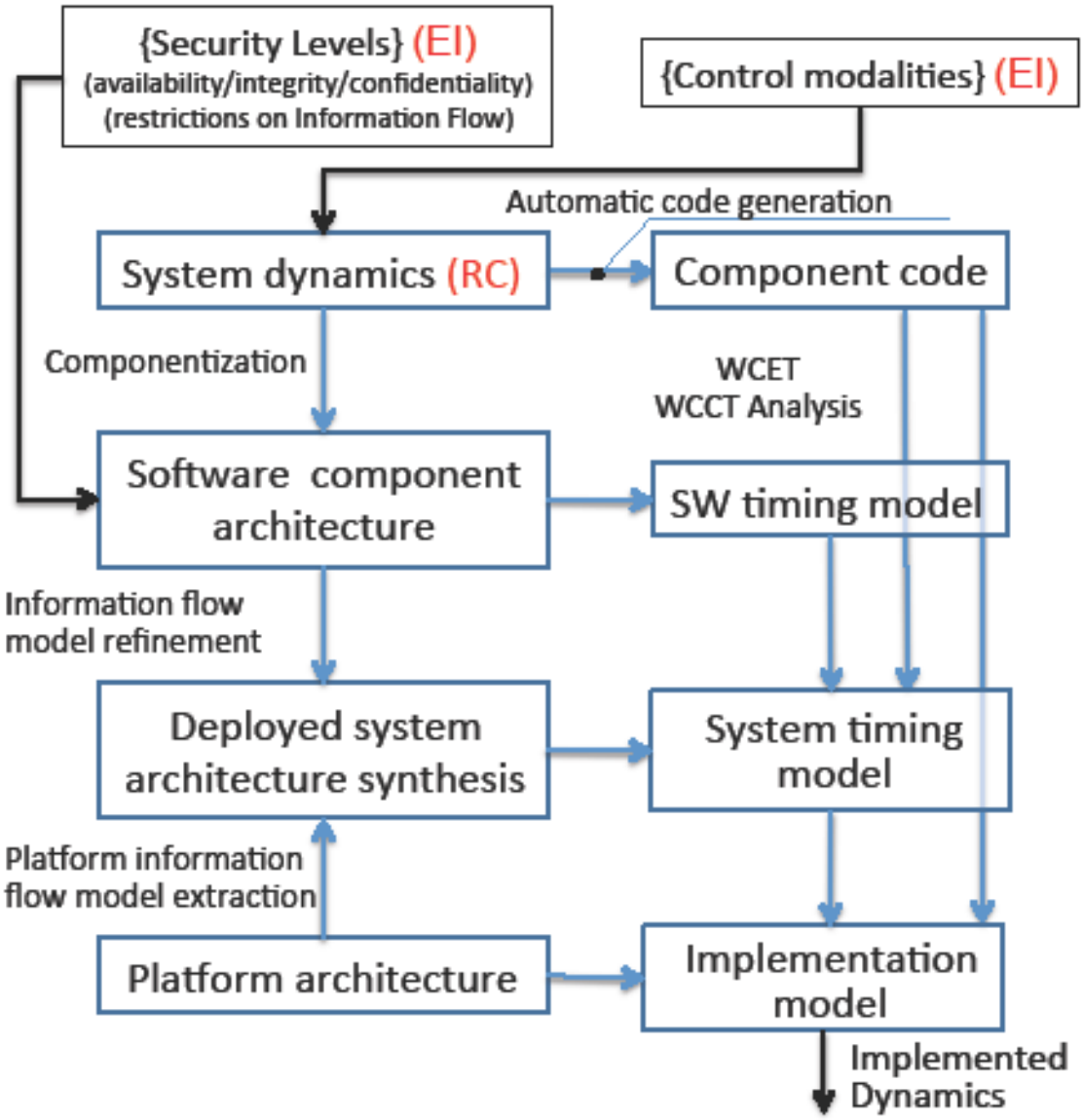
Validation Using Multi-Scale CPS Simulation/Emulation



Co-Experimentation

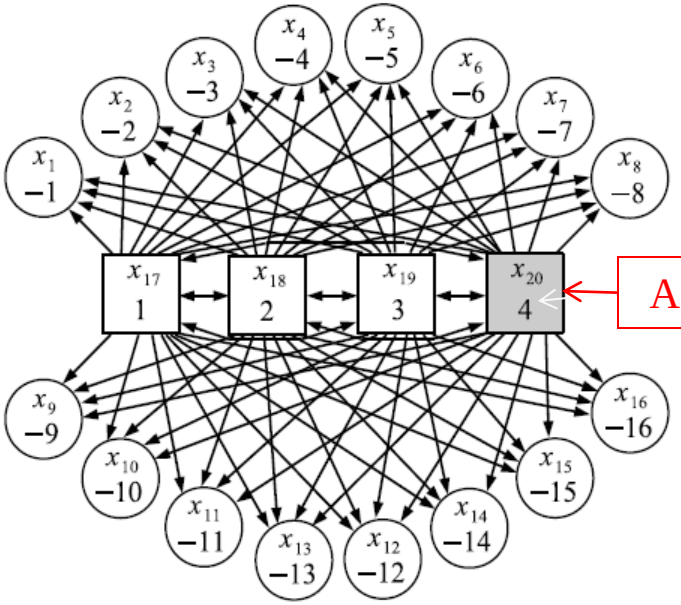


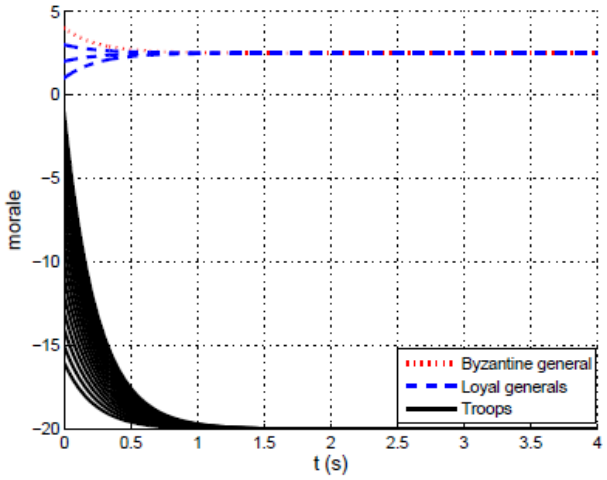
Co-Design


{Security Levels} (EI)
(availability/integrity/confidentiality)
(restrictions on Information Flow)
{Control modalities} (EI)
Automatic code generation
System dynamics (RC)
Component code
Componentization
WCET
WCCT Analysis
Software component architecture
SW timing model
Information flow model refinement
Deployed system architecture synthesis
System timing model
Platform information flow model extraction
Platform architecture
Implementation model
Implemented Dynamics

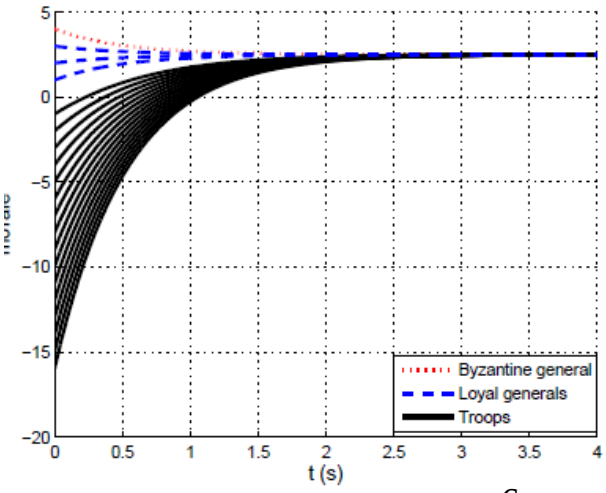
Example: Resilient Cooperative Control in the Presence of Adversaries

- Resilient Asymptotic Consensus
  - Continuous Time [HSCC2011,HSCC2012]
  - Discrete Time
  - [HiCoNS 2012, JSAC 2013]
- Resilient Asymptotic Synchronization
  - LTI Systems
  - [HSCC 2013]





(a) LCP.

