Systems Modeling Language (SysML) Overview

SysML Merge Team (SMT)
March 16, 2006
**Draft**

These slides are in process of development by the SMT team and are subject to change as the SysML specification is finalized. Please check for current SysML status at syseng.omg.org.

SMT Chair: Sanford Friedenthal
sanford.friedenthal@lmco.com
Caveat

This material is based on Draft V0.99 of the SysML specification and is still subject to change prior to adoption.
Topics

- Motivation & Background
- Diagram Overview
- SysML Modeling as Part of an SE Process
- SysML in a Standards Framework
- Transitioning to SysML
- Summary
Motivation & Background
SE Practices for Describing Systems

**Past**

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

**Future**

Moving from Document centric to Model centric

- Airplane
- ATC
- Pilot
- Past
- Future

Past Future

Executed cmd

Initiate Taxi

Report Status

Initiate power-up

Power-up

Direct taxiway

Authorize

Request to proceed
Model Based Engineering Benefits

- Improved communications
- Improved design quality
  - Reduced errors and ambiguity
  - More complete representation
- Reduced maintenance costs
- Enhanced knowledge capture
System-of-Systems

Modeling Needed to Manage System Complexity
Stakeholders Involved in System Acquisition

Customers

Developers/Integrators

Regulators

Project Managers

Testers

Vendors

Modeling Needed to Improve Communications
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 XMI and the evolving AP233 standard (in-process)

SysML is Critical Enabler for Model Driven SE
UML/SysML Status

UML V2.0
- Updated version of UML that offers significant capability for systems engineering over previous versions
- Finalized in 2005 (formal/05-07-04)

UML for Systems Engineering (SE) RFP
- Established the requirements for a system modeling language
- Issued by the OMG in March 2003

SysML
- Industry Response to the UML for SE RFP
- Addresses most of the requirements in the RFP
- Current version 0.99 and expected to begin adoption in April ‘06
- Being implemented by 5 vendors
SysML Merge Team (SMT)

Industry & Government
- American Systems, BAE SYSTEMS, Boeing, Deere & Company, EADS-Astrium, Eurostep, Lockheed Martin, Motorola, NIST, Northrop Grumman, oose.de, Raytheon, THALES

Vendors

Academia
- Georgia Institute of Technology

Liaison Organizations
- INCOSE, AP233 WG
Diagram Overview
Relationship Between SysML and UML

- UML 2
- SysML
- UML reused by SysML
- SysML extensions to UML
- UML not required by SysML (UML - UML4SysML)
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**

Legend:
- ![Same as UML 2](image)
- ![Modified from UML 2](image)
- ![New diagram type](image)
4 Pillars of SysML – ABS Example

Structure

- **bdd** [block] Anti-LockController [Block Definition Diagram]
  - **«block»** Library::Electronic Processor
  - **«block»** BrakingSystem::Anti-Lock Controller
  - **«block»** Library::Electro-Hydraulic Valve
  - **d1**: Traction Detector
  - **m1**: Brake Modulator

- **ibd** [block] Anti-LockController [Internal Block Diagram]
  - **d1**: Traction Detector
  - **m1**: Brake Modulator

Behavior

- **sd** ABS_ActivationSequence [Sequence Diagram]
  - **act** PreventLockup [Activity Diagram]
  - **stm** TireTraction [State Diagram]
  - **par** [constraintBlock] StraightLineVehicleDynamics

Requirements

- **req** [package] VehicleSpecifications [Requirements Diagram - Braking Requirements]
  - **Vehicle System Specification**
  - **Braking Subsystem Specification**
  - **«requirement»** StoppingDistance
    - **id**: 102
    - **text**: The vehicle shall stop from 60 mph within 150 ft on a clean dry surface.
  - **«requirement»** Anti-LockPerformance
    - **id**: 337
    - **text**: Braking subsystem shall prevent wheel lockup under all braking conditions.

Parametrics

- **constraint** BrakingForce Equation
  \[ f = (tf*bf)*(1-tl) \]
- **constraint** Acceleration Equation
  \[ F = ma \]
- **constraint** DistanceEquation
  \[ v = dx/dt \]
- **constraint** VelocityEquation
  \[ a = dv/dt \]
The vehicle shall stop on a clean dry surface from 60 mph within 150 ft.

