Model-Based SE Using SysML

Part 1: Integrating Design and Assessment M&S

Russell Peak, Chris Paredis, Leon McGinnis

Georgia Institute of Technology
Product & Systems Lifecycle Management Center
www.pslm.gatech.edu

Part 1 Speaker: Russell Peak

Note: This is the 99-slide “standard edition” presentation. A 187-slide “extended edition” with additional context material is available here: http://www.pslm.gatech.edu/projects/incose-mbse-msi/
Model-Based SE Using SysML

Part 1: Integrating Design and Assessment M&S

Abstract

This presentation highlights Phase 1 results from a modeling & simulation effort that integrates design and assessment using SysML. An excavator testbed illustrates interconnecting simulation models with associated diverse system models, design models, and manufacturing models. We then overview Phase 2 work-in-process including a mobile robotics testbed and associated SysML-driven operations demonstration.

The overall goal is to enable advanced model-based systems engineering (MBSE) in particular and model-based X (MBX) [1] in general. Our method employs SysML as the primary technology to achieve multi-level multi-fidelity interoperability, while at the same time leveraging conventional modeling & simulation tools including mechanical CAD, factory CAD, spreadsheets, math solvers, finite element analysis (FEA), discrete event solvers, and optimization tools.

This Part 1 presentation overviews the project context and several specific components. Part 2 focuses on manufacturing aspects including factory design, process planning, and throughput simulation.

This work is sponsored by several organizations including Lockheed and Deere and is part of the Modeling & Simulation Interoperability Team [2] in the INCOSE MBSE Challenge (with applications to mechatronics as an example domain).

[1] The X in MBX includes engineering (MBE), manufacturing (MBM), and potentially other scopes and contexts such as model-based enterprises (MBE).

Citations


Contact

Russell.Peak@gatech.edu, Georgia Institute of Technology, Atlanta, www.msl.gatech.edu
Collaboration Approach
Primary Current Team

• Deere & Co.
  - Roger Burkhart

• Georgia Institute of Technology (GIT)
  - Russell Peak, Chris Paredis, Leon McGinnis, & co.
  - Leveraging collaborations in
    PSLM Center SysML Focus Area (see next slide)

• Lockheed Martin
  - Sandy Friedenthal

• Vendor collaboration
GIT Product & Systems Lifecycle Management Center

Leveraging Related Efforts

www.pslm.gatech.edu

- SysML-related projects:
  - Deere, Lockheed, Boeing, NASA, NIST, TRW Automotive, ...
- Other efforts based at GIT:
  - NSF Center for Compact & Efficient Fluid Power
  - SysML course development
    - For Professional Masters in SE program, continuing ed. short courses, ...
  - Other groups & labs
  - Vendor collaboration (tool licenses, support, ...)
- Consortia & other GIT involvements:
  - INCOSE Model-Based Systems Engineering (MBSE) effort
  - NIST SE Tool Interoperability Plug-Fest
  - OMG (SysML, ...)
  - PDES Inc. (APs 210, 233, ...)
- Commercialization efforts:
  - www.VentureLab.gatech.edu-based spin-off company (InterCAX):
    Productionizing tools for executable SysML parametrics
Contents

• Phase 1 Overview and Results
  – From August, 2007 to August, 2008

• Phase 2 Progress
  – From August, 2008 to August, 2009
Contents

• Problem Description
  – Challenge Team Objectives
  – Characteristics of Mechatronic Systems

• Technical Approach
  – Techniques and Testbeds

• Deliverables & Outcomes

• Collaboration Approach
MBSE Challenge Team Objectives
Phase 1: 2007-2008

Overall Objectives

• Define & demonstrate capabilities for advanced modeling & simulation interoperability (MSI)

• Phase 1 Scope
   
   Domain: Mechatronics
   
   – Capabilities: Methodologies, tools, requirements, and practical applications
   
   – MSI subset: Connecting system specification & design models with multiple engineering analysis & dynamic simulation models

• Test & demonstrate how SysML facilitates effective MSI

Note: The objectives to date are primarily based on projects in the GIT PSLM Center sponsored by industry and government—see backup slides.
MBSE Challenge Team Objectives
Phase 1: 2007–2008

Specific Objectives

1. Define modeling & simulation interoperability (MSI) method
2. Define SysML and tool requirements to support MSI
   1. Provide feedback to vendors and OMG SysML 1.1 revision task force
3. Demonstrate MSI method with 3+ engineering analysis and dynamic simulation model types
   1. Include representative building block library: fluid power
   2. Include hybrid discrete/continuous systems described by differential algebraic equations (DAEs)
4. Develop roadmap beyond Phase 1
Interoperability Method Objectives for MBSE

<table>
<thead>
<tr>
<th>Enabling Capabilities</th>
<th>Reduced Time</th>
<th>Reduced Cost</th>
<th>Reduced Risk</th>
<th>Increased Understanding</th>
<th>Increased Corporate Memory</th>
<th>Increased Artifact</th>
<th>Increased Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increased Knowledge Capture &amp; Completeness</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Increased Modularity &amp; Reusability</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Increased Traceability</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Reduced Manual Re-Creation</td>
<td>□</td>
<td>□</td>
<td></td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Increased Automation</td>
<td>□</td>
<td>□</td>
<td></td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Reduced Modeling Effort</td>
<td>□</td>
<td></td>
<td></td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Increased Analysis Intensity</td>
<td></td>
<td></td>
<td></td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>
Mechatronics Architecture

Software

- Functions
- Operating Modes
- State Machines
- Control Systems
- ... 
- Modules, Libraries
- Messages
- Protocols
- Code
- ... 

Interface

- Displays
- User Controls
- Haptics
- Remote Links
- ... 

“Mechanical System”

- Kinematics & Dynamics
- Powertrain
- Thermal
- Fluids
- Electric Power
- ... 

Electronics

Electronic Control Unit (ECU)

Actuators

Sensors

Communications Bus

Controller

Feedback Control Loop

Reference Sensor
Contents

• Problem Description
  – Characteristics of Mechatronic Systems
  – Challenge Team Objectives

• Technical Approach
  – Techniques and Testbeds

• Deliverables & Outcomes

• Collaboration Approach
Overall Technical Approach

• Technique Development
  – “Federated system model” framework technology
    • A.k.a. collective product model
  – Modeling & simulation interoperability (MSI) method
  – Graph transformation technology
    – etc.

• Testbed Implementations & Execution

• Iteration
Technical Approach—Subset

• Standards-based framework technology
  – Federated system models
  – Utilize SysML where appropriate (esp. parametrics)

• Modeling & simulation interoperability (MSI) method
  – Harmonize, generalize, extend new & existing work
  – COBs, CPM, KCM, MACM, MRA, OOSEM, ...