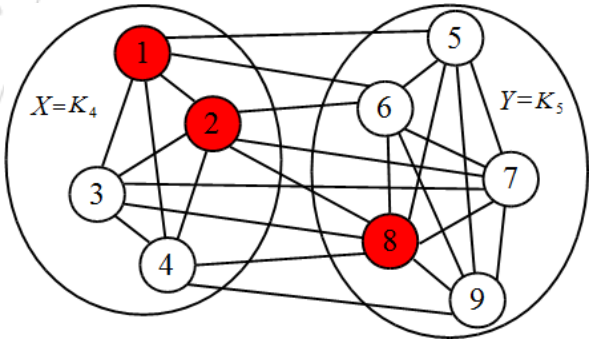


(b) ARC-P.

Adversary Models

- Crash Adversary
- Malicious Adversary
  - Must convey the same information to all neighbors
    - Local broadcast model
- Byzantine Adversary
  - Can convey different information to different neighbors
- All adversaries are omniscient
  - Topology of the network
  - States and algorithms of the other nodes
  - Other adversaries (can collude)

- F-Total Model
  - At most F adversaries in the entire network
- F-Local Model
  - At most F adversaries in the neighborhood of any normal node
- f-Fraction Local Model
  - At most a fraction f of adversaries in the neighborhood of any normal node



3-total, 3=local, (3/5)-fraction local

Resilient Asymptotic Consensus

- Hybrid system dynamics

$$x_i(t+1) = f_{i,\sigma(t)}(t, x_i(t), \{x_{(j,i)}(t)\}),\ i \in \mathcal{N}, j \in \mathcal{N}_i^{\text{in}}, t \in \mathbb{Z}_{\geq 0}, \mathcal{D}_{\sigma(t)} \in \Gamma_n$$

- Agreement Condition

$$\lim_{t\to\infty} \Psi(t) = 0 \quad \text{where } \Psi(t) = M_{\mathcal{N}}(t) - m_{\mathcal{N}}(t)$$

- Safety Condition

$$x_i(t) \in \mathcal{I}_t = [m_{\mathcal{N}}(t), M_{\mathcal{N}}(t)], \quad \forall t \in \mathbb{Z}_{\geq 0}, \forall i \in \mathcal{N}$$

- Weighted Mean-Subsequence-Reduced (W-MSR) Algorithm

$$x_i(t+1) = w_{(i,i)}(t)x_i(t) + \sum_{j \in \mathcal{N}_i^{\text{in}}(t) \setminus \mathcal{R}_i(t)} w_{(j,i)}(t)x_{(j,i)}(t)$$

Robust Networks

Threat	Scope	Necessary	Sufficient
Crash & Malicious	F-Total	$(F+1,F+1)$-robust	$(F+1,F+1)$-robust
Crash & Malicious	F-Local	$(F+1,F+1)$-robust	$(2F+1)$-robust
Crash & Malicious	f-Fraction local	f-fraction robust	p-fraction robust, where $2f < p \leq 1$
Byzantine	F-Total & F-Local	Normal Network is $(F+1)$-robust	Normal Network is $(F+1)$-robust
Byzantine	f-Fraction local	Normal Network is f-robust	Normal Network is p-robust where $p > f$

- [IEEE JSAC April 2013]
- Normal network is the network induced by the normal nodes
- Necessary Conditions for F-Total and F-Local are necessary for any successful DTRAC algorithm

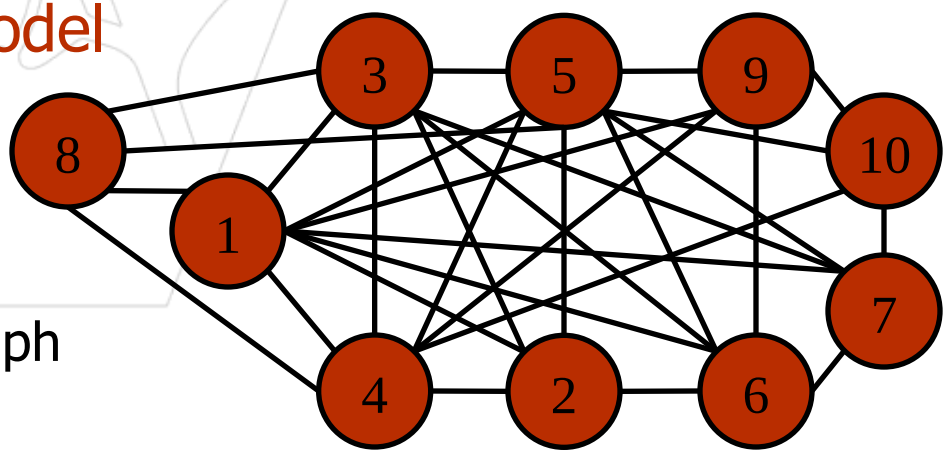
Construction of Robust Networks

- Let $D=(V, E)$ be a nontrivial (r,s)-robust digraph . Then, $D'=(V \cup \{v_{\text{new}}\}, E \cup E_{new})$, where v_{new} is a new node added to D and E_{new} is the directed edge set related to v_{new}, is (r,s)-robust if

$$d^{\text{in}}_{v_{\text{new}}} \geq r + s - 1$$

Preferential-attachment model

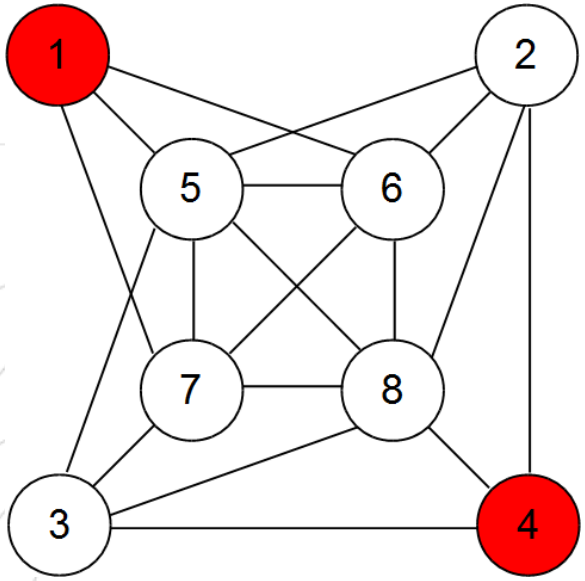
- Initial graph: K_5
- K_5 is (3,2)-robust
- Num edges / round: 4
- End with (3,2)-robust graph



EI-Aware RC Design

- Attacker strategy
  - F-local model
  - Compromise a node with probability p_i
- Defender strategy
  - Select neighbors to cooperate
- Objective
  - Determine equilibria within the class of mixed strategies
- Resilient control
  - Implement the outcome using W-MSR

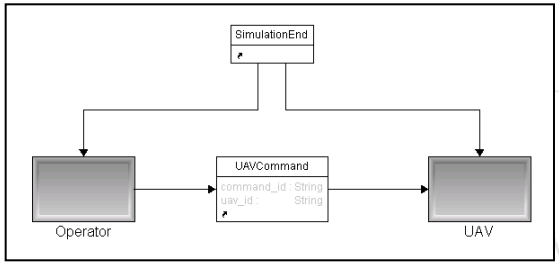
$$x_i(t+1) = w_{(i,i)}(t)x_i(t) + \sum_{j \in \mathcal{N}_i^{\text{in}}(t) \setminus \mathcal{R}_i(t)} w_{(j,i)}(t)x_{(j,i)}(t)$$



Research Challenge:
Can we characterize robust network topologies?

