SYSTEMS ENGINEERING RESEARCH CENTER

Georgia Tech Research Institute

RT-204: Systemic Security and the Role of Hierarchical Design in Cyber-Physical Systems

Sponsor: DASD(SE)

By

Dr. Valerie B. Sitterle

Mr. Tom McDermott

10th Annual SERC Sponsor Research Review

November 8, 2018

FHI 360 CONFERENCE CENTER

1825 Connecticut Avenue NW, 8th Floor

Washington, DC 20009

10 YEARS of SERC

www.sercuarc.org

Overview

Goal = Cyber resiliency

- The ability of a defense system to ***maintain mission-effective capability*** under adversary offensive cyber operations

How?

- Design an effective control structure that eliminates or reduces adverse events (STAMP goal)

What are we trying to do?

- Identify and develop methods to discover and evaluate security vulnerabilities for cyber-physical systems (CPS) that are amenable to scalable, intuitive, and re-usable computational implementation

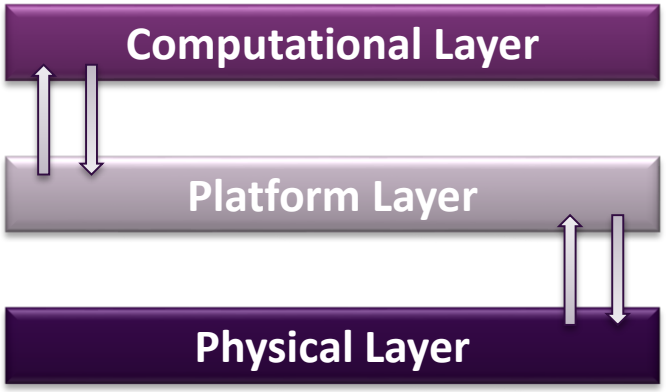
What is different?

- Approach based on ***functional*** abstraction
- ***Ecosystem*** view = system, threat, new security patterns

Cyber-Physical Systems

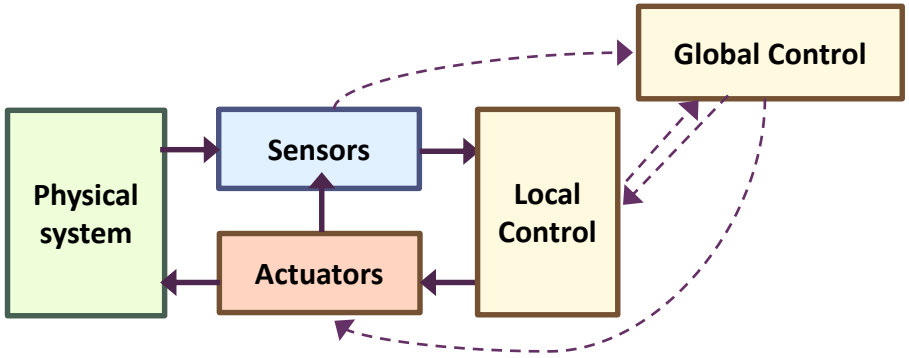
What are cyber-physical systems? 2 views - "Engineered Systems that …"

Are comprised of heterogeneous sensing, computational, and actuating components to collect, process, and physically act on information



SW models, platform models, physical models

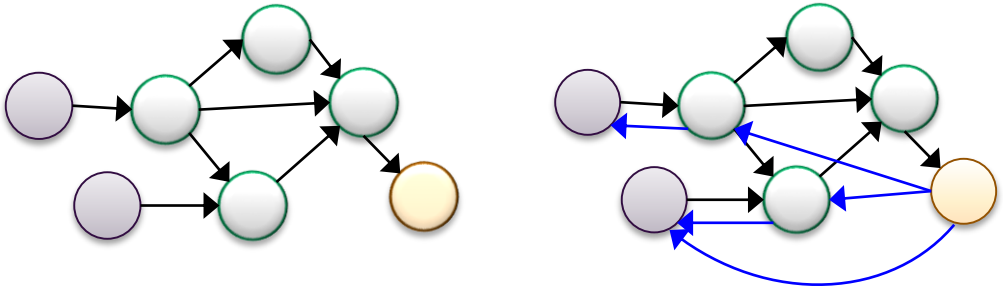
Integrate physical and cyber components where ***relevant functions are realized through interactions*** between the physical and cyber parts



