Model Based Systems Engineering using SysML

4th MODPROD Workshop on Model-Based Product Development

February 10, 2010

Sanford Friedenthal
Lockheed Martin
sanford.friedenthal@lmco.com
Topics

- Model-based Systems Engineering (MBSE) Motivation and Scope
- System Modeling Using SysML
- System Model as an Integration Framework
- SysML-Modelica Integration
- SysML Status
- Summary
MBSE Motivation and Scope
SE Practices for Describing Systems

**Past**

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

**Future**

Moving from Document centric to Model centric
Model-based Systems Engineering (MBSE)

- Formalizes the practice of systems development through use of models
- Broad in scope
  - Integrates with multiple modeling domains across life cycle from system of systems to component
- Results in quality/productivity improvements & lower risk
  - Rigor and precision
  - Communications among system/project stakeholders
  - Management of complexity

Life Cycle Support

© Copyright Lockheed Martin Corporation All Rights Reserved
System Description

- **Document-Based System Engineering:**

  Where is truth?

  - Document 1
    - A < B
  - Document 2
    - A = B
  - Document 3
    - A > B

  Inconsistencies within and among documents

- **Model-Based System Engineering:**

  Model enforces consistency

  Model Repository

© Copyright Lockheed Martin Corporation All Rights Reserved
System Modeling Using SysML
System Modeling

Requirements

Functional/Behavioral Model
- Start
- Shift
- Accelerate
- Brake

Performance Model
- Control
- Input
- Power
- Equations
- Vehicle
- Dynamics

System Model

Structural/Component Model
- Engine
- Transmission
- Transaxle

Other Engineering Analysis Models
- Mass
- Productivity
- Structural
- Safety
- Cost
- Model

Integrated System Model Must Address Multiple Aspects of a System
What is SysML?

- A graphical modeling language in response to the UML for Systems Engineering RFP developed by the OMG, INCOSE, and AP233
  - a UML Profile that represents a subset of UML 2 with extensions

- Supports the specification, analysis, design, verification, and validation of systems that include hardware, software, data, personnel, procedures, and facilities

- Supports model and data interchange via XML Metadata Interchange (XMI®) and the evolving AP233 standard (in-process)

SysML is Critical Enabler for MBSE
SysML Diagram Taxonomy

SysML Diagram

Behavior Diagram

Requirement Diagram

Structure Diagram

Activity Diagram

Sequence Diagram

State Machine Diagram

Use Case Diagram

Block Definition Diagram

Internal Block Diagram

Package Diagram

Same as UML 2

Modified from UML 2

New diagram type

Modified from UML 2

New diagram type
**4 Pillars of SysML – ABS Example**

1. **Structure**
   - **Library: Electronic Processor**
   - **Library: Anti-Lock Controller**
   - **Traction Detector**
   - **Brake Modulator**

2. **Behavior**
   - **Sequence Diagram**
     - `d1:Traction Detector`
     - `m1:Brake Modulator`
     - `detTrkLos()`
     - `modBrkFrc()`
     - `sendSignal()`
     - `modBrkFrc(traction_signal:boolean)`
     - `sendAck()`

3. **Requirements**
   - **Vehicle System Specification**
     - **Stopping Distance**
       - Id = "10.2"
       - Text = "The vehicle shall stop from 60 miles per hour within 150 ft on a clean dry surface."
   - **Anti-Lock Performance**

4. **Parametrics**
   - **Straight Line Vehicle Dynamics**
     - **Braking Force Equation**
       - \( f = \frac{f_t f_0}{(1 + t)} \)
   - **Acceleration Equation**
     - \( a = \frac{m}{sec^2} \)
   - **Distance Equation**
     - \( v = dx/dt \)
     - \( x: m \)
     - \( t: sec \)
   - **Velocity Equation**
     - \( a = d^2v/dt^2 \)
SysML Model Used to Elaborate System and Component Requirements
System Model as an Integration Framework
MBSE Must Integrate across Modeling Domains

- Ops/Mission Analysis
- Logistics Support
- Manufacturing
- Integration & Test
- Performance Simulation
- Engineering Analysis
- System Design
- Algorithm Development
- Software Design
- Hardware Design
- Human System Integration

© Copyright Lockheed Martin Corporation All Rights Reserved
Using System Architecture Model as an Integration Framework

- **Analysis Models**: $U(s)$, $G(s)$, $\int$

- **System Architecture Model**

- **Hardware Models**

- **Software Models**

- **Verification Models**

- **Req'ts Allocation & Design Integration**

- **Black Box System Specification**
Using the System Architecture Model to Flowdown Requirements

System-of-System Level
- 1st Level Of Decompositions
- How Our System Contributes to the Overall Mission

System Level
- Derives Subsystems
- Allocates Requirements to Subsystems

Element Level
- Derives Hardware and Software Components
- Allocates Requirements to Components

Component Design & Implementation Level

Trade Studies, Simulation, Specification Reviews, etc.