Command and Control Wind-Tunnel (C2WT)

C2W **integration** models
(data flow, timing, parameters)



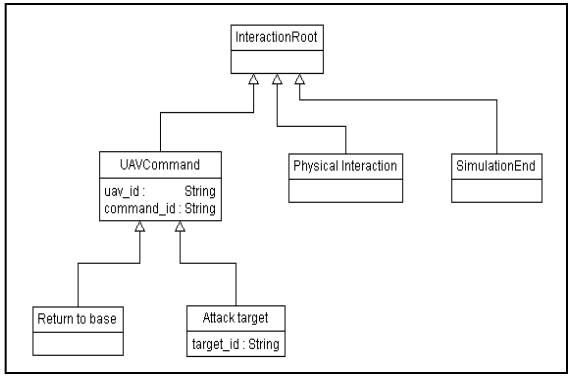
configuration

Based on C2WT models configuration files are generated for the various simulation components.
Configure how the component is connected to the simulation (input-output binding)

Domain specific
C2W simulation components

OMNET component	CPN component
Simulink component	Delta3D component

C2W **data** models
(interaction and object models)



transformation

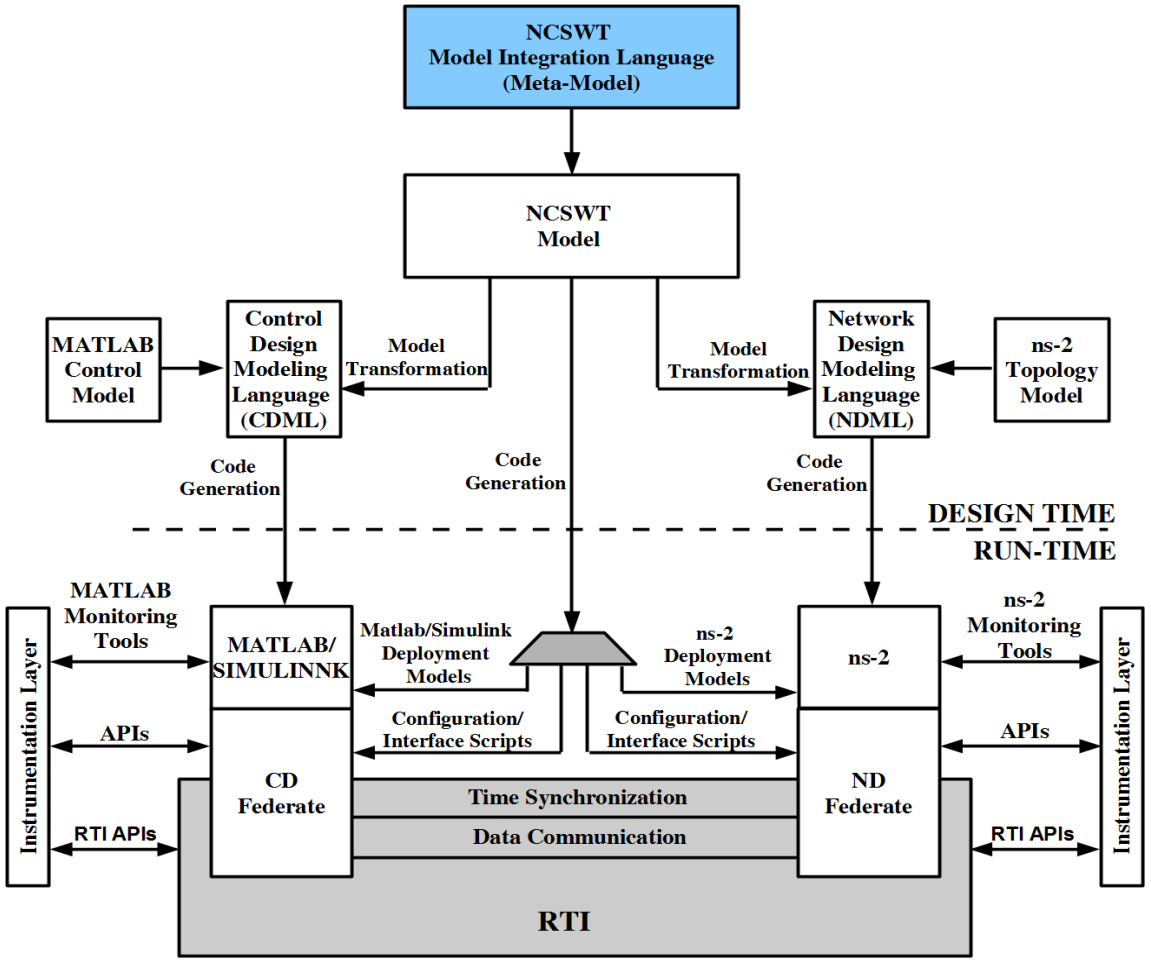
Federates have to have a common data model to be able to share data.
• Data model can be imported from domain specific models
• Domain specific models can be generated from data models

Domain specific
simulation models

Omnet models	CPN models
Simulink models	…

Network Control System Simulation

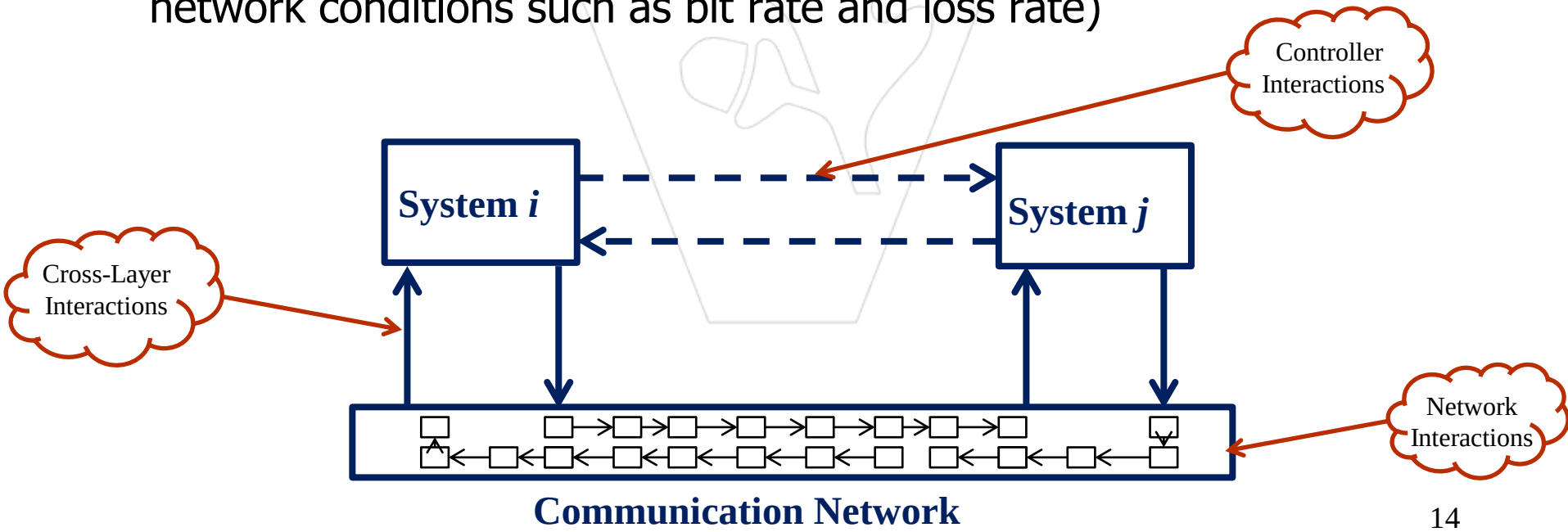
Model-based Design and Integration (Design-Time Modeling Environment)