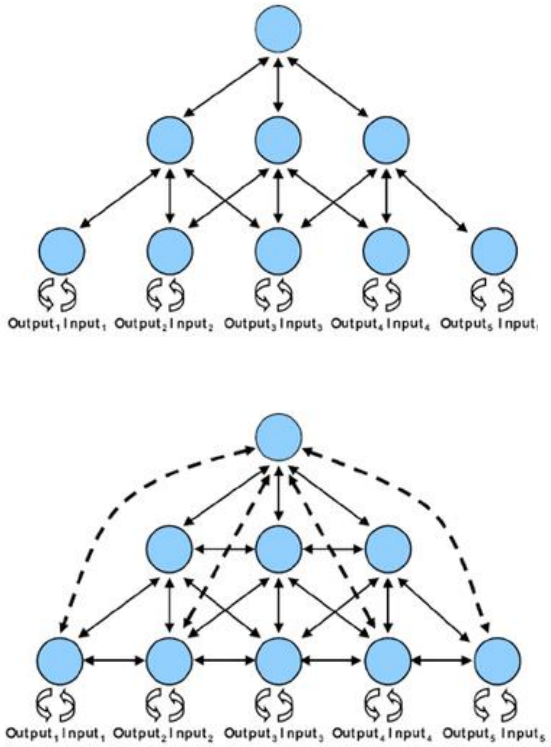
Integration is key to system behavioral abstraction

What Makes CPS Challenging for SE Analysis?

- **Acyclic vs cyclic**
  - Strictly linear (topological) ordering of execution tasks – only occurs in acyclic structures



- **Structure-Function**
  - Hierarchical vs heterarchical vs random
  - Organizational and ecological/ biosystems views differ
  - In CPS, many essential system properties such as stability, safety, performance are expressed in terms of physical ***behavior***
  - ***"Bolt-on"*** technologies change structure = change function



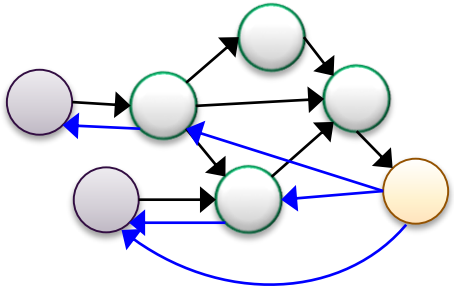
Berntson, Norman, & Cacioppo, 2011.

RT 204 Research Approach

Presume we have a system model [MBSE]

How do we effectively query that model… to produce relevant system representations (i.e., construct a model transform)?

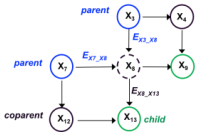
How do we discover what constitutes functions and flows relevant to our analysis?



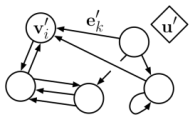
Once we obtain a reduced graph projection of our model, what are efficient approaches to…

- Augment with threat vector functional patterns?

- Attribute micropatterns for functional state?

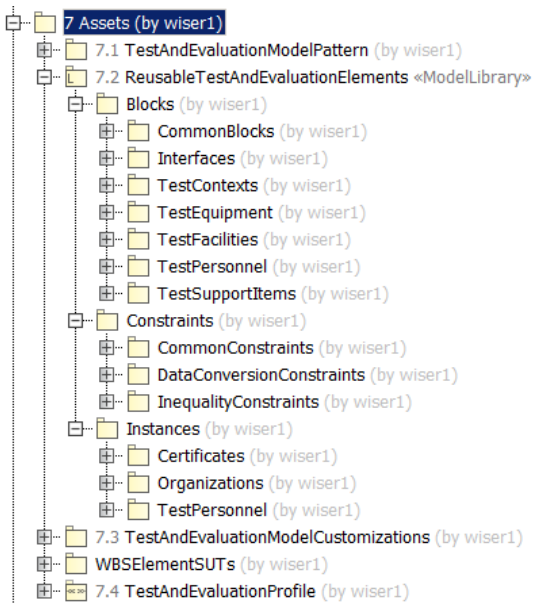


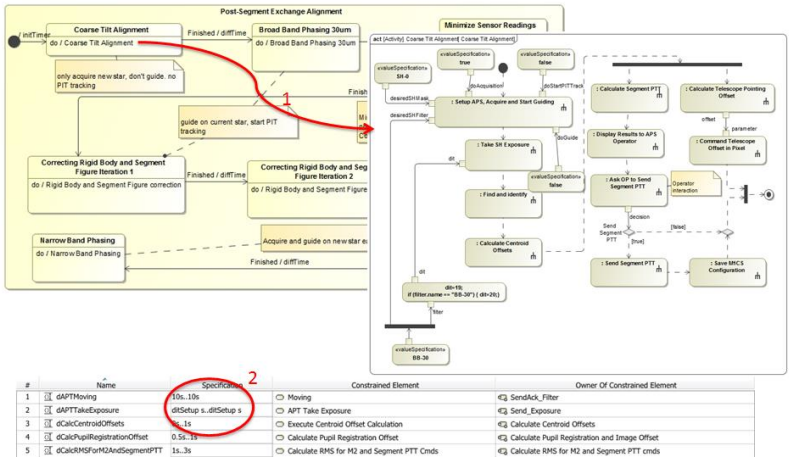
- Determine if functional state space is preserved?



What did we Learn ?