• Testbeds
  – Develop and test techniques iteratively
  – Implement test cases for verification & validation
  – Produce reference examples
  – Produce open resources
    (e.g., SysML-based fluid power libraries)
Example Federated System Model

Logical composition of models based on various ontologies/schemas (from native tools, standards, in-house)

Fluid Dynamics
- Standard: CFD
- Software -
- Status: In Development
- Boeing,

Optics
- Standard: NODIF
- Software - TBD
- Minolta, Olympus

Structural Analysis
- Standard: AP209
- Software: MSC Patran, Thermal Desktop
- Status: In Production
- Lockheed Martin, Electric Boat

Thermal Radiation Analysis
- Standard: STEP-TAS
- Software: Thermal Desktop, TRASYS
- Status: In Production
- ESA/ESTEC, NASA/JPL & Langely

Propulsion
- Standard: STEP-PRP
- Software:
- Status: In Development
- ESA, EADS

Electrical Engineering
- Standard: AP210
- Software: Mentor Graphics
- Status: Prototyped
- Rockwell, Boeing

Cabling
- Standard: AP212
- Software: Mentor Graphics
- Status: Prototyped
- Daimler-Chrysler, ProSTEP

Software Engineering
- Standard: UML - (AP233 interface In Development)
- Software: Rational Rose, Argo, All-Together
- Status: In Production
- Industry-wide

Mechanical Engineering
- Standard: AP203, AP214
- Software: Pro-E, Caddis, SolidWorks, AutoCad, SDRC IDEAS, Unigraphics, others
- Status: In Production
- Aerospace Industry Wide, Automotive Industry

Systems Engineering
- Standard: AP233
- Software: Statemate, Doors, Matrix-X, Slate, Core, RTM
- Status: In development / Prototyped
- BAE SYSTEMS, EADS, NASA

PDM
- Standard: STEP PDM Schema/AP232
- Software: MetaPhase, Windchill, Insync
- Status: In Production
- Lockheed Martin, EADS, BAE SYSTEMS, Raytheon

Machining
- Standard: STEP-NC/AP224
- Software: Gibbs,
- Status: In Development / Prototyped
- STEP-Tools, Boeing

Inspection
- Standard: AP219
- Software: Technomatics, Brown, eSharp
- Status: In Development
- NIST, CATIA, Boeing, Chrysler, AIAG

Life-Cycle Management
- Standard: PLCS
- Software: SAP
- Status: In Development
- BAE SYSTEMS, Boeing, Eurostep

Spacecraft Development

Adapted from 2001-12-16 - Jim U'Ren, NASA-JPL
Model-Centric Framework
*Produce, Merge, Enrich, Consume*

http://eislab.gatech.edu/pubs/journals/2004-jcise-peak/ (where “collective product model” ≡ “federated system model”)

Producer Tools
(Primary Authoring)

Enricher Tools
(Secondary Authoring)

Consumer Tools
(e.g., Solvers)

Federated System Model
*Meta-Building Blocks:*
- Information models & meta-models
  - International standards
  - Industry specs
  - Corporate standards
  - Local customizations
- Modeling technologies:
  - Express, UML, SysML, COBs, OWL, XML, …
Technical Approach—Subset

• Standards-based framework technology
  - Federated system models
  - Utilize SysML where appropriate (esp. parametrics)

• Modeling & simulation interoperability (MSI) method
  - Harmonize, generalize, extend new & existing work
  - COBs/SysML, CPM, KCM, MACM, MRA, OOSEM, ...

• Testbeds
  - Develop and test techniques iteratively
  - Implement test cases for verification & validation
  - Produce reference examples
  - Produce open resources
    (e.g., SysML-based fluid power libraries)
The Four Pillars of SysML

1. Structure
2. Behavior
3. Requirements
4. Parametrics
Model vs. Diagrams

Reality
- Envisioned or actual

Model
- Computer-oriented
- Master repository
- Complete for intended scope

Diagrams
- Human-oriented
- Subset views

Tools
- Authoring, viewing, executing, ...

Acknowledgements: Selected portions from Friedenthal et al. 2008 and MagicDraw samples.
SysML Technology Status

www.omgsysml.org

• Spec v1.0: 2007-09  v1.1: 2008-11  v1.2: WIP
  v2.x: RFI preparation workshop - 2008-12
  http://www.omg.org/spec/SysML/

• Vendor support

• Learning infrastructure
  – Books, vendor courses, academic courses,
    INCOSE/OMG tutorial, public examples, etc.

• Growing production usage
  – http://www.pslm.gatech.edu/events/frontiers2008/
  – OMG SysML Info Days – 2008-12

• Overall status: Healthy and growing 😊
“Wiring Together” Diverse Models via SysML
Level 1: Intra-Template Diversity

[Diagram showing mechanical CAD model, CAE model (FEA), and symbolic math models]

[Peak et al. 2007—Part 2]
“Wiring Together” Diverse Models via SysML
Level 2: Inter-Template Diversity (per MIM 0.1)

Naval Systems-of-Systems (SoS) Panorama—An Envisioned Complex Model Interoperability Problem Enabled by SysML/MIM/COBs

Based on HMX 0.1
2008-02-20

**Legend**
- Parametric associativity
- Tool & native model associativity
- Composition relationship (re-usage)

**a0. Descriptive Resources**
ECAD & MCAD Tools
Tribon, CATIA, NX, Cadence, ...

Systems & Software Tools
DOORS, E+, MagicDraw, Studio, Eclipse, ...

Operation Mgt. Systems

Libraries & Databases
Classification Codes, Materials, Personnel, Procedures, ...

**b0. Federated Descriptive Models**

**c0. Context-Specific Models**

**c1. Simulation Templates**
(of diverse behavior & fidelity)

**c2. Optimization Templates**

**d0. Simulation Building Blocks**

**e0. Solver Resources**
Evacuation Codes
Egress, Exodus, ...

General Math
Mathematica, Maple, Matlab ...

CFD
Flotherm, Fluent, ...

FEA
Abaqus, Ansys, Patran, Nastran, ...

Discrete Event
Arena, Quest, ...

2D
Propeller Hydrodynamics

3D

Damaged Stability

Navigation Accuracy
Technical Approach—Subset

• Standards-based framework technology
  – Federated system models
  – Utilize SysML where appropriate (esp. parametrics)

• Modeling & simulation interoperability (MSI) method
  – Harmonize, generalize, extend new & existing work
  – COBs, CPM, KCM, MACM, MRA, OOSEM, ...

• Testbeds
  – Develop and test techniques iteratively
  – Implement test cases for verification & validation
  – Produce reference examples
  – Produce open resources
    (e.g., SysML-based fluid power libraries)
Excavator Modeling & Simulation Testbed

Tool Categories View

SysML Tools
- RSA/E+ / SysML
  - Factory Model
- No Magic / SysML
  - Excavator System Model
- RSA/E+ / SysML
  - Excavator Executable Scenario

Interface & Transformation Tools (VIATRA, XaiTools, ...)