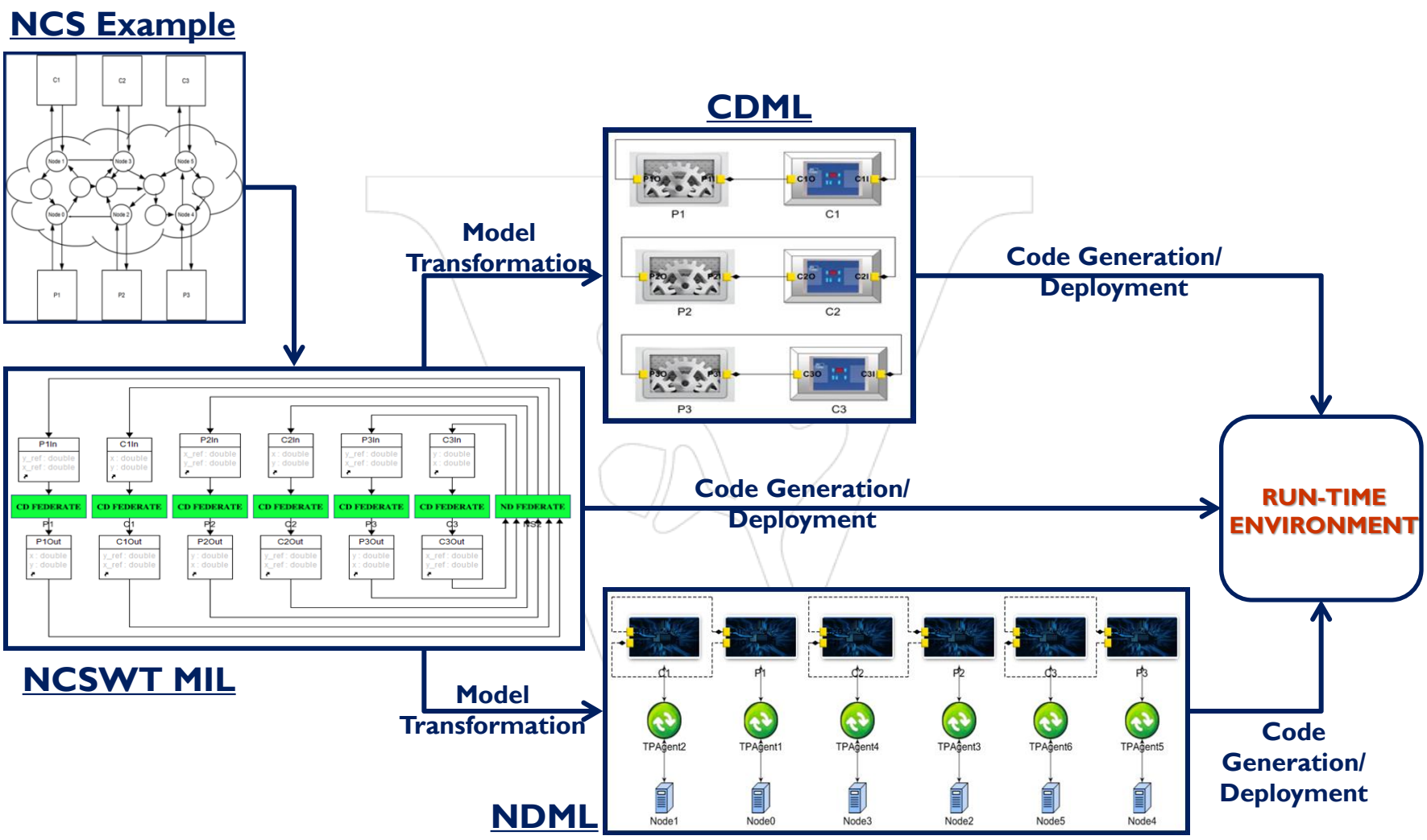
HLA-based Run-Time Simulation Environment

Network Control System MIL

- Network interactions: Information exchange that flows through the communication network (e.g., sensor and control signals)
- Controller interactions: Information exchange by means other than the communication network (e.g., a proximity sensor on a UAV for obstacle detection)
- Cross-Layer interactions: Information exchange between the network and application layers of a network protocol stack (e.g., network conditions such as bit rate and loss rate)



Model-Based Design and Integration

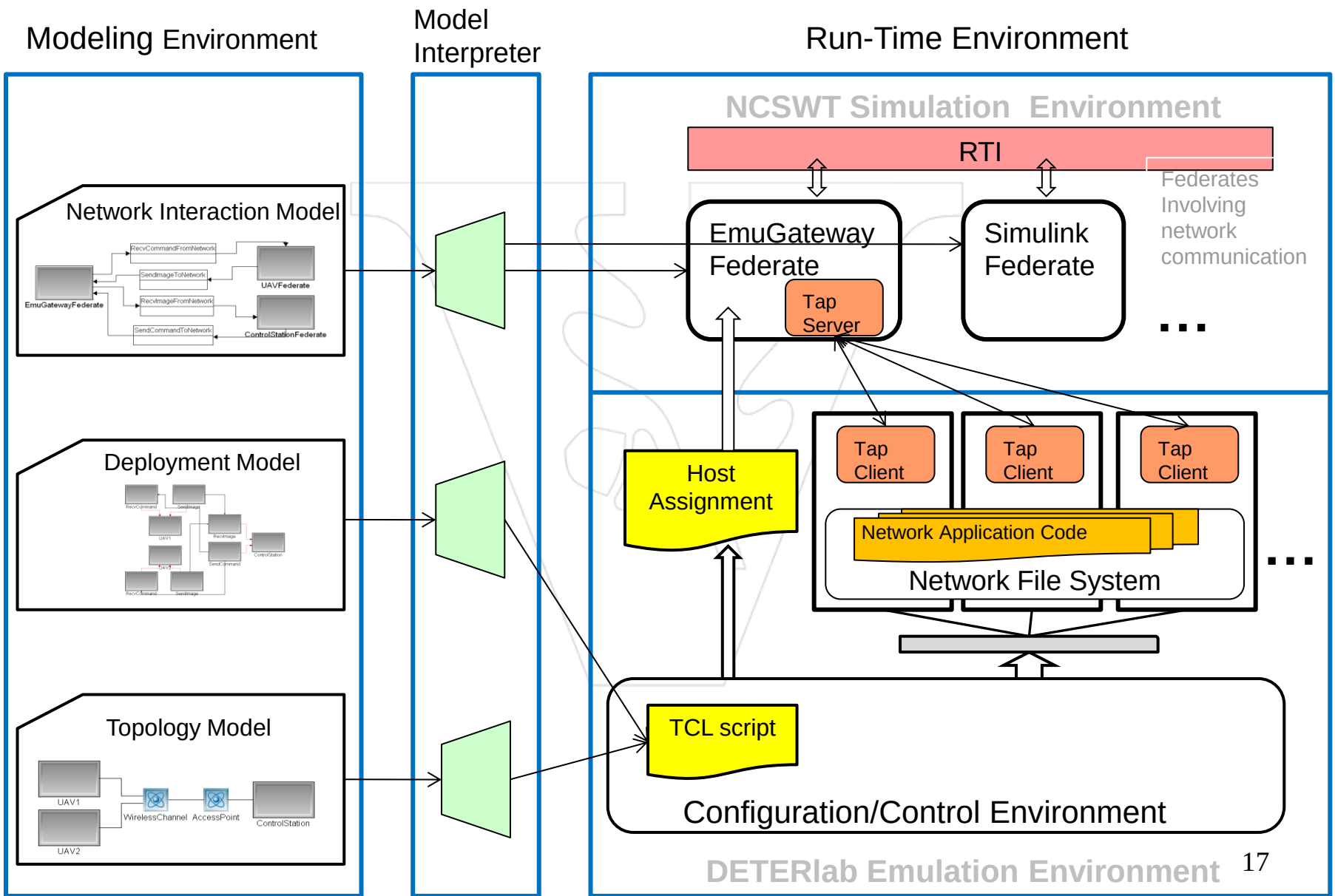


Multi-Scale CPS Emulation

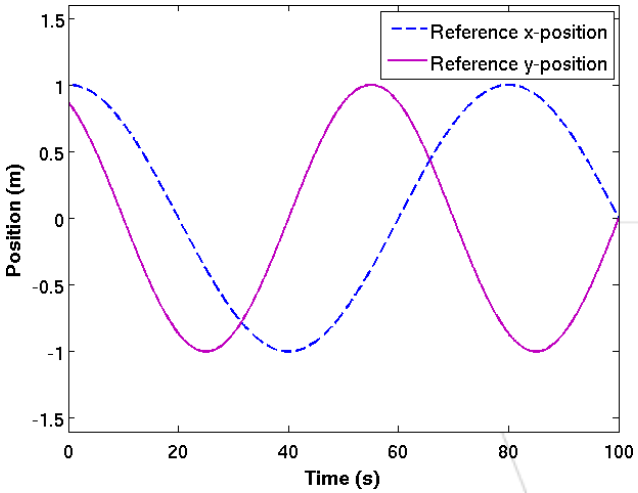
- Integrated Simulation and Emulation Platform for Cyber-Physical System Security Experimentation (iSEE)
  - Greater realism and accuracy with truthful protocol implementation and real network traffic delivery
  - Providing a computing platform where prototypes of software components can be deployed
- Modeling environment for system specification and experiment configuration
  - System model of CPS
  - Security experiment scenario configuration
- Run-time environment that supports experiment execution
  - DETERlab: large number of tools available for emulate network attacks