Reaching a Query-able Representation





- SysML system models
  - Common definitions
  - One model with different projections
  - But not easily query-able
- OpenMBEE
  - Normalizes representation
  - Underlying semantic model in (flat) JSON
- JSON LD
  - Process OpenMBEE JSON system model description to produce structured, linked data representation in form open to queries
  - ”Context" to provide additional mappings from JSON to an RDF model

Discovery via Query for Reduced Graph Representation

- **Resource Description Framework (RDF)**
  - Uses a graph structure as the underlying data model
  - Defined in form of triples
  - Edge always directs toward Object

- **SPARQL**
  - RDF query language to retrieve and manipulate relationship-structured data
  - Provides a full set of analytic query operations whose schema is intrinsically part of data

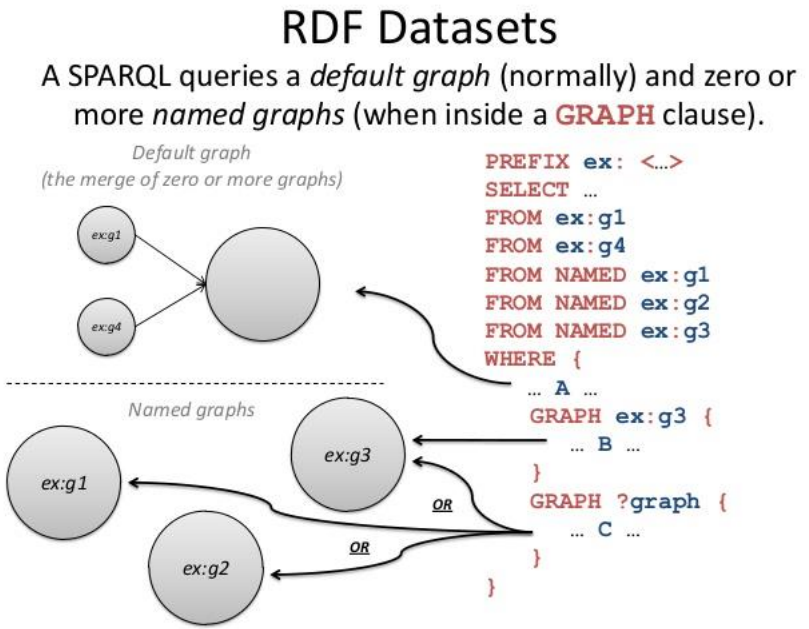
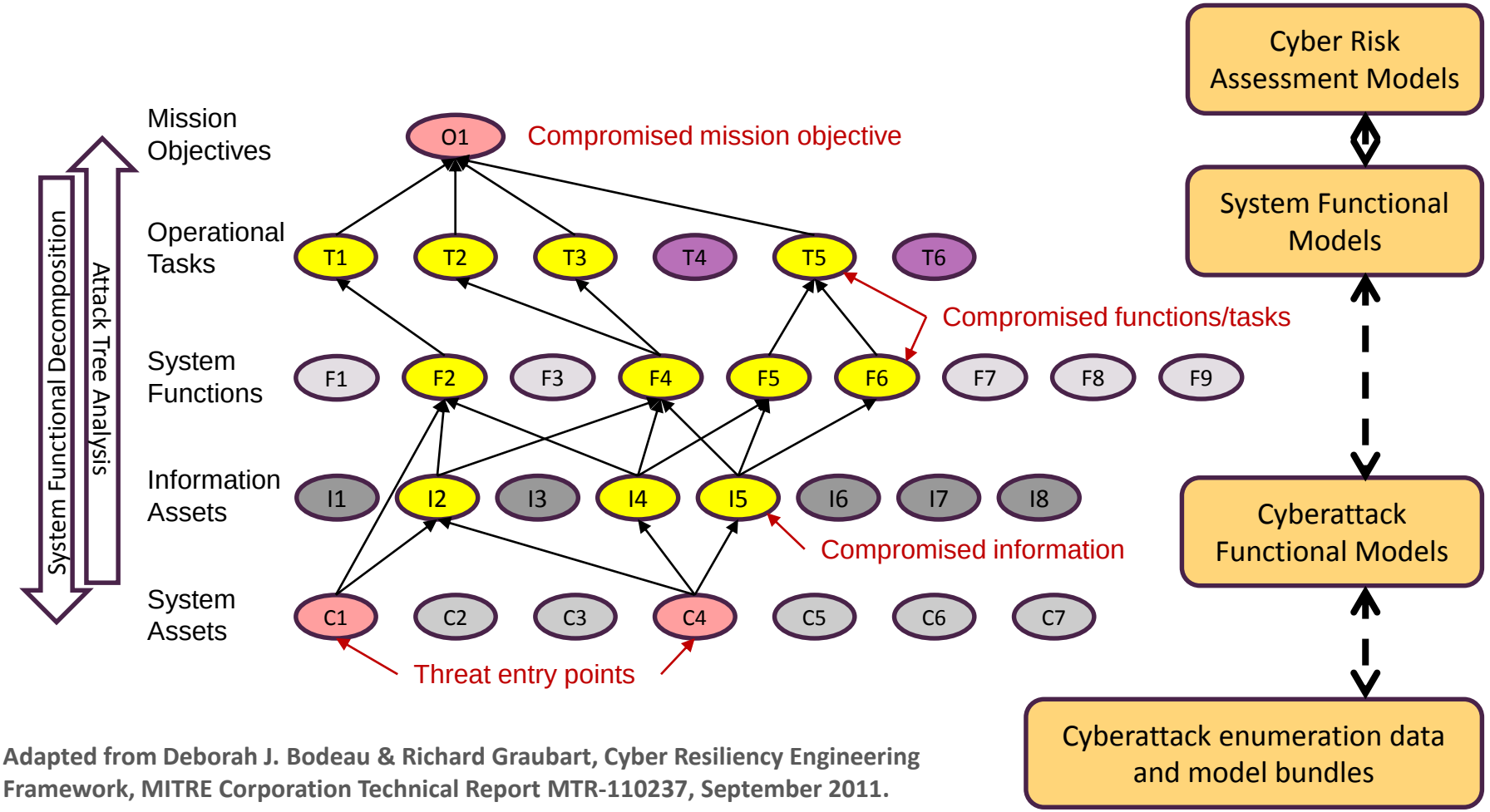