Traditional Descriptive Tools
- NX / MCAD Tool
  - Excavator Boom Model
- FactoryCAD
  - Factory Layout Model
- Excel
  - Production Ramps

Traditional Simulation & Analysis Tools
- ModelCenter
  - Optimization Model
- Ansys
  - FEA Model
- Mathematica
  - Reliability Model
- Excel
  - Cost Model
- Dymola
  - Dig Cycle Model
- eM-Plant
  - Factory Simulation

2008-02-25a

Georgia Institute of Technology
Excavator Modeling & Simulation Testbed

Interoperability Patterns View (MSI Panorama per MIM 0.1)

- **a0. Descriptive Resources** (Authoring Tools, ...)
  - MCAD Tools
    - NX
  - Data Mgt. Tools
    - Excel
  - System & Req. Tools
    - MagicDraw
  - RSO/E+
  - Factory CAD Tools
    - FactoryCAD

- **b0. Federated Descriptive Models**
  - Federated Excavator Model
  - Operations
  - Hydraulics Subsystem
  - Boom Subsystem
  - Dig Site
  - Dump Trucks

- **c0. Context-Specific Simulation Models**
  - Excavator Sys-Level Models
    - Optimization Model
      - Objective Function
    - Cost Model
    - Reliability Model
    - Dig Cycle Model
  - Boom Linkage Models
    - Stress/Deformation Models
      - Extensional Linkage Model
      - Plane Stress Linkage Model
  - Boom Mfg. Assembly Models
  - Assembly Process Models
    - MM1 Queuing Assy Model
    - Discrete Event Assy Model

- **d0. Simulation Building Block Libraries**
  - Cost Concepts
  - Optimization Concepts
  - Reliability Concepts
  - Solid Mechanics
  - Fluid Mechanics
  - Queuing Concepts

- **e0. Solver Resources**
  - Optimizers
    - ModelCenter
  - Generic Math Solvers
    - Excel
    - Mathematica
  - Sys Dynamics Solvers
    - Dymola
  - FEA Solvers
    - Ansys
  - Discrete Event Solvers (Specialized)
    - eM-Plant / Factory Flow
Demo Scenario

• New market-driven targets:
  – 20% increase in dig rate (dirt volume / time)
  – 15% increase in mfg. production

• Check if existing design is sufficient by re-running SysML-enabled simulations

• If not, explore re-design trade space
  – Changes in bucket size, hydraulics, ...

• Re-do V&V using simulations on new design

• Explore manufacturing impact
  – Factory re-design and simulation
Excavator Modeling & Simulation Testbed

**Tool Categories View**

**SysML Tools**
- RSA/E+ / SysML
  - Factory Model
- No Magic / SysML
  - Excavator System Model
- RSA/E+ / SysML
  - Excavator Executable Scenario

**Interface & Transformation Tools**
(VIATRA, XaiTools, ...)

**Traditional Descriptive Tools**
- NX / MCAD Tool
  - Excavator Boom Model
- FactoryCAD
  - Factory Layout Model
- Excel
  - Production Ramps

**Traditional Simulation & Analysis Tools**
- ModelCenter
  - Optimization Model
- Ansys
  - FEA Model
- Mathematica
  - Reliability Model
- Excel
  - Cost Model
- Dymola
  - Dig Cycle Model
- eM-Plant
  - Factory Simulation

2008-02-25a
Earth-Moving Enterprise

SysML package diagram (pkg)
Excavator Model Tree

Summary View (mostly unexpanded) in MagicDraw SysML Tool

- Life-cycle Domain (by Administrator)
  - Earth-moving_Domain_Life-cycle_As-Is (by Administrator)
  - Earth-moving_Domain_Life-cycle_To-Be (by Administrator)
    - Relations
      - ExcavatorA123_Life-cycle (by Administrator)
        - Development (by Administrator)
          - Relations
            - Analysis (aimed at artifacts owned by design phase) (by Administrator)
            - Requirements (by Administrator)
            - This is a ... (by Administrator)
        - End-of-life_Disassembly_&_Recycling (by Administrator)
        - Excavator (by Administrator)
          - Interface_Definitions (by Administrator)
          - Item_&_Interface_Definitions (by Administrator)
          - Maintenance (by Administrator)
          - Manufacturing (by Administrator)
          - Model_Support (by Administrator)
            - CT_Definitions (by Administrator)
            - Stereotypes (by Administrator)
            - ValueTypes (by Administrator)
          - MOE_Parametrics (by Administrator)
          - Operational (by Administrator)
          - Site (by Administrator)
            - Use_Cases (by Administrator)
            - viewpoints (by Administrator)
          - Earth_moving_Life-Cycle_Domain (by Administrator)
          - Excavator (by Administrator)
          - Operational_Domain (by Administrator)
          - Top-level_BDD (by Administrator)
          - [Property] (by Administrator)
        - Dig_cycle_Relationships (by Administrator)
        - Hydraulic_system_relationships (by Administrator)
        - ProjectCaseStudy (by Administrator)
        - Structural_Linkage (by Administrator)
        - Earth-moving_Domain_Life-cycle_BDD (by Administrator)
        - Excavator_Life-cycle_System_BDD (by Administrator)
        - Earth-moving_Domain_Life-cycle_Model_Organization (by Administrator)
Excavator Operational Domain

Top-Level Context Diagram in SysML

Excavator Description

The Excavator is used as a part of a larger enterprise designed to perform earth removal. An excavator is assigned to remove dirt from a specific location and place it into a truck for transport to a new location. The excavator operates in a dig cycle mode to perform this operation.

The Dump Truck is used to receive dirt and move and unload it when it becomes full of full volume or mass capacity.

The Soil is what is removed by the excavator, and may vary in consistency and is uncertain in properties, as well as depends on atmospheric conditions for some properties.

The Atmosphere affects the soil, as well as is used by the Excavator Drive-train to produce mechanical energy.

The operator controls the excavator position, velocity, and loading characteristics.

The maintainer access the excavator to replenish fuel, oil, and grease, as well as perform long-term maintenance.

The Dig Cycle use case governs the actions of the excavator as it removes dirt from the ground and places it in the truck.

The Maintain use case governs the timings and periods for which maintenance must occur. Typically the timing is much greater than the length of time required to fill a truck from multiple dig cycles, and is at a minimum daily startup check.
Excavator Operational Domain

First Level of Detail—bdd (SysML block definition diagram)
Excavator Operational Domain

First Level of Detail—ibd (SysML internal block diagram)
Excavator Operational Domain

Top-Level Use Cases
Excavator Dig Cycle

Activity Diagram

1. Identify Dig Position
2. Within Reach?
   - Yes
   - Truck available?
     - Yes
     - Extend Bucket to penetration
     - Retract Bucket and sweep back
     - Lift Bucket for Ground Clearance
     - Pivot Carriage for Truck Alignment
     - Truck Capacity Available?
       - Yes
       - Lift Bucket for Truck Clearance
       - Retract Bucket
9. Quantity Excavated OK?
   - Yes
   - No
   - Pivot Carriage for Penetration Alignment
   - No
   - Idle
   - No
Excavator Requirements & Objectives

req - SysML Requirements Diagram
System Objective Function—Excavator
Context: Operational Enterprise

Mathematical Form

$$f = \sum_{i=1}^{n} k_i \text{moe}_i + \sum_{i,j; i \neq j}^{n} k_{ij} \text{moe}_i \text{moe}_j$$

SysML Parametrics Form
Excavator Test Case

Selected System Breakdowns

- Excavator
  - Dig_Cycle_Time: Real
  - Cycle_Load_Size: Real
  - Fuel_Burn_rate: Fuel_Flow

- MechanicalSubsystem
  - Arm
  - CarriageTruss
  - CarriageBody

- ElectricalSubsystem

- ControlsSubsystem

- HydraulicsSubsystem
  - PowerSubsystem
    - Tank: Tank
    - Pump: VariableDisplacementPump