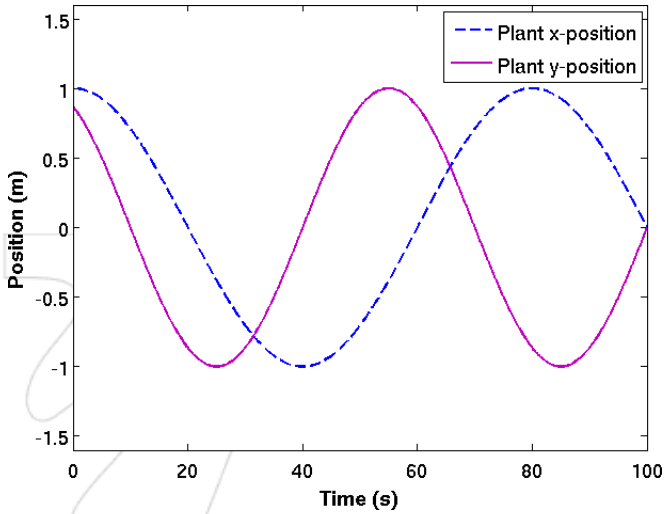
[HiCoNS 2012]

iSEE Framework

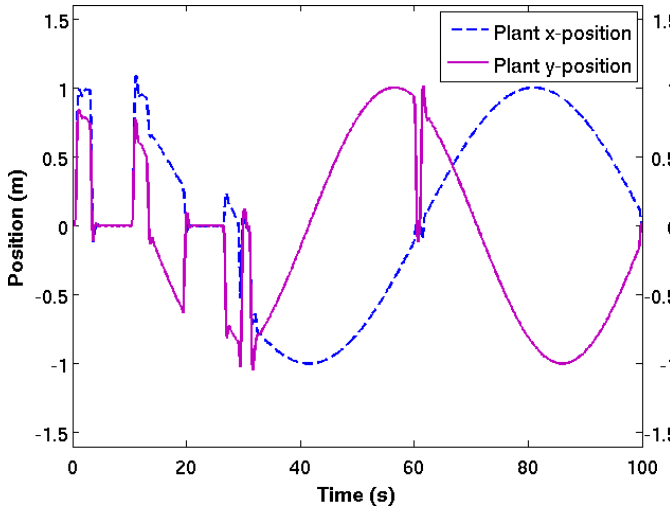


Impact of Security Attacks

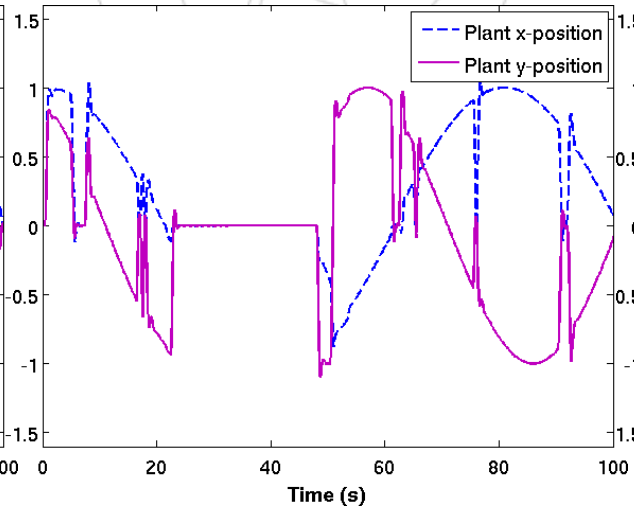




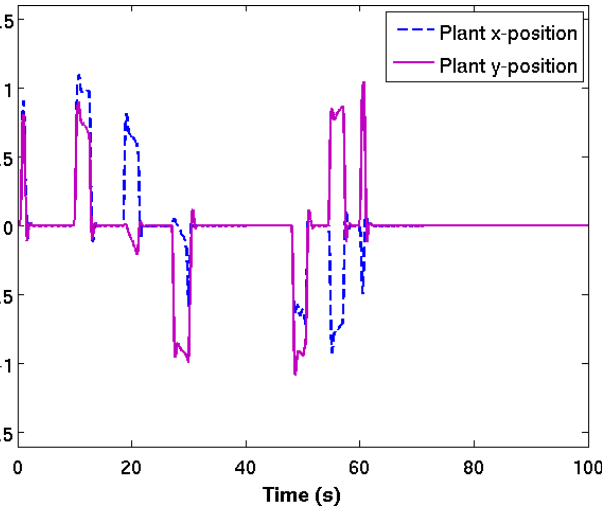
10% loss rate



20% loss rate

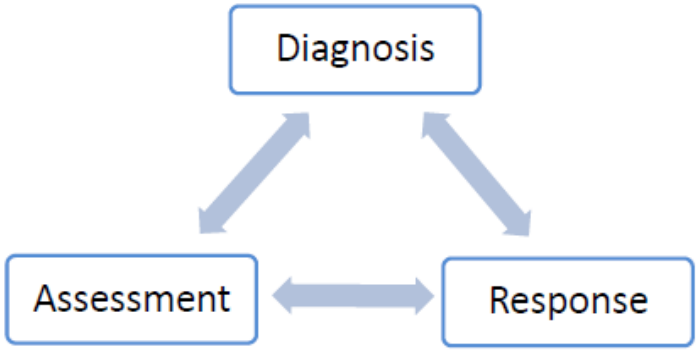


30% loss rate

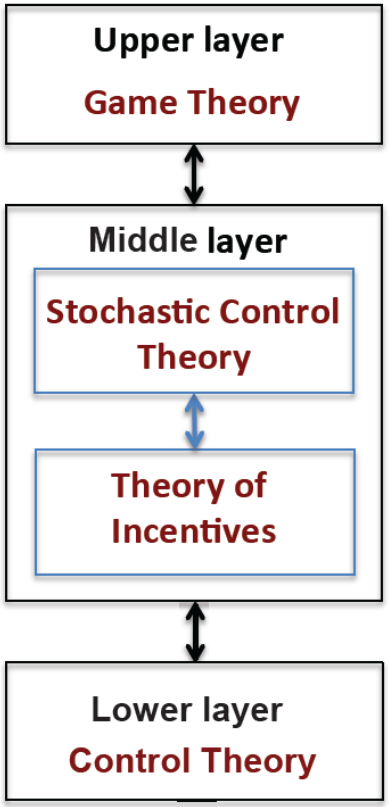


Research Challenges

- Integration of resilient control design and validation
  - Threat assessment
  - Attack diagnosis
  - Attack resilient control