Image from 'SPARQL by Example: The cheat sheet', https://www.slideshare.net/LeeFeigenbaum/sparql-cheat-sheet

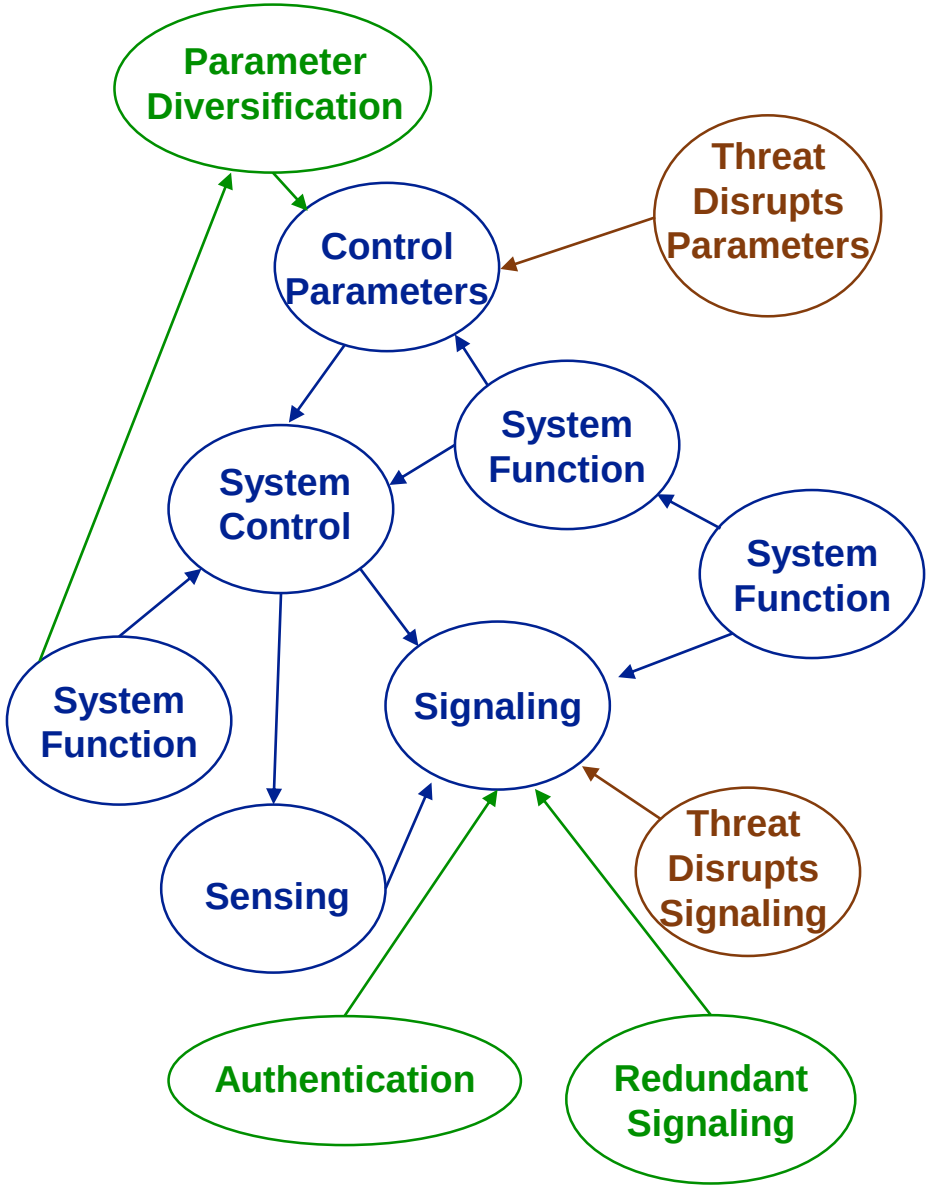
Layered View of a System for Analysis



Adapted from Deborah J. Bodeau & Richard Graubart, Cyber Resiliency Engineering Framework, MITRE Corporation Technical Report MTR-110237, September 2011.