- FlowControlSubsystem
  - ArmServo: 5Port3WayServo
  - bucketServo: 5Port3WayServo
  - swingServo: 5Port3WayServo
  - boomCheck: CheckValve
  - armCheck: CheckValve
  - bucketCheck: CheckValve
  - swingCheck: CheckValve
  - boomServo: 5Port3WayServo
  - relief: ReliefValve
  - comparator1: ShuttleValve
  - comparator2: ShuttleValve
  - comparator3: ShuttleValve

- ActuationSubsystem
  - armCyl: DoubleActingHydraulicCylinder
  - boomCyl1: DoubleActingHydraulicCylinder
  - boomCyl2: DoubleActingHydraulicCylinder
  - bucketCyl: DoubleActingHydraulicCylinder
  - swingMotor: HydraulicMotor
  - DoubleActingHydraulicCylinder
  - Reliability: Distributed Property

- BucketMechanism
  - Boom
  - Crowd
  - Bucket
Excavator Modeling & Simulation Testbed

Tool Categories View
Hydraulic Circuit Diagram

Pressure-Compensated, Load-Sensing Excavator—ISO 1219 notation

Engineering Schematic

Mechanical Interface
SysML Schematic (ibd) —— Basic View

Pressure-Compensated, Load-Sensing Excavator
SysML Schematic (ibd) — Detailed View

Pressure-Compensated, Load-Sensing Excavator
Hydraulics Subsystem Simulation Model

```
<<block>>
<<external>>
BodyMechanics
url : String = ../../../javaModel/ExcavatorModelBlock.java
name : String = model
baseFrame : Frame_b
cyBoomRightBase : TransFlange_b
cyBoomLeftBase : TransFlange_b
cyArmBase : TransFlange

cyBoomRightRod : TransFlange_a
cyBoomLeftRod : TransFlange_a
cyArmRod : TransFlange_a
cyBucketRod : TransFlange_a
swingFlange : RotFlange
```

```
<<block>>
<<external>>
World
url : String = ../../../javaModel/Library/modelica/2.2.0/package.mo
name : String = model
frame_b : Frame_b
```

```
<<block>>
<<external>>
Excavator_Dig_Cycle

<<constraint>>
Translode2
parameters
a : TransFlange_a
b : TransFlange_b
```

```
<<block>>
<<external>>
Hydraulics
boomCyBaseR : TransFlange_a
boomCyBaseL : TransFlange_a
armCyBase : TransFlange_a
bucketCyBase : TransFlange_a
swingFlange : RotFlange
commandSignal : RealInput
boomCyRodR : TransFlange_b
boomCyRodL : TransFlange_b
armCyRod : TransFlange_b
bucketCyRod : TransFlange_b
boom_s_int : Real = 1.19
bucket_s_int : Real = 0.77
arm_s_int : Real = 0.71
```

```
<<block>>
<<external>>
DigCycleCommands

commandSignal : RealOutput
```

```
<<block>>
```

```
<<block>>
```
Excavator Case Study

Native Tool Models: Modelica

Hydraulics Model

Multi-Body System Dynamics Model
(linkages, ...)

Multi-Body System Dynamics Model
(environment

p_amb = 101325
T_amb = 288.15)
Simulation in Dymola

Modelica Lexical Representation (auto-generated from SysML)

package ExcavatorExample

class ExcavatorDigCycle

end ExcavatorDigCycle;

cmd ExcavatorExample;

[Johnson, 2008 - Masters Thesis]
Excavator Modeling & Simulation Testbed

Tool Categories View

SysML Tools
- RSA/E+ / SysML
  - Factory Model
- No Magic / SysML
  - Excavator System Model
- RSA/E+ / SysML
  - Excavator Executable Scenario

Interface & Transformation Tools
(VIATRA, XaiTools, ...)

Traditional Descriptive Tools
- NX / MCAD Tool
  - Excavator Boom Model
- FactoryCAD
  - Factory Layout Model
- Excel
  - Production Ramps

Traditional Simulation & Analysis Tools
- ModelCenter
  - Optimization Model
- Ansys
  - FEA Model
- Mathematica
  - Reliability Model
- Excel
  - Cost Model
- Dymola
  - Dig Cycle Model
- eM-Plant
  - Factory Simulation

2008-02-25a
Recurring Problem: Maintaining Multiple Views

- Multiple stakeholders with different views and tools
- Models of different system aspects
- Different views are not independent
Approach: Model Transformation

1. Define meta-models
2. Define a model transformation
   - Create graphs of correspondence between meta-models
   - Define transformation rules from SysML to Modelica and vice-versa
   - Triple Graph Grammar (TGG)
3. Compile rules (MOFLON) and load as plug-in

(Czarnecki, K., & Hellen, S., 2006)
Capturing Domain Specific Knowledge in Graph Transformations*

- **Requirements & Objectives**
- **Topology Generation***
- **System Alternatives**
- **Model Composition***
- **System Behavior Models**
- **Model Translation***
- **Executable Simulations**
- **Simulation Configuration***
- **Design Optimization**

**SysML**

**MAsCoMs**

**Dymola**

**ModelCenter**

**Simulation configuration**

**Optimizer**

**Excavator**

**Latin Hypercube Sampling**

**Total Utility**

**Dig Cycle Traj**

**Arm Swing Bucket**

**Trajectory behavior model**

**Executable Simulations**

Dymola

**Simulation Configuration**

ModelCenter

**Model Translation**

**System Behavior Models**

MAsCoMs

**System Alternatives**

**Topology Generation**

**Requirements & Objectives**
Excavator Modeling & Simulation Testbed

Tool Categories View

SysML Tools

- RSA/E+ / SysML
  - Factory Model
- No Magic / SysML
  - Excavator System Model
- RSA/E+ / SysML
  - Excavator Executable Scenario

Interface & Transformation Tools
(VIATRA, XaiTools, ...)

Traditional Descriptive Tools

- NX / MCAD Tool
  - Excavator Boom Model
- FactoryCAD
  - Factory Layout Model
- Excel
  - Production Ramps

Traditional Simulation & Analysis Tools

- ModelCenter
  - Optimization Model
- Ansys
  - FEA Model
- Mathematica
  - Reliability Model
- Excel
  - Cost Model
- Dymola
  - Dig Cycle Model
- eM-Plant
  - Factory Simulation

2008-02-25a
Wrap Dynamic Simulation as ModelCenter Model in SysML

Fully qualified name points to ModelCenter model

Stereotypes define input/output causality
DOE Model in SysML

LatinHyperCube sampler

Reference Property

Model
Automatic Export to and Execution in ModelCenter
Application in Case Study: Optimization under uncertainty with kriging model

• Optimization under uncertainty
• Latin Hypercube sampler used to predict expected value
• Kriging model used in conjunction with sampler to generate response surface to reduce computational cost

Objectives:
• Maximize Efficiency
• Minimize Cost

Design variables:
• bore diameters
SysML Model
Optimization under uncertainty with kriging model
Trade Study Optimization Results