(a) Mission Concept of Operations
(b) System Level
(c) Element Level
(d) Component Design & Implementation Level

(from John Watson/LMC SysML Info Days presentation)
System Decomposition Process using SysML:

1. **Analyze System Level Requirements**
2. **Analyze System Services**
3. **Identify the Subsystem**
4. **Analyze Subsystem Collaboration to Satisfy the System Services**
5. **Incorporate Additional Analysis as Needed**
6. **Derive and Allocate Requirements to Subsystem**
   - **Continue?**
     - Yes
     - No: **Complete Subsystem Specs**
   - **Trade Studies, R&D, Simulation, Specification Reviews, etc.**
     - **The Subsystem shall .... Derived Requirements**

*(from John Watson/LMC SysML Info Days presentation)*

© Copyright Lockheed Martin Corporation All Rights Reserved
System Architecture Model to Support Tradeoff Analysis

Cost
Reliability
Performance

Optimization

Effectiveness
\[ E = \text{Sum} [w_1 \times P + w_2 \times R + w_3 \times C] \]

Criteria | Weight | Alt 1 | Alt 2 | Alt 3 |
--- | --- | --- | --- | --- |
Performance | 0.5 | 7 | 5 | 5 |
Reliability | 0.2 | 4 | 6 | 5 |
Cost | 0.3 | 3 | 5 | 8 |
Effectiveness | 5.2 | 4.2 | **5.9** |
SysML-Modelica Integration
Integrating SysML with Simulation

GIT* Project

Modelica
Lexical Representation
(auto-generated from SysML)

[Johnson, 2008 - Masters Thesis]

* Georgia Institute of Technology
SysML-Modelica Working Group

- Initiated December 2008 at OMG SE DSIG Meeting

- Objective:
  - Leverage the strengths of both SysML and Modelica by integrating them to create a more expressive and formal MBSE language.
  - Define a formal Transformation Specification: a SysML4Modelica profile and a mapping between the profile and Modelica

- Members from both SysML and Modelica Community
  - Yves Bernard
  - Roger Burkhart
  - Hans-Peter de Koning
  - Sandy Friedenthal
  - Peter Fritzson
  - Nerijus Jankevicius
  - Alek Kerzhner
  - Chris Paredis (Chair)
  - Nicolas Rouquette
  - Wladimir Schamai

- Plan
  - Submit Transformation Spec to OMG at June 2010 meeting

- Working Group Site
SysML-Modelica Transformation

Transformation Specification
- Part I — Introduction
- Part II — SysML4Modelica profile
- Part III — Modelica meta-model
- Part IV — SysML-Modelica mapping
- Annex A – Robotic Sample Problem
Robot Context Diagram
Actuator Internal Block Diagram

Diagram showing the internal block diagram of an actuator, with components such as Sensor (as), Controller (c), Motor Assembly (ma), Gear (gb), and connections like cmd, cmd2c, and RotationalEnergyFlow.
Computing Overall Robot Effectiveness

```
par { Block } RobotAnalysisContext[ Overall Effectiveness ]

<<constraint>>
wm : Weight Model
  totalMass : kg
  mass : kg

<<constraint>>
pm : Power Model
  averagePower : W
  power : W

<<constraint>>
rm : Reliability Model
  reliability : Real
  reliability : Real

<<constraint>>
cm : Cost Model
  cost : us$
  cost : us$

<<constraint>>
tm : Trajectory Error Model
  maxPositionError : m
  trajectory error : m

<<objectiveFunction>>
obj : Objective Function

utility : Real

effectiveness : Real
```


Trajectory Error Analysis

```
par [Block] RobotAnalysisContext [ TrajectoryErrorAnalysis ]
```

```
rd : Robot Domain

<block>
  rob : Robot
    positionAccuracy : m
    axis1.positionGain : Real
    axis2.positionGain : Real
    axis3.positionGain : Real
    axis4.positionGain : Real
    axis5.positionGain : Real
    axis6.positionGain : Real

<Constraint>
  tm : Trajectory Error Model
    posGain1 : Real
    posGain2 : Real
    posGain3 : Real
    posGain4 : Real
    posGain5 : Real
    posGain6 : Real
    maxPositionError : m
```
Creating the SysML4Modelica Model
SysML4Modelica Analytical Model
SysML Status
OMG SysML™ Status

- **Specification**
  - OMG Beta Specification in May ’06
  - Available Specification v1.0 in Sept ’07
  - Available Specification v1.1 in Nov ‘08
  - Revision task force for v1.2 in Dec ‘09
  - Revision task force for v1.3 in process

- **Adoption**
  - Multiple vendor implementations available
  - Increasing number of early adopters across industry
  - Being introduced into academia
  - Books available (4)
  - SysML Certification being developed
  - DISR Emerging Standard (March 26, 2009)

- **Information can be found on the OMG SysML Website at** [http://www.omgsysml.org/](http://www.omgsysml.org/)
Summary

- MBSE is a key practice to advance complex systems development
- Standards such as SysML and Modelica are critical enablers of MBSE
- Multiple tool vendors implementing SysML
- System architecture model and standards based approach facilitate integration across modeling domains
- SysML-Modelica Integration provides synergistic capability for system design and analysis
- Growing interest and application of MBSE