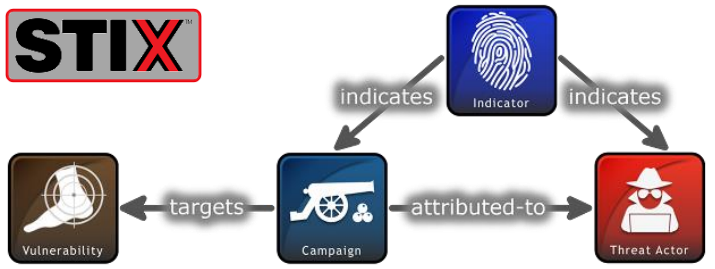
Pattern Matching for Preferential Graph Augmentation

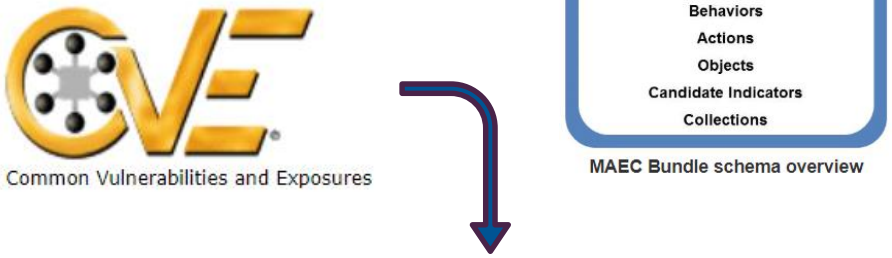
- **Extract system functional information**
  - Reduced system representation in form of Directed Graph
- **Extract relationships between threat vectors and functional assets**
  - Semantic mapping of attack vector descriptors to targeted assets
- **Extract a semantic mapping of Blue design patterns to:**
  - Their functional capabilities
  - Assets they require to achieve capabilities
  - Critical functions/assets they will protect
  - Specific threat capabilities and/or threat assets they are designed to detect or counter through direct connective action



Threat Extraction and Abstraction

- Follow research linking threat data to attack patterns:
  - SERC Security Engineering: CYBOK
  - Siemens M&S of cyberattacks
  - DARPA CASE program





Common Vulnerabilities and Exposures

Common Weakness Enumeration
A Community-Developed List of Software Weakness Types

1008 - Architectural Concepts
- Audit - *(1009)*
- Authenticate Actors - *(1010)*
- Authorize Actors - *(1011)*
- Cross Cutting - *(1012)*
  - Information Exposure Through Timing Discrepancy - *(208)*
  - Missing Report of Error Condition - *(392)*
  - Improper Cleanup on Thrown Exception - *(460)*
  - Missing Standardized Error Handling Mechanism - *(544)*
  - Client-Side Enforcement of Server-Side Security - *(602)*
  - Improper Check or Handling of Exceptional Conditions - *(703)*
  - Improper Check for Unusual or Exceptional Conditions - *(754)*
  - Reliance on Cookies without Validation and Integrity Checking in a Security Decision - *(784)*
  - Reliance on Untrusted Inputs in a Security Decision - *(807)*
- Encrypt Data - *(1013)*

CAPEC

1000 - Mechanisms of Attack
- Collect and Analyze Information - *(118)*
  - Excavation - *(116)*
  - Interception - *(117)*
  - Footprinting - *(169)*
  - Reverse Engineering - *(188)*
  - Protocol Analysis - *(192)*
  - Fingerprinting - *(224)*
- Inject Unexpected Items - *(152)*
  - Parameter Injection - *(137)*
  - Code Inclusion - *(175)*
  - Resource Injection - *(240)*
  - Code Injection - *(242)*
  - Command Injection - *(248)*
    - LDAP Injection - *(136)*
    - Flash Injection - *(182)*
    - IMAP/SMTP Command Injection - *(183)*
    - Linux Terminal Injection - *(249)*
    - XML Injection - *(250)*
    - SQL Injection - *(66)*
      - Command Line Execution through SQL Injection - *(108)*
      - Object Relational Mapping Injection - *(109)*
      - SQL Injection through SOAP Parameter Tampering - *(110)*
      - Expanding Control over the Operating System from the Database - *(470)*
      - Blind SQL Injection - *(7)*
    - OS Command Injection - *(88)*
  - Local Execution of Code - *(549)*
  - Traffic Injection - *(594)*
  - Fault Injection - *(624)*

Base Function Attack Models for CPS

- Fuzzy Attack – adds a new uniform distribution to disrupt the original signal flow.
- Interruption Attack – denies the signal flow.
- Man-in-the-Middle Attack – changes the signal flow to one controlled by the attacker.
- Overflow Attack – extends the number of bits in a digital message to overflow the receive buffer.
- Down-sampling Attack – reduces the sampling rate of the signal flow.
- Control Parameter Attack – replaces control parameters to induce control errors.
- Coordinated Attacks – combinations of these patterns that affect different or redundant system functions.