Auto-generated optimization model in ModelCenter

Design space visualized in ModelCenter

Design optimization model in SysML with auto-updated results
Model-Based SE Using SysML
Part 2: Integrating Mfg Design and Simulation
MCAD-SysML Interface Scenarios

UGS/Siemens NX

RSD/E+

SysML Model

SysML Model Import

Model Changes Propagate to CAD Tool

User SysML Model Manipulation

Parametrics Execution

Simulation Execution*

Engineering Analysis Models

XaiTools COB Services

Georgia Tech XaiTools™

* = work-in-process
MCAD Native Model and Tool UIs
UGS/Siemens NX
MCAD Model (Subset) in SysML

RSD/E+
UAV System Design Problem: LittleEye

Network-Centric Warfare Context — SysML/DoDAF Model

Source: No Magic Inc. and InterCAX LLC
Road Scanner System Problem

LittleEye UAV Squadron

![Diagram of LittleEye System]

NumberMilesScannedPer24Hours = NumberAvailablePlanes * MilesScannedPerHour * 24

NumberAvailableSystems = MINIMUM(NumberAvailablePlanes, NumberAvailableCrews, NumberAvailableFuelLoads)

NumberAvailablePlanes = 0.5 * (NumberAvailablePlanesByDay + NumberAvailablePlanesByNight) * DutyCyclePlane


NumberAvailablePlanesByDay = MINIMUM(NumberPlanes, NumberDayCameras)

NumberAvailablePlanesByNight = MINIMUM(NumberPlanes, NumberNightCameras)

NumberAvailableCrews = NumberCrews * CrewTimeOn

NumberAvailableFuelLoads = FuelSupplyPerDay / DailyFuelLoadPerPlane

Information Provided:

MilesScannedPerHour = 40 mph

CrewTimeOn = .42 (120 hours on over 12 day period)

DutyCycleTurnaround = .23 (30 hours turnaround per 100 flight minutes)

DutyCycleCameraRefit = .02 (15 minutes refit per 12 hour period)

DutyCycleMaintenance = .09 (3 hours maintenance per 30 flight hours)

DailyFuelLoadPerPlane = 50 gallons
Next-generation object-oriented spreadsheet-like capabilities.
Enabling Executable SysML Parametrics
Commercialization by InterCAX LLC in Georgia Tech VentureLab incubator program

*Advanced technology for graph management and solver access via web services.*

Plugins Prototyped by GIT (to SysML vendor tools)
1) Artisan Studio [2/06]
2) EmbeddedPlus [3/07]
3) NoMagic [12/07]

COB Services (constraint graph manager, including COTS solver access via web services)

Native Tools Models

COTS = commercial-off-the-shelf (typically readily available)

Traditional COTS or in-house solvers

COB API

Next-Generation Spreadsheet

Execution via API messages or exchange files

Ansys (FEA Solver)

Mathematica (Math Solver)

\[ \Delta L = \frac{FL}{EA} + \alpha \Delta L \]
## Productionizing/Deploying GIT XaiTools™ Technology for Executing SysML Parametrics

[www.InterCAX.com](http://www.InterCAX.com)

<table>
<thead>
<tr>
<th>Vendor</th>
<th>SysML Tool</th>
<th>Prototype by GIT</th>
<th>Product by InterCAX LLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artisan</td>
<td>Studio</td>
<td>Yes</td>
<td>&lt;tbd&gt;</td>
</tr>
<tr>
<td>EmbeddedPlus</td>
<td>E+ SysML / RSA</td>
<td>Yes</td>
<td>&lt;tbd&gt;</td>
</tr>
<tr>
<td>No Magic</td>
<td>MagicDraw</td>
<td>Yes</td>
<td>ParaMagic™ 15.5 (Jul 21, 2008 release)</td>
</tr>
<tr>
<td>Telelogic/IBM</td>
<td>Rhapsody/Tau</td>
<td>&lt;tbd&gt;</td>
<td>&lt;tbd&gt;</td>
</tr>
<tr>
<td>Sparx Systems</td>
<td>Enterprise Arch.</td>
<td>&lt;tbd&gt;</td>
<td>&lt;tbd&gt;</td>
</tr>
<tr>
<td>n/a</td>
<td>XMI import/export</td>
<td>Yes</td>
<td>&lt;tbd&gt;</td>
</tr>
<tr>
<td>Others &lt;tbd&gt;</td>
<td>Others &lt;tbd&gt;</td>
<td>&lt;tbd&gt;</td>
<td>&lt;tbd&gt;</td>
</tr>
</tbody>
</table>

[1] Full disclosure: InterCAX LLC is a spin-off company originally created to commercialize technology from RS Peak’s GIT group. GIT has licensed technology to InterCAX and has an equity stake in the company. RS Peak is one of several business partners in InterCAX. Commercialization of the SysML/composable object aspects has been fostered by the GIT VentureLab incubator program ([www.venturelab.gatech.edu](http://www.venturelab.gatech.edu)) via an InterCAX VentureLab project initiated October 2007.
Solver Access via XaiTools Web Services (XWS)
S1: General Multi-Solver Setup

Client Machines

Rich Client

XaiTools Client
(e.g. ParaMagic)

SysML-based COB models

Server Machines

XaiTools Web Services
Servlet Container

Apache Tomcat

SOAP Servers

XaiTools Solver Wrappers

FEA Solvers

Ansys, Patran, Abaqus, ...

Math Solvers

Mathematica

Engineering Service Bureau

Syntax/Interpretive Engineering SiB
Solver Access via XaiTools Web Services (XWS)
S2: ParaMagic-Mathematica Setup (current product = XWS 2.2)
Broadly Applicable Technology

Examples of Executable SysML Parametrics

- Road scanning system using unmanned aerial vehicle (UAVs)
- Space systems orbit planning
- Energy systems
- ... 
- Mechanical part design and analysis (FEA)
- ... 
- Insurance claims processing and website capacity model
- Financial model for small businesses
- Banking service levels model
- ...
Using a Spectrum of Modeling Technologies

• Spectrum
  – Mental calculations
  – Back-of-envelope hand calculations
  – Spreadsheets
  – ... 
  – SysML (with executable parametrics)
  – ... 

• Varying characteristics
  – Quickness, flexibility, structure, modularity, reusability, self-validation/constraints, ...
Contents

• Problem Description
  – Characteristics of Mechatronic Systems
  – Challenge Team Objectives

• Technical Approach
  – Techniques and Testbeds

• Deliverables & Outcomes

• Collaboration Approach
Deliverables & Outcomes
Phase 1 (Aug 2008)

• Solution and supporting models
  – Excavator test case models, test suites, …

• MBSE practices used
  – Modeling & simulation interoperability (MSI) method, …

• Model interchange capabilities
  – Tests between SysML tools, CAD/CAE tools, …

• MBSE metrics/value
  – See “Benefits” slide with candidate metrics

• MBSE findings, issues, & recommendations
  – Issue submissions to OMG and vendors, publications, …

