Modeling and Simulation for Testbeds: Semantics and Languages

Building a Model-Based Simulation Integration Platform for Rapid Synthesis of Distributed Heterogeneous Simulations

Janos Sztipanovits

Mar. 31, 2015
Command and Control (C2) Architecture Analysis
(AFOSR/PRET project with UC Berkeley and George Mason, 2006-2009)

C2 issues to be studied experimentally:

- **Distributed Mission Operation**
  - Synchronization and coordination
  - Distributed dynamic decision making
  - Network effects

- **Seamless Integration of Manned/Unmanned Assets**
  - Mixed-Initiative Teams

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

- **System Level Impact Analysis**
  - Cyber attacks and Resilient solutions
  - Strategy/gaming
How can we integrate the simulated heterogeneous system components?
How can we integrate the simulation engines?
How can we rapidly synthesize and deploy integrated simulations?
Integrated control, communication, and power system
(Pilot and joint experiment with WSU, 2011)
5-Bus Example: Power Grid Model

Tool: SimPower/MATLAB
Semantics: Continuous Time
5-Bus Example: Communication Model

Tool: NS-2
Semantics: Discrete Event

Other Tools: OMNeT++
OPNET, TrueTime,..
5-Bus Example: Control Center Model

**Tool:** MATLAB  
**Semantics:** Discrete time

**Other Tools:** DEVS, LabView,  
**Semantics:** Discrete Event
Integration Challenges

Simulation

• Simulators have different timing models
• Execution needs to be coordinated
• Data needs to be shared
• Different time-scale and resolution
• Logical time v.s. real time
• Different simulation engines

How to integrate the simulators?
How to integrate the models?
How to execute coordinated experiments?

• Modeling languages are different
• Semantics is different: continuous time, discrete time, discrete event
• Simulated systems are interacting but modeling languages do not have construct to express them
• No support for specifying experiments
Integrating *models*

- Heterogeneous models for different domains: human organizations, communication networks, C2 software systems, vehicle simulations, etc. These models need to talk to each other somehow.
- Needed: an overarching *integration model* that *connects* and *relates* these heterogeneous domain models in a logically coherent framework.

Integrating *simulations*

- Heterogeneous simulators and emulators for different domains: Colored Petri Nets, OMNET++, DEVS, Simulink/Stateflow, Delta3D, etc.
- Needed: an underlying *software infrastructure* that *connects* and *relates* the heterogeneous simulators in a logically and temporally coherent framework.

**Key idea:** Integration is about messages and shared data across system components. Why don’t we model these messages and shared data elements and use these models to facilitate model and system integration?
Use Case for Model- and Tool Integration: Lumped Parameter Dynamics

Model Integration
(Cyber-Physical/Multi-domain/Multi-Physics Modeling & Composition)

Formal Verification
• Qualitative reasoning
• Relational abstraction
• Model checking

Simulation
• Open Modelica
• Delta Theta
• Dymola

Distributed Co-Simulation
• NS3
• OMNET
• Delta-3D
• CPN

Composition
• Continuous Time
• Discrete Time
• Discrete Event
• Energy flows
• Signal flows
• Geometric

Equations
Modelica-XML

Simulink/
Stateflow
Embedded
Software
Modeling
Hybrid
Bond
Graph
Modelica
TrueTime
Functional
Mock-up
Unit

High Level
Architecture
Interface (HLA)

S-function
FMU-ME
FMU-CS
Model and Simulation Integration Approach in C2WT

**Model Integration Language - MIL**

- Hierarchical Ported Models / Interconnects
- Information Flows
- Interactions

**Semantic Backplane**

- Structural Semantics
- Behavioral Semantics
- Transformation Semantics

**Semantic Interface**

**MIL Translators**

- MIL ↔ SL/SF Fed
- MIL ↔ OMNeT++ Fed
- MIL ↔ SUMO Fed

**Domain Specific Simulation Tools**

- Dymola
- SUMO
- HLA Federates
- HLA BUS

**Model Translators**

- SL/SF IF MetaModel
- OMNeT++ IF MetaModel
- SUMO IF Metamodel

Institute for Software Integrated Systems
World-class, interdisciplinary research with global impact.
Model and Simulation Integration Approach in C2WT

Model Integration Language - MIL

- Hierarchical Ported Models /Interconnects
- Information Flows
- Interactions

Semantic Backplane

- Structural Semantics
- Behavioral Semantics
- Transformation Semantics

Semantic Interface

Domain Specific Simulation Tools

- HLA Federates
- HLA BUS

Model Translators

- MIL ↔ SL/SF Fed
- MIL ↔ OMNet Fed
- MIL ↔ SUMO Fed

Institute for Software Integrated Systems
World-class, interdisciplinary research with global impact.
## Semantic Backplane