Rashid, Wan, Quiros, Canedo, Al Faruque; **Modeling and Simulation of Cyberattacks for Resilient Cyber-Physical Systems;** 2017 13th IEEE Conference on Automation Science and Engineering (CASE) Xi'an, China, August 20-23, 2017

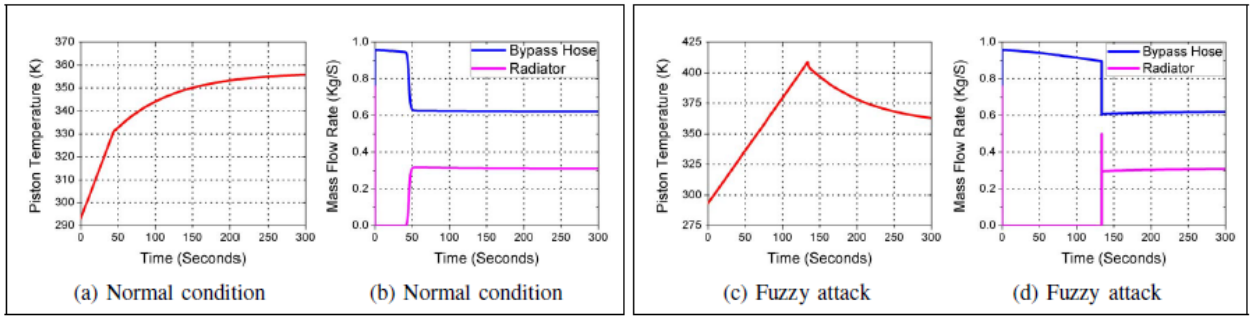


Fig. 3: Simulation results without attacks and with fuzzy attack

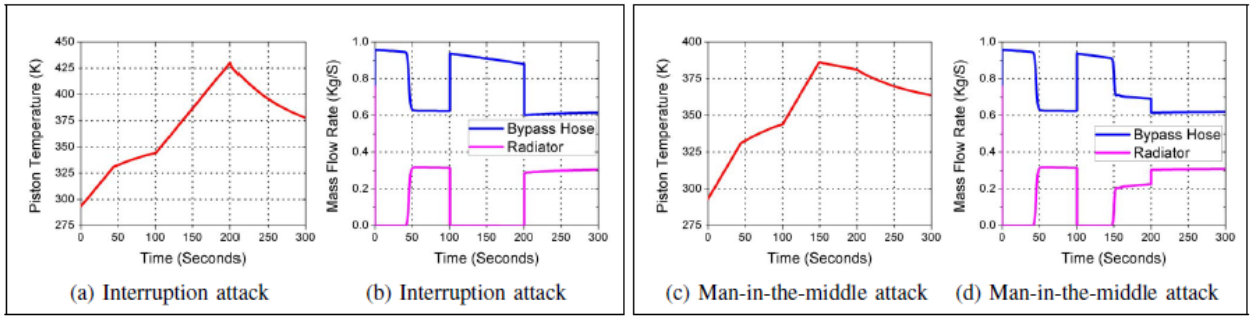
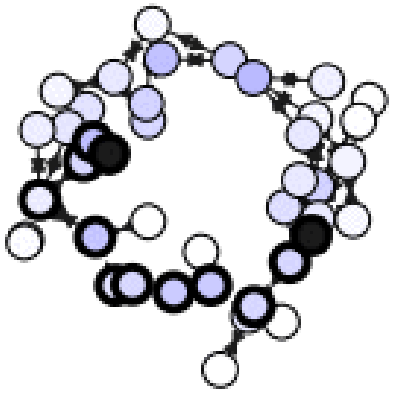


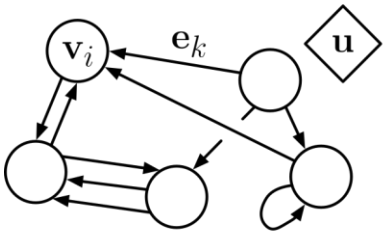
Fig. 4: Simulation results with interruption and man-in-the-middle attack

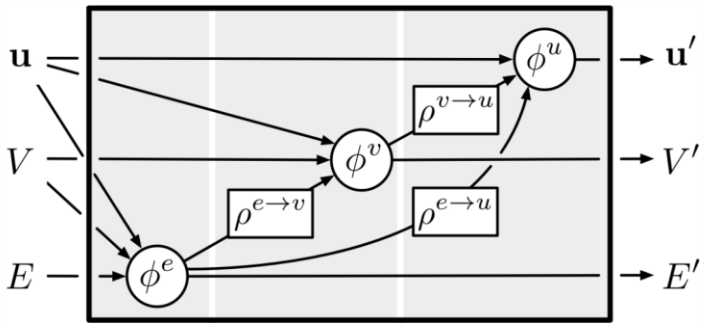
Once We Have a Graph…..