• Training material
  – Examples, tutorials, …

• Plan forward (Phase 2 and beyond)
Primary Public Reporting Venues

- Call for Participation @ IS’07
  - Jun 26, 2007 in San Diego
- Phase 1 Status Update @ IW’08 MBSE Workshop #2
  - Jan 25, 2008 in Albuquerque
- Phase 1 Status Update @ Frontiers Workshop
  - May 14, 2008 in Atlanta
- Phase 1 Status Update @ IS’08
  - Jun 15-19, 2008 in Utrecht
- Phase 1 Final Report & Archive of Models
  - Aug 2008 [proprietary deliverable]
  - May 2009 (estimate) via website [public version]
- Phase 2 Status Updates @ IW’09, etc.
- Misc. reports/updates/publications @ various venues
  - OMG meetings, NDIA, society & vendor conferences, ...
Contents

• Phase 1 Overview and Results
  – From August, 2007 to August, 2008

• Phase 2 Progress
  – From August, 2008 to August, 2009
MBSE Challenge Team Objectives
Phase 2: 2008–2009

Overall Objectives

• Refine & extend beyond Phase 1 capabilities for modeling & simulation interoperability (MSI)

• Phase 2 Scope [new aspects]
  – *Domains*: Primary: Mechatronics (expanded excavator testbed)
    Secondary: Others to demo reusability
  – *Capabilities*: Methodologies, tools, requirements, and practical applications (MI M v2, …)
  – *MSI subset*: Connecting system specification & design models with multiple engineering analysis
  – *Deployment*: Productionizing techniques & tools to enable ubiquitous practice

• Advance & demo how SysML facilitates effective MSI
MBSE Challenge Team Objectives
Phase 2: 2008–2009

Specific Objectives

1. Extend modeling & simulation interoperability method: MIM 2.0
   1. Generalizations: graph transformations, variable topology, reusability, parametrics 2.x, trade study support, inconsistency mgt., E/MBOM extensions, method workflow, ...
   2. Specializations: software, closed-loop control, electronics, ...
   3. Interfaces to new tools: Matlab/Simulink, ECAD, Arena, ...

2. Refine SysML and tool requirements to support MIM 2.0
   1. Provide feedback to vendors and OMG SysML 1.2/2.x task forces

3. Demonstrate extensions in updated testbed

4. Define deployment plan and initiate execution

5. Refine roadmap beyond Phase 2
Knowledge Composition Methodology for Effective Analysis Problem Formulation in Simulation-based Design

Manas Bajaj
manas.bajaj@gatech.edu
Georgia Tech

Engineering Information Systems Lab
www.eislab.gatech.edu
Systems Realization Lab
www.srl.gatech.edu

Addressing challenges of variable topology and analysis intent...
Abstract

In simulation-based design, a key challenge is to formulate and solve analysis problems efficiently to evaluate a large variety of design alternatives. The solution of analysis problems has benefited from advancements in commercial off-the-shelf math solvers and computational capabilities. However, the formulation of analysis problems is often a costly and laborious process. Traditional simulation templates used for representing analysis problems are typically brittle with respect to variations in artifact topology and the idealization decisions taken by analysts. These templates often require manual updates and “re-wiring” of the analysis knowledge embodied in them. This makes the use of traditional simulation templates ineffective for multi-disciplinary design and optimization problems.

Based on these issues, this dissertation defines a special class of problems known as variable topology multi-body (VTMB) problems that characterizes the types of variations seen in design-analysis interoperability. This research thus primarily answers the following question:

**How can we improve the effectiveness of the analysis problem formulation process for VTMB problems?**

The knowledge composition methodology (KCM) presented in this dissertation answers this question by addressing the following research gaps: (1) the lack of formalization of the knowledge used by analysts in formulating simulation templates, and (2) the inability to leverage this knowledge to define model composition methods for formulating simulation templates. KCM overcomes these gaps by providing: (1) formal representation of analysis knowledge as modular, reusable, analyst-intelligible building blocks, (2) graph transformation-based methods to automatically compose simulation templates from these building blocks based on analyst idealization decisions, and (3) meta-models for representing advanced simulation templates—VTMB design models, analysis models, and the idealization relationships between them.

Applications of the KCM to thermo-mechanical analysis of multi-stratum printed wiring boards and multi-component chip packages demonstrate its effectiveness—handling VTMB and idealization variations, and enhanced computational efficiency (from several hours in existing methods to few minutes). In addition to enhancing the effectiveness of analysis problem formulation, the KCM is envisioned to provide a foundational approach to model formulation for generalized variable topology problems.

Main sponsor: NIST (Ray, Sriram, Fenves, Brady, et al.)
Primary Capabilities

◆ Variations in system design topology
◆ Variations in idealization intent
◆ Efficiency
  – 90%+ faster
  – Reusable analysis building blocks (ABBs)
  – Automated composition from building blocks
    » Formal approach based on graph transformations
  – Meta-models for design and behavior model abstractions
  – Libraries of ABBs, transformation patterns, and rules
SysML Parametrics
Flattened Graphs
Examples

SysML Parametrics Flattened Graphs

1. Spring systems (with animation)
2. Road scanning system using LittleEye UAVs
3. Flap linkage mechanical design
4. Multi-year business financial model

For further information on these examples, see backup slide below entitled “SysML Parametrics—Suggested Starting Points” for these references:
- Examples 1 and 3: Peak et al. 2007 (ISO7 Parts 1 and 2)
- Examples 2 and 4: Zwemer and Bajaj 2008 (Frontiers Workshop)
SysML Parametrics Graph Visualization
[in collaboration with InterCAX—A. Scott Fall 2008 internship]

• Flattened graph [aka COB constraint graph]
  – Flattened graph \( \cong \) graph among value types
  – Block encapsulation not shown

• Purpose
  – Alternative way to understand / interact with a given model
    • Primitive connections/relationships, structure, complexity, ...
  – Enables visual/intuitive model comparisons
  – Possible additional SysML view of models

• Status
  – Prototype plugin that leverages ygraph toolkit
  – Auto-generates flattened graph from MagicDraw
  – Construction animation and static final view
SysML and Mobile Robotic Systems: A Research Testbed and Educational Platform

Status Update: 2009-Feb-17

Georgia Tech Modeling & Simulation Lab – www.msl.gatech.edu
Russell Peak (PI), Bennett Wilson, Brian Aikens, Michael Qin

→ Background & Objectives
  • Operational Control Using SysML Activities
    – Demonstration
  • Status & Next Steps
Institute for Personal Robots in Education (IPRE) — http://www.roboteducation.org/

Robots Available!
The robots we have developed for use in our curricula are now available for sale. The entire robot kit (robot + Fluke add on board) is available for $149.95 at Georgia Robotics.

Check out the list of schools who are testing the IPRE Robot.

Summer Workshops
We're offering two IPRE Summer Undergraduate Faculty Enhancement Workshops (K-12 teachers also invited). In these workshops we will introduce you to a fresh approach to CS1 that uses robots. Attendees will be given a personal robot kit and curricular materials to take home. The curriculum is Python-based.

June 11, 12, 13: Bryn Mawr College.
July 7, 8, 9: Georgia Institute of Technology.

Enter a reservation request.

$10K Awards
The Institute for Personal Robots in Education (IPRE) is pleased to announce the availability of awards to faculty interested in testing robots in education. The deadline for applications is April 22, 2008. Read more here.