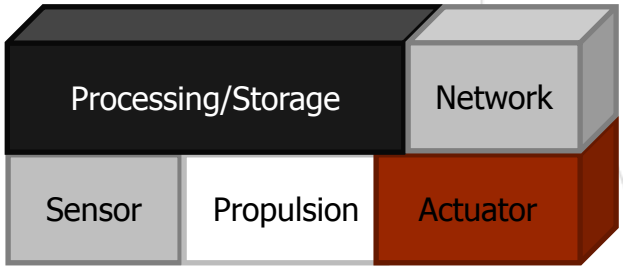


- Integration of hierarchical EI+RC design and validation

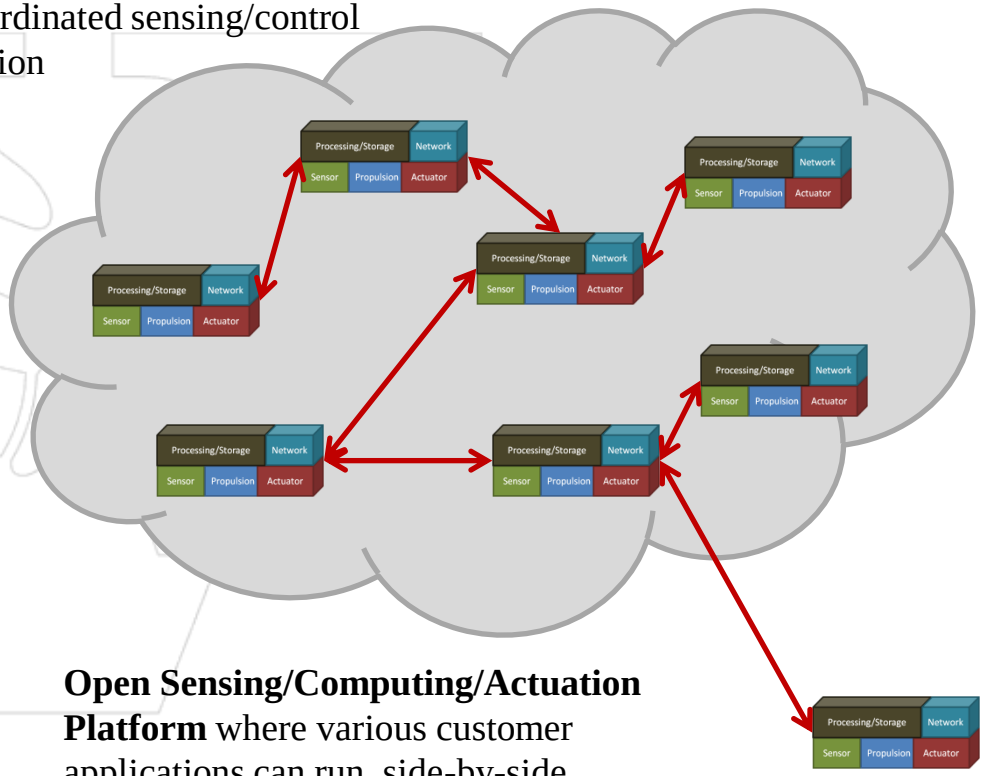


A Distributed Sensor/Control Network Platform

Networked node with local processing and storage, sensors, actuators, and propulsion system:



Nodes for an ad-hoc network that has 1+ ground-link and performs a coordinated sensing/control function



<u>Examples</u>:

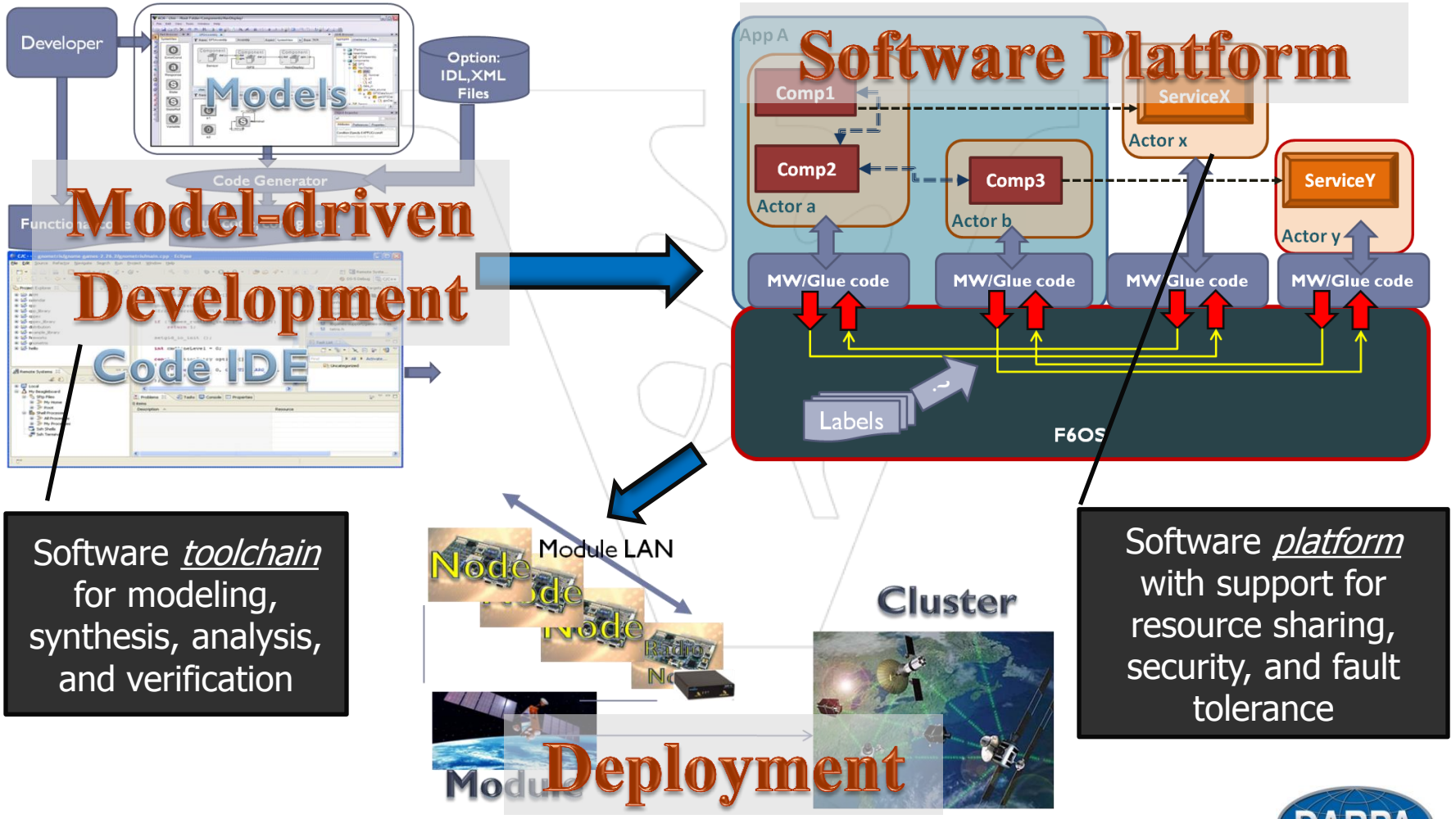
- Swarm of UAVs performing tornado damage surveillance
- Fleet of UUVs performing collecting climate change data from oceans
- *DARPA System F6:* Fractionated Spacecraft – A Space Global Common

Open Sensing/Computing/Actuation Platform where various customer applications can run, side-by-side.

<u>Challenges</u>:

- Networked, distributed control
- Fault-resilience
- Applications with different trust and security levels must run side-by-side