<table>
<thead>
<tr>
<th>Functions</th>
<th>(Meta)Models</th>
<th>Languages</th>
<th>Tools</th>
<th>Role</th>
</tr>
</thead>
</table>
| **Metamodelling**          | ![Metamodelling Diagram](image1) | **MetaGME** | • GME (WebGME)  
• WebGME-2-Formula | • DSML spec.  
• Constraint Checking  
• Metaprog.           |
| **Transformation Modeling**| ![Transformation Diagram](image2) | **UMTL**  | • GReAT  
• UDM | • Transf. spec.  
• Compiling spec to transformer |
| **Formal Metamodelling**   | ![Formal Metamodelling](image3) | **Formula (MSR)** | • Domain Comp.  
• Trace Gen.  | • Metamod. checking  
• Example gen.  
• Semantic units |
| **Formal Transformation Modeling** | ![Formal Transformation](image4) | | • Semantic Anchoring | • Semantics for complex DSMLs  
• Composition |
Model and Simulation Integration Approach in C2WT

Model Integration Language - MIL
- Hierarchical Ported Models / Interconnects
- Information Flows
- Interactions

Semantic Backplane
- Structural Semantics
- Behavioral Semantics
- Transformation Semantics

Model Translators
- MIL ↔ SL/SF Fed
- MIL ↔ OMNet Fed
- MIL ↔ SUMO Fed

Domain Specific Simulation Tools
- Dymola
- Modelica
- MATLAB Simulink
- OMNeT++
- SUMO
- Delta 3D

HLA Federates

HLA BUS
What is High-Level Architecture (HLA)?

• An IEEE standard for “interoperable” and “reusable” models and simulations.
  – Most used specification (also used in the demo) is IEEE HLA 1.3 (1998)
  – Most recent specification is IEEE HLA 1516 (2000+)

• DoD-wide policy requires ALL defense models and simulations to comply with the standard.

• Primary goal is to provide a general purpose infrastructure for “distributed” simulation and analysis.

• Software implementing the HLA specification is called Run-Time Infrastructure (RTI).
  – Several commercial and open-source RTIs are available.
  – In the demo we used an open-source RTI PORTICO v2.0.1 implemented in Java language ([http://porticoproject.org/](http://porticoproject.org/)).

• *Semantics of the Model Integration Language for C2WT is defined by HLA’s service models* (federation mgmt., declaration mgmt., object mgmt.time mgmt., data distribution mgmt., ownership mgmt.)
Example: Integration model of a specific C2 scenario

Federates (component simulators) publish and subscribe to various types of interactions. (→)
Specific dataflows across networks are specified via ported federates and dataflow connections (-----)
Federates are autogenerated from integration models
Other tool integration examples and capabilities

**Library of supported tools and mechanisms:**

- **Other simulation tools** (NS-2, Delta3D, Google Earth, Java/C/C++, FMU-CS, etc.)
- **Passive federates** (e.g. Loggers, monitors, etc.)
- **Live components** (e.g. Emergency response, Traffic conditions, Human-in-the-loop, etc.)
- **Advanced support** (e.g. Legacy FOMs, COAs, Expt. Config., Remote deployment, Gaming, etc.)
Ongoing efforts

• With NIST:
  – Building automation with Cyber & Network Effects Analysis

• With AFRL:
  – System Science of SecUrity and REsilience (SURE): Threat modeling, Cyber effects analysis, Resilient Architectures, Decentralized security

• Global Cities Challenge (sequel to Smart America)
  – Real-time Optimized Metro Routes (from an App) based on real-time traffic input and look-ahead of traffic demands based on historical information. Also, support for analytics to improve metro efficiency in a number of ways.
Key URLs and Contact

• Cyber-Physical Systems – Virtual Organization – http://cps-vo.org
• C2WT community wiki – https://wiki.isis.vanderbilt.edu/OpenC2WT
• Functional Mock-up Interface – www.fmi-standard.org

• Contact:

  Himanshu Neema  
  Senior Staff Engineer  
  Institute for Software Integrated Systems, Vanderbilt University  
  Email: himanshu@isis.vanderbilt.edu  
  Ph: +1-615-497-8136, Fax: 343-7440  
  http://www.isis.vanderbilt.edu