Our Mission
The Institute for Personal Robots in Education (IPRE) applies and evaluates robots as a context for computer science education. IPRE is a joint effort between Georgia Tech and Bryn Mawr College sponsored by Microsoft Research. At Georgia Tech, IPRE is associated with Robotics and the College of Computing. At Bryn Mawr College, IPRE is associated with the Computer Science Department.

Our old site is here in case you need to find something that isn't on the new site yet.
Background

• Leveraging Institute for Personal Robots in Education (IPRE) — http://www.roboteducation.org/
  – Multi-university/corporation educational environment
  – Ex. Used in intro comp sci course @ GIT (CS1301)

• Key elements
  – Mobile robots: IPRE Scribbler, Roomba, SRV-1
    • Sensors, cameras, Bluetooth, firmware, PCB ECAD, ...
  – Mobile robotics s/w platform: Myro (Python)
    • Primitive operations ... image processing, intro ~AI, ...
  – Domain context
    • Multi-unit systems, command & control, reusability, ...

• Low-cost and open (non-proprietary)
Objectives—Big Picture

• Research & demonstration testbed
  • Achieve Phase 2 objectives (INCOSE MBSE MSI Team)
    – System run-time operation aided by SysML
    – Embedded software / firmware
      • Hardware-software relations, real-time factors, ...
    – Executable SysML across multiple constructs
      • Activities, parametrics, state machines ...
    – Misc: instance levels, versioning/config mgt.

• SysML education platform
  – Usage in hands-on courses
    (industry short courses, university courses, ...)
  – Model it and run it!
SysML and Mobile Robotic Systems: A Research Testbed and Educational Platform

Status Update: 2009-Feb-17

Georgia Tech Modeling & Simulation Lab – www.msl.gatech.edu
Russell Peak (PI), Bennett Wilson, Brian Aikens, Michael Qin

• Background & Objectives
• Operational Control Using SysML Activities – Demonstration
• Status & Next Steps
from myro import *
initialize("com29")
forward(1, 1)
turnRight(1, .4)
forward(1, 1)
turnRight(1, .4)
forward(1, 1)
turnRight(1, .4)
forward(1, 1)
stop()
from myro import *
initialize("com29")
senses()
beep(1, 440)
forward(1, 1)
turnRight(1, .4)
forward(1, 1)
beep(1, 440)
turnRight(1, .4)
forward(1, 1)
turnRight(1, .4)
forward(1, 1)
stop()
Exercise 456 on Moon: Rover - Unmanned
Mission: Pick up 10 kg of rocks at two specified locations

WP 1
- Time: 000 minutes
- Task 1: Travel WP 1 to WP 2
- Power = 500 units
- Power Rate: 1 unit per minute (traveling and at stops)
- Rover Weight = 100 kg
- Report all data attributes
- Heading: 120 degrees for
- Time: 30 minutes

WP 2
- Stop at Target
- Task 2: Pick up rocks
- 10 kilos
- 10 minutes
- Task 3: WP 2 to WP 3
- Heading: 060 degrees
- Time: 60 minutes
- Report All Data attributes

WP 3
- Task 4: WP 3 to WP 4
- Report all data attributes
- Heading: 300 degrees
- Time: 40 minutes

WP 4
- Stop at Target
- Task X: Pick up rocks
- 10 kilos
- 10 minutes
- Report all data attributes
- Task X WP 4 to WP 5
- Heading: 200 degrees
- Time: 50 minutes

WP 5
- Stop
- Report all data

Report All Data attributes
Contents

• Problem Description
  – Characteristics of Mechatronic Systems
  – Challenge Team Objectives

• Technical Approach
  – Techniques and Testbeds

• Deliverables & Outcomes

• Summary & Collaboration Approach
SE Practices for Describing Systems

Past / Now

- Specifications
- Interface requirements
- System design
- Analysis & trade-off
- Test plans

Now / Future

Moving from Document-centric to Model-centric
What you can do with a SysML model ...

- Describe requirements, system structure, & allocations
- Generate and/or link to simulations & verify requirements
- Support system trade studies
- Link to domain models & analyses: S/W, M/ECAD, ...
- I.e., do the Vee and more ... (e.g., support system operation)
## Benefits of SysML-based Approach

### Primary Impacts

<table>
<thead>
<tr>
<th>Enabling Capabilities</th>
<th>Reduced Time</th>
<th>Reduced Cost</th>
<th>Reduced Risk</th>
<th>Increased Understanding</th>
<th>Increased Corporate Memory</th>
<th>Increased Artifact Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increased Knowledge Capture &amp; Completeness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increased Modularity &amp; Reusability</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increased Traceability</td>
<td></td>
<td></td>
<td></td>
<td>■</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reduced Manual Re-Creation</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increased Automation</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reduced Modeling Effort</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increased Analysis Intensity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■</td>
<td>■</td>
</tr>
</tbody>
</table>

### Precision Knowledge for the Model-Based Enterprise

*Georgia Institute of Technology*
MBSE Challenge
Model Interoperability Team
Open “Call for Participation”

• Systems engineering drivers in commercial settings
  – Increased system complexity
  – Cross-disciplinary communication/coordination

• Enhancement possibilities based on interest
  – Sponsoring other demonstrations and testbeds
  – Developing shared models and libraries
  – etc.

• Primary contacts
  – Russell Peak [Russell.Peak@gatech.edu]
  – Sandy Friedenthal [sanford.friedenthal@lmco.com]
  – Roger Burkhart [BurkhartRogerM@JohnDeere.com]
Additional Resources
SysML Parametrics—Suggested Starting Points

Introductory Papers/Tutorials

  http://eislab.gatech.edu/pubs/conferences/2007-incose-is-1-peak-primer/
  http://eislab.gatech.edu/pubs/conferences/2007-incose-is-2-peak-diversity/

Example Applications

  http://eislab.gatech.edu/pubs/seminars-etc/2008-06-incose-is-mbse-mechatronics-msi-peak/

Commercial Tools and Other Examples/Tutorials

  http://www.pslm.gatech.edu/events/frontiers/

See slides below for additional references and resources.

[1] Full disclosure: InterCAX LLC is a spin-off company originally created to commercialize technology from RS Peak’s GIT group. GIT has licensed technology to InterCAX and has an equity stake in the company. RS Peak is one of several business partners in InterCAX. Commercialization of the SysML/composable object aspects is being fostered by the GIT VentureLab incubator program (www.venturelab.gatech.edu) via an InterCAX VentureLab project initiated October 2007.
MBX/SysML-Related Efforts at Georgia Tech

• SysML Focus Area web page
  – http://www.pslm.gatech.edu/topics/sysml/
  – Includes links to publications, applications, projects, examples, courses, commercialization, etc.
  – Frontiers 2008 workshop on MBSE/MBX, SysML, ...

• Selected projects
  – Deere: System dynamics (fluid power, ...)
  – Lockheed: System design & analysis integration
  – NASA: Enabling technology (SysML, ...)
  – NIST: Design-analysis interoperability (DAI)
  – TRW Automotive: DAI/FEA (steering wheel systems ... )
Selected GIT MBX/SysML-Related Publications

Some references are available online at http://www.pslm.gatech.edu/topics/sysml/. See additional slides for selected abstracts.