An Open Software Infrastructure



Software *toolchain* for modeling, synthesis, analysis, and verification

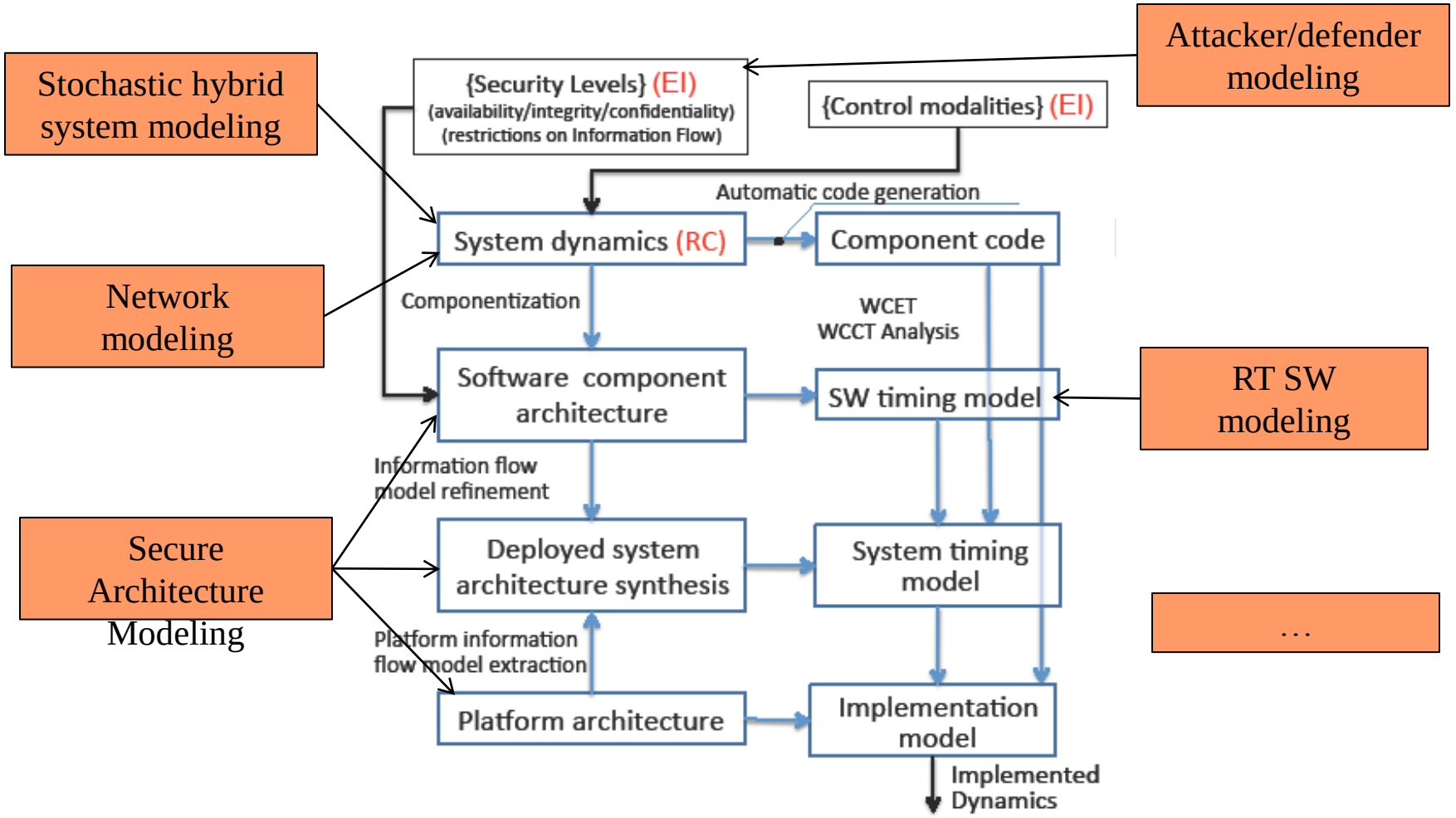
Software *platform* with support for resource sharing, security, and fault tolerance

The F6 IAP Reference Implementation will be released under the MIT/X Open Source License.

Research Challenges

Area	Background	Challenge
System/Security Co-Design	The model-based development framework supports Multi-Level Security on software components	How to verify the security of information flows for a suite of component-based applications?
	The software platform APIs provide robust and protected interfaces via generated code	How to verify that the APIs are robust and resilient to security attacks?
Robust Networked Control	The software platform has the open interfaces to implement various distributed control schemes for vehicle flight control	Given that malicious applications and network interference may be deployed, how to ensure that critical control functions are available even under attacks?
Threat assessment and diagnostics	The software platform has the open interfaces to add security/threat monitors	How to implement a threat monitor for a CPS Cloud?
Mechanism design	The physical platform is operated by an owner who makes it available to (not necessarily completely trusted) customers and their software applications	How to balance economic incentives and resilience techniques to achieve an economical operation of the cloud platform even under fault scenarios and security attacks?

Co-Design Requires Modeling Languages



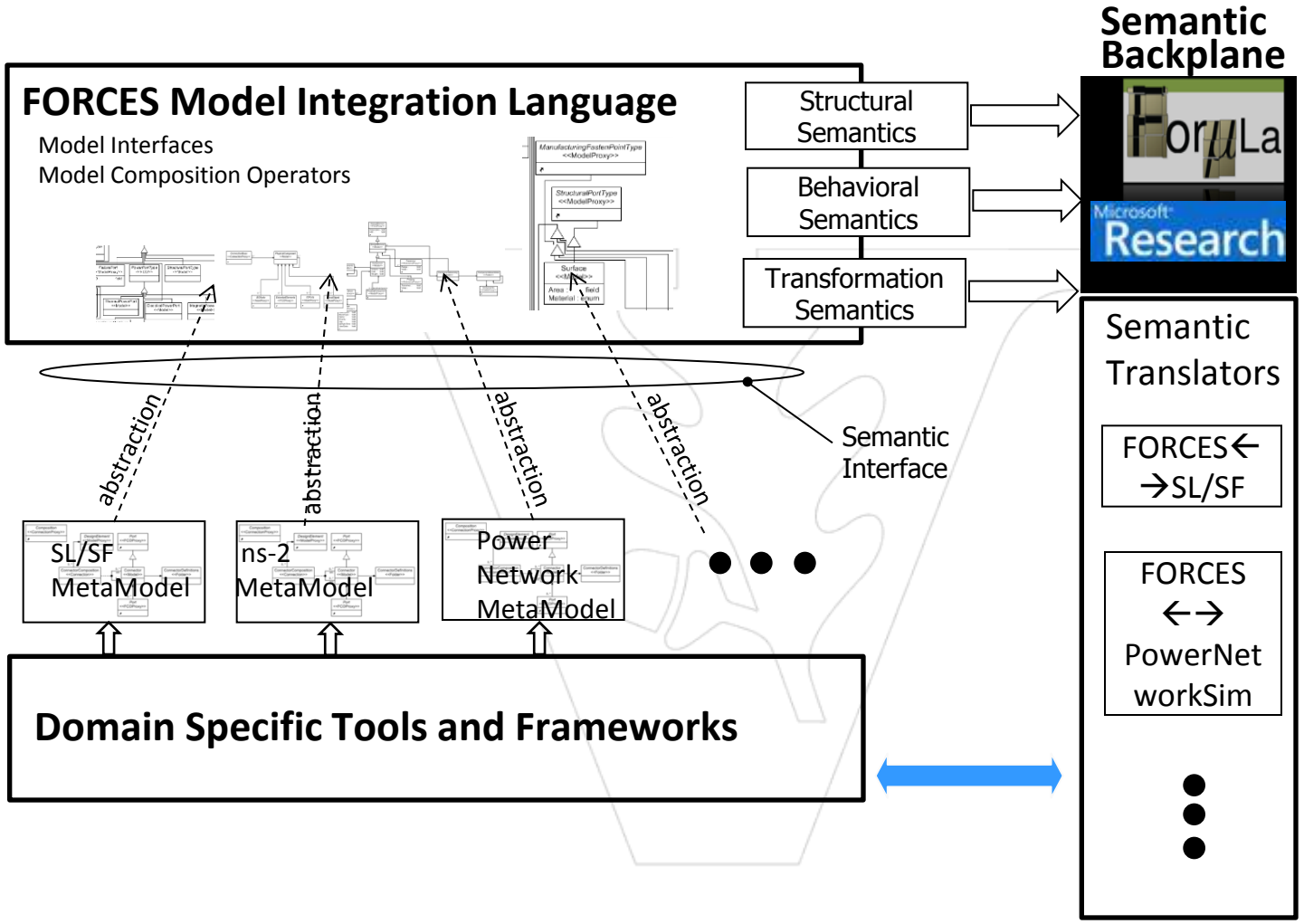
Domain Specific Modeling Languages

How Should We Choose Modeling Language(s)?

- Define yet another modeling language?
- Choose one that already exists and broad enough to cover the design domain?
- Create a new standard or update an old one?

- What are the implications on tools?
- How about "my freedom of abstractions"?
- What is the language agility and evolution trajectory?

