MODEL-INTEGRATED DESIGN IN SOFTWARE, SYSTEMS AND CONTROL ENGINEERING

Janos Sztipanovits
ISIS, Vanderbilt University

SERC Workshop
October 5, 2011
doTransition (fsm as FSM, s as State, t as Transition) =
require s.active
step exitState (s)
step if t.outputEvent <> null then
  emitEvent (fsm, t.outputEvent)
step activateState (fsm, t.dst)

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:
  • Automotive
  • Avionics
  • Sensors…

Tools:
  • Modeling
  • Analysis
  • Verification
  • Synthesis

Challenges:
  • Cost
  • Benefit only narrow domains
  • Island of Automation

Mathematical and physical foundations
**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/)

---

**Domain Specific Design Automation Environments:**
- Automotive
- Avionics
- Sensors...

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

**Explicit Semantic Foundation**
- Structural
- Behavioral
Use Case 1: Cyber Physical Systems

Components span:
- Multiple physics
- Multiple domains
- Multiple tools

Physical
- Functional: implements some function in the design
- Interconnect: acts as the facilitators for physical interactions

Cyber
- Computation and communication that implements some function
- Requires a physical platform to run/to communicate

Cyber-Physical
- Physical with deeply embedded computing and communication

DARPA AVM Program
CPS Design Flow Requires Model Integration

<table>
<thead>
<tr>
<th>Architecture Design</th>
<th>Integrated Multi-physics/Cyber Design</th>
<th>Detailed Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling Exploration</td>
<td>Modeling Simulation V&amp;V</td>
<td>Modeling Analysis</td>
</tr>
<tr>
<td>Rapid exploration Exploration with integrated optimization and V&amp;V</td>
<td>Deep analysis</td>
<td></td>
</tr>
</tbody>
</table>

- Architecture Modeling
- Design Space + Constraint Modeling
- Low-Res Component Modeling

- Architecture Modeling
- Design Space + Constraint Modeling
- Dynamics Modeling (ODE)
- Computational Behavior Modeling
- CAD/Thermal Modeling
- Manufacturing Modeling

- Architecture Modeling
- Dynamics, RT Software, CAD, Thermal, …
- Detailed Domain Modeling (FEM)

Domain Specific Modeling Languages
Physical components are involved in multiple physical interactions (multi-physics)

Model Integration Challenge: Implementation Layers

<table>
<thead>
<tr>
<th>Physical design</th>
<th>Software design</th>
<th>System/Platform Design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Plant Dynamics Models</strong></td>
<td><strong>Controller Models</strong></td>
<td><strong>Resource Management Models</strong></td>
</tr>
<tr>
<td><strong>Software Architecture Models</strong></td>
<td><strong>Software Component Code</strong></td>
<td><strong>System Architecture Models</strong></td>
</tr>
</tbody>
</table>

**Dynamics:** \[ B(t) = \kappa_p (B_1(t),...,B_j(t)) \]
- **Properties:** stability, safety, performance
- **Abstractions:** continuous time, functions, signals, flows,…

**Software:** \[ B(i) = \kappa_c (B_1(i),...,B_k(i)) \]
- **Properties:** deadlock, invariants, security,…
- **Abstractions:** logical-time, concurrency, atomicity, ideal communication,…

**Systems:** \[ B(t_j) = \kappa_p (B_1(t),...,B_k(t)) \]
- **Properties:** timing, power, security, fault tolerance
- **Abstractions:** discrete-time, delays, resources, scheduling,…

**Source of resilience:** systems science principles for decoupling across design layers (such as passive dynamics to decouple stability from implementation induced time-varying delays)
Model Integration Language (MIL)

- Hierarchical Ported Models / Interconnects
- Structured Design Spaces
- Meta-model Composition Operators

Use Case 2: “C2 Wind Tunnel”

Model-Based Experiment Integration Environment: C2WT

Issues to be studied experimentally:

- **Distributed Command and Control**
  - Synchronization and coordination
  - Distributed dynamic decision making
  - Network effects

- **Information Sharing**
  - Shared situation awareness
  - Common Operation Picture (COP)
  - Network effects

- **Advanced Cooperative Control**
  - Cooperative search algorithms

AFOSR PRET Program
How can we integrate the models?
How can we integrate the simulated heterogeneous system components?
How can we integrate the simulation engines?
Model Integration Architecture in C2WT