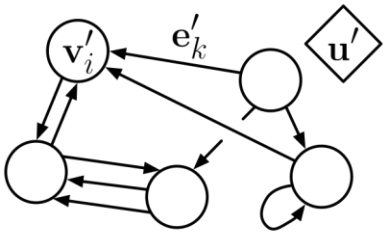
- **Is the functional state space of our system preserved?**
  - Multiple options for dynamic evolution of a graphical state space representation. Are they scalable?

- Graph_nets
  - DeepMind library to create, manipulate, and train graph networks to reason about graph-structured data



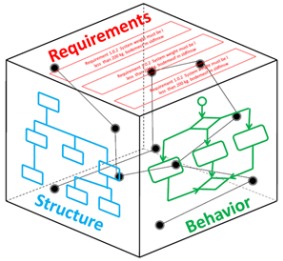




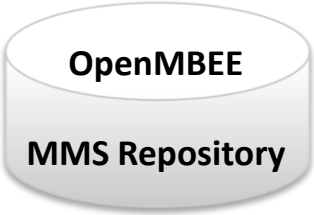


Bringing it Back Together

Formally expressed system model



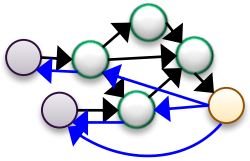
OpenMBEE linked data architecture



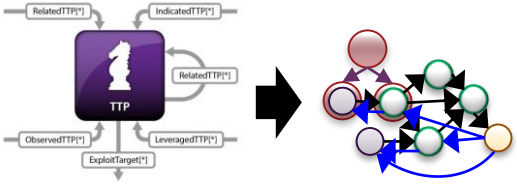
Query-able graph-structured data model

Function and data flow query/ discovery

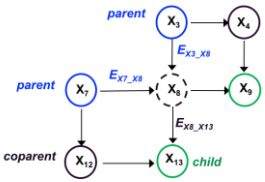
Reduced Graph functional architecture representation



Threat vector functional patterns relevant to system

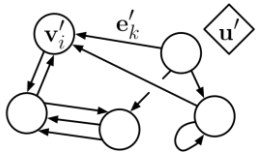


Reduced ecosystem graph



Attribute with dynamic state abstractions

Is functional state of system preserved?



Critical Research Challenges - 1

- We need relevant CPS models
  - 30 m Telescope, Railway Control System
  - Are these models simple enough, rich enough, expressed in the "right" ways for this approach to discover relevant functional abstractions?

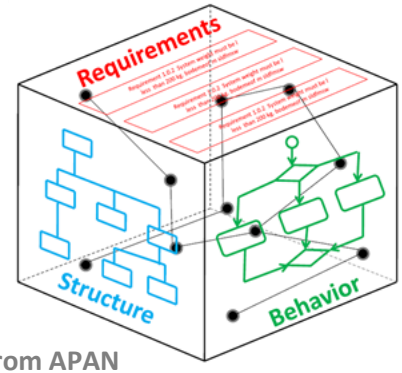
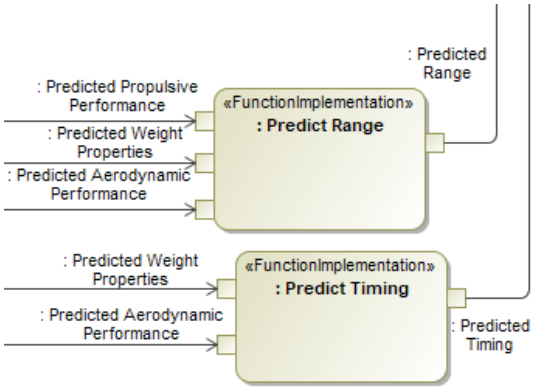


Image from APAN

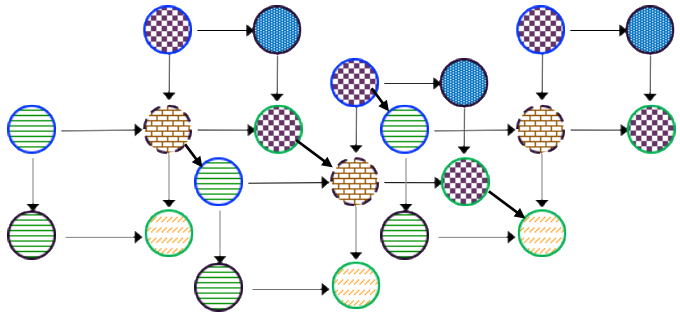
- What modeling formalisms / best practices within do we need to specify for this approach to work?
  - Functional, Logical, Physical Structure linkages?