Research Challenge: FORCES MIL



Impact: Open Language Engineering Environment → Adaptability of Process/Design Flow → Accommodate New Tools/Frameworks , Accommodate New Languages

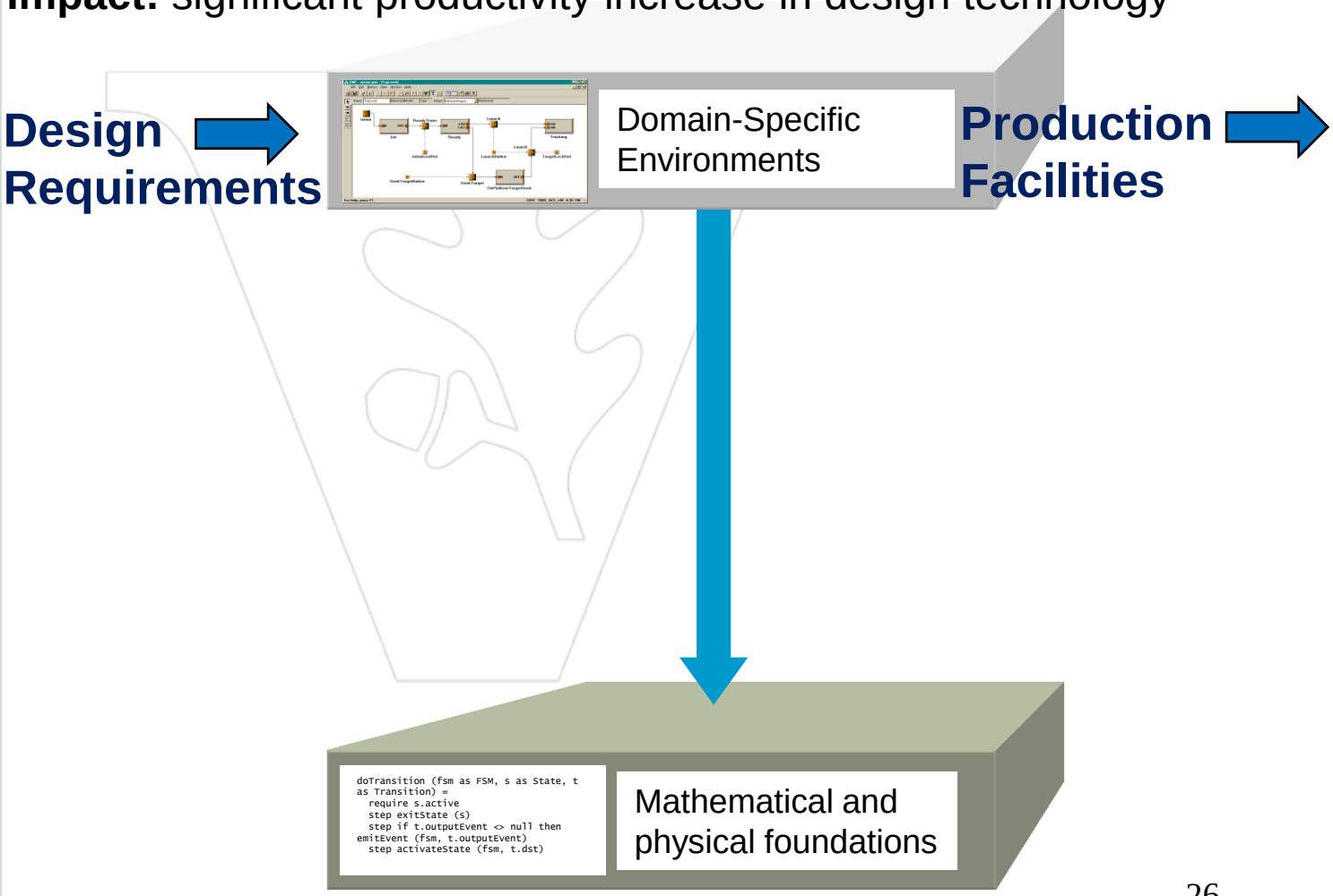
Model-Based Design

Key Idea: Use models in domain-specific design flows and ensure that final design models are rich enough to enable production of artifacts with sufficiently predictable properties.
Impact: significant productivity increase in design technology

Domain Specific Design Automation Environments:
- *Power*
- *Transportation*

Tools:
- *Modeling*
- *Analysis*
- *Verification*
- *Synthesis*

Challenges:
- *Cost of tools*
- *Benefit only narrow domains*
- *Islands of Automation*



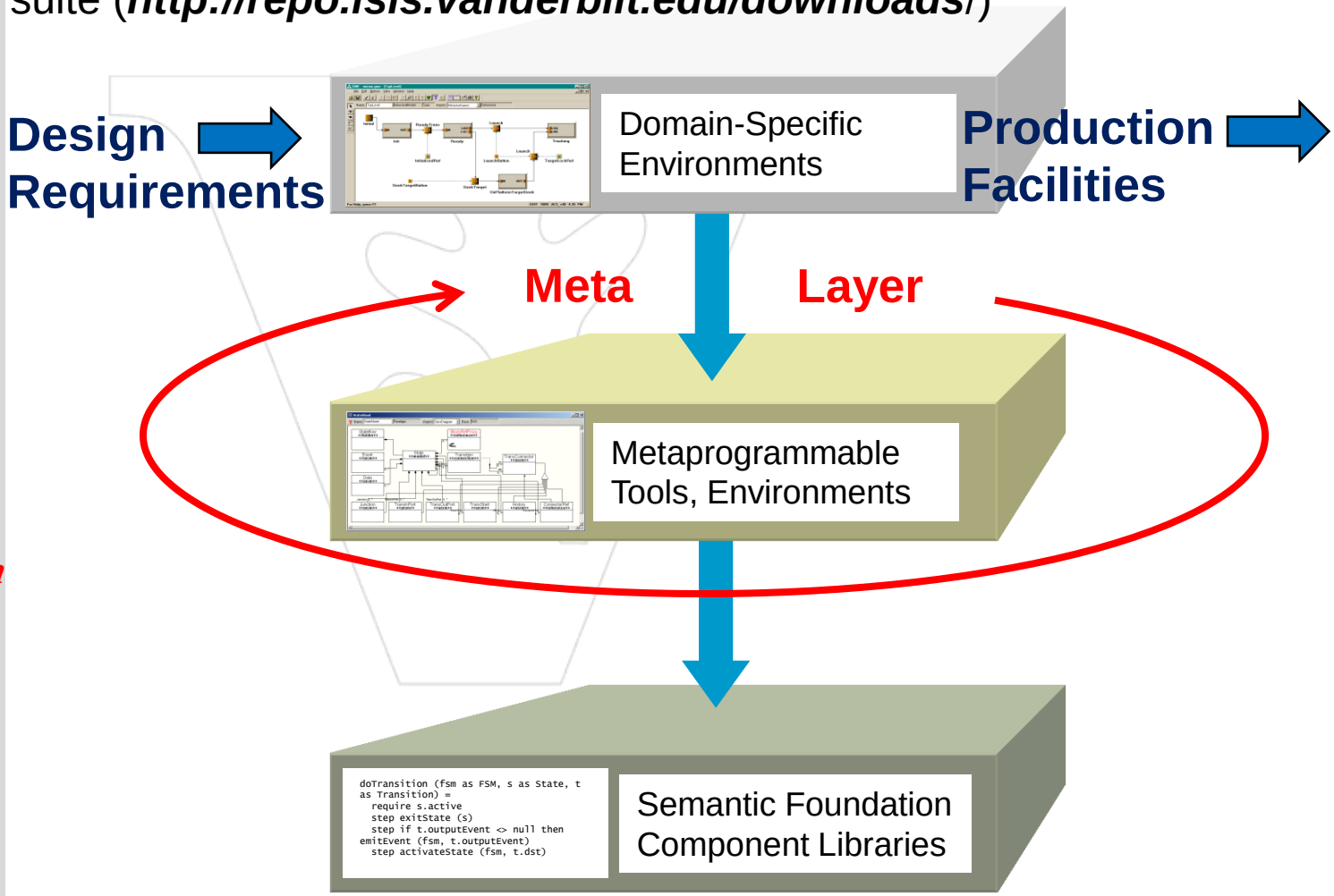
Metaprogrammable Design Tools

Domain Specific Design Automation Environments:
- *Power*
- *Transportation*

Metaprogrammable Tool Infrastructure
- *Model Building*
- *Model Transf.*
- *Model Mgmt.*
- *Tool Integration*

Explicit Semantic Foundation
- *Structural*
- *Behavioral*

Key Idea: Ensure reuse of high-value tools in domain-specific design flows by introducing a metaprogrammable tool infrastructure.
VU-ISIS implementation: Model Integrated Computing (MIC) tool suite (***http://repo.isis.vanderbilt.edu/downloads/***)



Summary

- Tools for model-based integrated EI+RC design
- Multi-scale CPS simulation and emulation
- FORCES modeling integration language