Simulator

- Delta3D
- Simulink Federate(s)
- HLA-RTI
- OMNet Federate
- CPN Federate
- OMNet

Integration models

- Dataflow models
- Interaction models
- Deployment models

Generators
Simulation Integration Architecture in C2WT

Experiment Specification & Configuration

Model Integration Layer
- Controller Models
- Network Models
- Org. Models
- Fusion Models
- Env. Models

Component Models

Models
- Controller Models
- Network Models
- Org. Models
- Fusion Models
- Env. Models

Run-time
- Simulink Federate
- OmNet++ Federate
- CPN Federate
- DEVS Federate
- OGRE Federate

Instrumentation Layer

Simulation Integration Platform (HLA)

Simulation Data Distribution/Communication Middleware

Distributed Simulation Platform

https://wiki.isis.vanderbilt.edu/OpenC2WT
Example: Simulink model integration (Vehicle dynamics)

Original model (X4 simulator)

GME integration model

Modified model

HLA Run-Time Infrastructure (RTI)

Add input-output bindings

Output binding

Input binding

Code generation

Generated .m Receiver and Sender S-function code + .java code for representing Simulink federate

RTI runtime communication

Signal flow

Signal flow
Experiments: Impact of Cyber Attacks

- **Network attack:**
  - A sub-network with hundreds of zombie nodes attacks a critical router on the main network.
  - Flood attack on udp, tcp or ping
Summary

Questions:

- What are challenging systems application domains?
  Heterogeneous SoS domains (like CPS and C2).

- How does practice diverge from theory, and how do we connect?
  Precise compositionality is hard to achieve in heterogeneous systems, still, we need predictability. Need systems science principles for simplifying interactions and dependences (decoupling).

- Where are relevant technologies to be found?
  In cross-disciplinary interactions. E.g. scalability in embedded software verification may require tradeoffs in systems dynamics.

- What would be the most critical tools and products?
  Component-based and model-based design approaches and tools are and will be increasingly essential.
<table>
<thead>
<tr>
<th>Sublanguage / Capability</th>
<th>Formalism, Language Constructs, Examples</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Modeling</td>
<td>Hierarchical Module Interconnect - Components - Interfaces - Interconnects - Parameters - Properties</td>
<td>Systems Architect - Explore Design Space - Derive Candidate Designs</td>
</tr>
</tbody>
</table>
Example: Dynamics Modeling

**Physical Dynamics Modeling**
- Hybrid Bond Graphs
  - Efforts, Flows,
  - Sources, Capacitance, Inductance,
  - Resistance,
  - Transformers, Gyrators,

**Computational Dynamics Modeling**
- Dataflow + Stateflow + TT Schedule
  - Interaction with Physical Components
  - Cyber Components
  - Processing Components

**Component Engineer**
- model dynamics with Hybrid Bond Graphs

**System Engineers**
- Compose system dynamics

**Domain Engineers**
- design controller

**System Engineers**
- Processor allocate
- Platform Effects
### Example: Physical Structure and Manufacturing Modeling

<table>
<thead>
<tr>
<th>Solid Modeling (CAD / Geometry)</th>
<th>Structural Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Defined with Peer Roles:</td>
<td>- Axis</td>
</tr>
<tr>
<td></td>
<td>- Point</td>
</tr>
<tr>
<td></td>
<td>- Surface</td>
</tr>
<tr>
<td></td>
<td>- CAD Links</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manufacturing Modeling</th>
<th>Component Manuf. Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Make</td>
</tr>
<tr>
<td></td>
<td>- Material</td>
</tr>
<tr>
<td></td>
<td>- Fab Proc</td>
</tr>
<tr>
<td></td>
<td>- Complexity</td>
</tr>
<tr>
<td></td>
<td>- Shape/Wt</td>
</tr>
<tr>
<td>- OTS: Cost/unit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structural Interfaces</th>
<th>Standard Structural Interfaces (ex: SAE #1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Fastener Types, Num# ...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component Engineer</th>
<th>- Defines Structural Interface System Engineer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Defines Architecture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fastener Type</th>
<th>Nuts/Bolts/Washers (Hand)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumbeOfFasteners</td>
<td>12</td>
</tr>
<tr>
<td>FastenerDiameter</td>
<td>0.4375</td>
</tr>
<tr>
<td>FastenerPitch</td>
<td>14</td>
</tr>
<tr>
<td>FastenerEdgeDistance</td>
<td>0</td>
</tr>
</tbody>
</table>