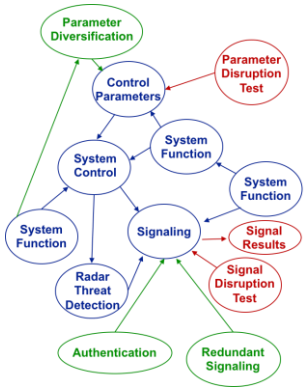
Critical Research Challenges - 2



- How do we discover what constitutes functions and flows relevant to our analysis within our query-able graph data model?
  - Depends significantly on how relationships expressed
  - Impacts how threat functional patterns augment our graph

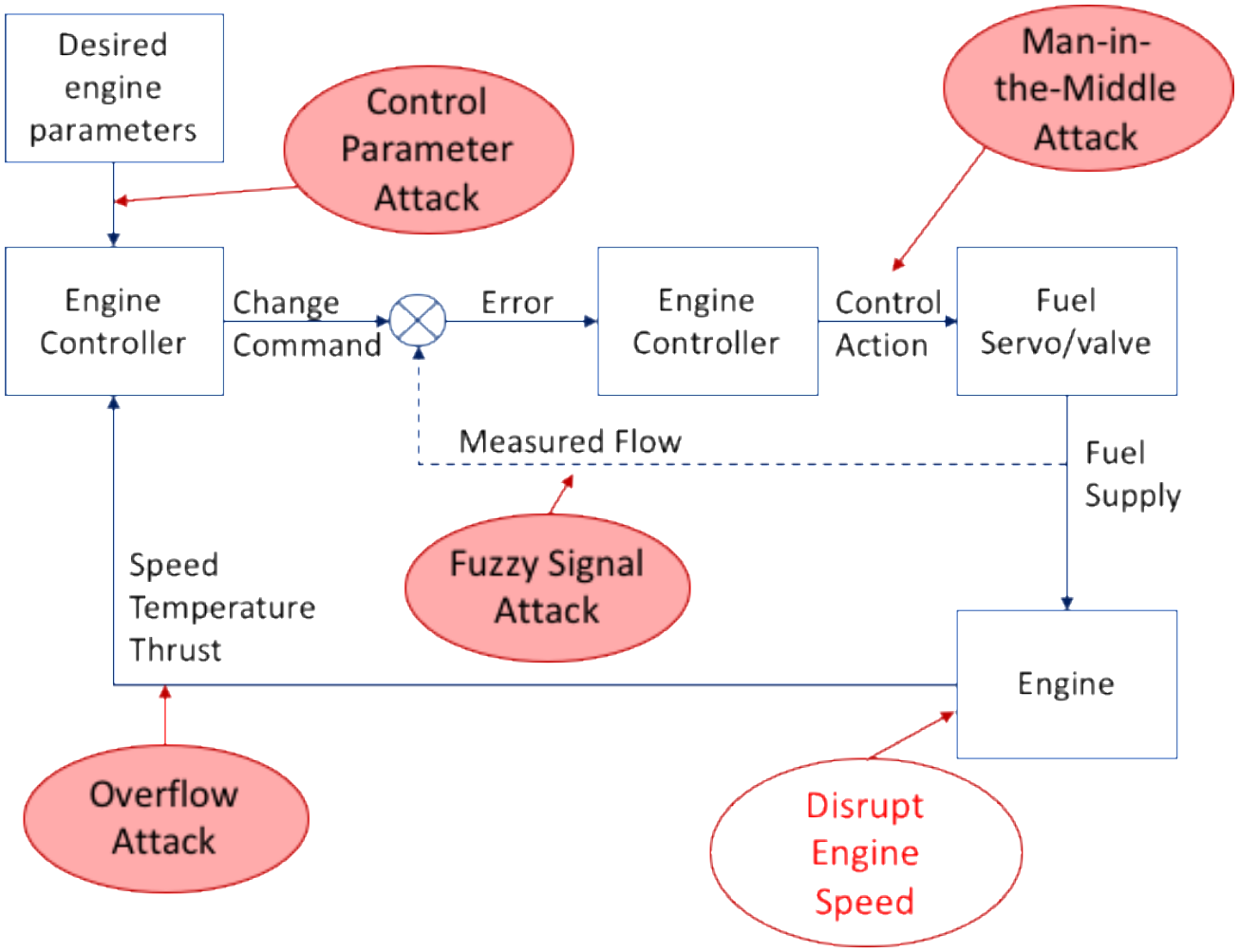
- What executable micropatterns can serve as abstract representations of system function (node states) and data flow (edge states)?
  - Scalability, re-use, and interpretability





- Can this approach produce a path forward to realize a test framework for assurance?
  - Explore threats to functions

Example Coordinated Attack Strategy on a Digital Engine Controller



The key is to create "safe" designs, not respond simply to known threats.

Acknowledgements

Georgia Tech Research Institute

- Mr. Nicholas Bollweg
- Dr. Dane Freeman
- Dr. Frank Patterson
- Dr. Zach Welz

SYSTEMS ENGINEERING RESEARCH CENTER

- Mr. Tom McDermott
- Ms. Megan Clifford

Portions of this material are based upon work supported, in whole or in part, by the United States DoD through the Systems Engineering Research Center (SERC) under Contract HQ0034-13-D-0004. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. The views and conclusions are those of the individual authors and participants, and should not be interpreted as necessarily representing official policies, either expressed or implied, of the DoD, any specific US Government agency, or the US Government in general.