- Kwon, Ky Sang, and Leon F. McGinnis, “SysML-based Simulation Framework for Semiconductor Manufacturing,” IEEE CASE Conference, Scottsdale, AZ, September 22-25, 2007. [Presents some technical details on the use of SysML to create formal generic models (user libraries) of fab structure, and how these formal models can be combined with currently available data sources to automatically generate simulation models.]
- Huang, Edward, Ramamurthy, Randeep, and Leon F. McGinnis, "System and Simulation Modeling Using SysML," 2007 Winter Simulation Conference, Washington, DC. [Presents some technical details on the use of SysML to create formal generic models (user libraries) of fab structure, and how these formal models can be combined with currently available data sources to automatically generate simulation models.]

- T.A. Johnson, C.J.J. Paredis, and R. Burkhart "Integrating Models and Simulations of Continuous Dynamics into SysML," in Proceedings of the 6th International Modelica Conference, March 3-4, 2008. [Describes how continuous dynamic models and simulations can be used in the context of engineering systems design within SysML. The design of a car suspension modeled as a mass-spring-damper system is used as an illustration.]
Abstract
This presentation overviews work-in-progress experiences and lessons learned from an excavator testbed that interconnects simulation models with associated diverse system models, design models, and manufacturing models. The goal is to enable advanced model-based systems engineering (MBSE) in particular and model-based X\(^1\) (MBX) in general. Our method employs SysML as the primary technology to achieve multi-level multi-fidelity interoperability, while at the same time leveraging conventional modeling & simulation tools including mechanical CAD, factory CAD, spreadsheets, math solvers, finite element analysis (FEA), discrete event solvers, and optimization tools. This work is currently sponsored by several organizations (including Deere and Lockheed) and is part of the Mechatronics & Interoperability Team in the INCOSE MBSE Challenge.

Citation

[1] The X in MBX includes engineering (MBE), manufacturing (MBM), and potentially other scopes and contexts such as model-based enterprises (MBE).
Part 1: A Parametrics Primer

OMG SysML™ is a modeling language for specifying, analyzing, designing, and verifying complex systems. It is a general-purpose graphical modeling language with computer-sensible semantics. This Part 1 paper and its Part 2 companion show how SysML supports simulation-based design (SBD) via tutorial-like examples. Our target audience is end users wanting to learn about SysML parametrics in general and its applications to engineering design and analysis in particular. We include background on the development of SysML parametrics that may also be useful for other stakeholders (e.g., vendors and researchers).

In Part 1 we walk through models of simple objects that progressively introduce SysML parametrics concepts. To enhance understanding by comparison and contrast, we present corresponding models based on composable objects (COBs). The COB knowledge representation has provided a conceptual foundation for SysML parametrics, including executability and validation. We end with sample analysis building blocks (ABBs) from mechanics of materials showing how SysML captures engineering knowledge in a reusable form. Part 2 employs these ABBs in a high diversity mechanical example that integrates computer-aided design and engineering analysis (CAD/CAE).

The object and constraint graph concepts embodied in SysML parametrics and COBs provide modular analysis capabilities based on multi-directional constraints. These concepts and capabilities provide a semantically rich way to organize and reuse the complex relations and properties that characterize SBD models. Representing relations as non-causal constraints, which generally accept any valid combination of inputs and outputs, enhances modeling flexibility and expressiveness. We envision SysML becoming a unifying representation of domain-specific engineering analysis models that include fine-grain associativity with other domain- and system-level models, ultimately providing fundamental capabilities for next-generation systems lifecycle management.

Part 2: Celebrating Diversity by Example

These two companion papers present foundational principles of parametrics in OMG SysML™ and their application to simulation-based design. Parametrics capabilities have been included in SysML to support integrating engineering analysis with system requirements, behavior, and structure models. This Part 2 paper walks through SysML models for a benchmark tutorial on analysis templates utilizing an airframe system component called a flap linkage. This example highlights how engineering analysis models, such as stress models, are captured in SysML, and then executed by external tools including math solvers and finite element analysis solvers.

We summarize the multi-representation architecture (MRA) method and how its simulation knowledge patterns support computing environments having a diversity of analysis fidelities, physical behaviors, solution methods, and CAD/CAE tools. SysML and composable object (COB) techniques described in Part 1 together provide the MRA with graphical modeling languages, executable parametrics, and reusable, modular, multi-directional capabilities.

We also demonstrate additional SysML modeling concepts, including packages, building block libraries, and requirements-verification-simulation interrelationships. Results indicate that SysML offers significant promise as a unifying language for a variety of models—from top-level system models to discipline-specific leaf-level models.

Citation


Part 1: A Parametrics Primer

http://eislab.gatech.edu/pubs/conferences/2007-incose-is-1-peak-primer/

Part 2: Celebrating Diversity by Example

http://eislab.gatech.edu/pubs/conferences/2007-incose-is-2-peak-diversity/
Composable Objects (COB) Requirements & Objectives

Abstract
This document formulates a vision for advanced collaborative engineering environments (CEEs) to aid in the design, simulation and configuration management of complex engineering systems. Based on inputs from experienced Systems Engineers and technologists from various industries and government agencies, it identifies the current major challenges and pain points of Collaborative Engineering. Each of these challenges and pain points are mapped into desired capabilities of an envisioned CEE System that will address them.

Next, we present a CEE methodology that embodies these capabilities. We overview work done to date by GIT on the composable object (COB) knowledge representation as a basis for next-generation CEE systems. This methodology leverages the multi-representation architecture (MRA) for simulation templates, the user-oriented SysML standard for system modeling, and standards like STEP AP233 (ISO 10303-233) for enhanced interoperability. Finally, we present COB representation requirements in the context of this CEE methodology. In this current project and subsequent phases we are striving to fulfill these requirements as we develop next-generation COB capabilities.

Citation

Associated Project
Abstract
SysML holds the promise of leveraging generic templates and processes across design and simulation. Russell Peak joins us to give an update on the latest efforts at Georgia Tech to apply this approach in various domains, including specific examples with a top-tier automotive supplier. Learn how you too may join this project and implement a similar effort within your own company to enhance modularity and reusability through a unified method that links diverse models. Russell will also highlight SysML’s parametrics capabilities and usage for physics-based analysis, including integrated CAD-CAE and simulation-based requirements verification. Go to [www.omgsysml.org](http://www.omgsysml.org) for background on SysML—a graphical modeling language based on UML2 for specifying, designing, analyzing, and verifying complex systems.

Speaker Biosketch
Russell S. Peak focuses on knowledge representations that enable complex system interoperability and simulation automation. He originated composable objects (COBs), the multi-representation architecture (MRA) for CAD-CAE interoperability, and context-based analysis models (CBAMs)—a simulation template knowledge pattern that explicitly captures design-analysis associativity. This work has provided the conceptual foundation for SysML parametrics and its validation.

He teaches this and related material, and is principal investigator on numerous research projects with sponsors including Boeing, DoD, IBM, NASA, NIST, Rockwell Collins, Shinko Electric, and TRW Automotive. Dr. Peak joined the GIT research faculty in 1996 to create and lead a design-analysis interoperability thrust area. Prior experience includes business phone design at Bell Laboratories and design-analysis integration exploration as a Visiting Researcher at Hitachi in Japan.

Citation