Braking subsystem shall prevent wheel lockup under all braking conditions.

Vehicle System Specification

- Requirement: Stopping Distance
  - id: 102
  - text: The vehicle shall stop from 60 mph within 150 ft on a clean dry surface.
  - Verified By

- Requirement: Anti-Lock Performance
  - id: 337
  - text: Braking subsystem shall prevent wheel lockup under all braking conditions.
  - Satisfied By

Vehicle weight

Tire friction

Duty Cycle

Braking force

Velocity Equation

Acceleration Equation

Straight Line Vehicle Dynamics

Modulate Braking Force

Detect Loss Of Traction

Traction Modulator

Traction Detector

Prevent Lock Up

Accelerate

Velocity Equation

Acceleration Equation

Stop the vehicle from 60 mph within 150 ft on a clean dry surface.

Satisfy the anti-lock performance requirement.

Verify the vehicle specifications.

Cross Connecting Model Elements

Structure

Behavior

Parametrics

Requirements
Cross Connecting Model Elements

**Structure**

- ibd [block] Anti-LockController
  - [Internal Block Diagram]
  - satisfies
    - <requirement> Anti-Lock Performance
  - allocatedFrom
    - <activity> DetectLossOfTraction
    - c1:modulator Interface
  - allocatedFrom
    - <connector> traction LossFlow:
      - m1:BrakeModulator
      - allocatedFrom
        - <activity> ModulateBrakingForce
        - Values
          - DutyCycle: Percentage
  - allocatedFrom
    - <activity> DetectLossOfTraction
    - d1:TractionDetector
    - allocatedFrom
      - <connector> traction LossFlow:
        - allocatedFrom
          - <activity> DetectLossOfTraction

**Behavior**

- act PreventLockup [Swimlane Diagram]
  - »allocate«
    - :TractionDetector
    - :BrakeModulator
  - traction LossFlow:
    - traction LossFlow:
      - »allocate«
        - :TractionDetector
        - :BrakeModulator
      - »allocate «
        - :modulatorInterface

**Parametrics**

- par [constraintBlock] StraightLineVehicleDynamics
  - «value»
    - Tire friction
    - DutyCycle
    - Braking force
    - Vehicle weight
  - «constraint»
    - BrakingForce Equation
      - \[ f = (tf*bf)*(1-tl) \]
    - DistanceEquation
      - \[ v = dx/dt \]
    - VelocityEquation
      - \[ a = dv/dt \]
    - Acceleration Equation
      - \[ F = ma \]
SysML Modeling as Part of the SE Process
System Development Process

Integrated Product Development (IPD) is essential to improve communications.

A Recursive V process that can be applied to multiple levels of the system hierarchy.
Typical Systems Modeling Activities

Major SE Development Activities

Analyze Needs

Define System Requirements

Optimize & Evaluate Alternatives

Define Logical Architecture

Validate & Verify System

Synthesize Physical Architecture

Common Subactivities

- Mission use cases/scenarios
- Enterprise model
- System use cases/scenarios
- Elaborated context
- Req’ts diagram
- Logical architecture
- Node diagram
- HW, SW, Data architecture
- Engr Analysis Models
- Trade studies
- Test cases/procedures
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**

Legend:
- Same as UML 2
- Modified from UML 2
- New diagram type
Capturing Requirements in the Model

Artisan Tool
Requirements Breakdown

Containment Used to Model the Content of a Specification
Problem and Rationale

Problem and Rationale can be attached to any Model Element to Capture Issues and Decisions
Operational Use Cases
Activities

- Activities used to specify flow of I/O and control
- SysML extensions to Activities
  - Support for continuous flow modeling
  - Alignment of activities with Enhanced Functional Flow Block Diagram
SysML EFFBD Profile

EFFBD - Enhanced Functional Flow Block Diagram

Aligning SysML with Classical Systems Engineering Techniques
Activity Modeling with Swimlanes

Allocation of parts via swim lanes
Distill Water Activity Diagram (Continuous Flow Modeling)

Representing Distiller Example in SysML Using Continuous Flow Modeling
Interactions

- Sequence diagrams provide representation for message based behavior
  - represents flow of control
- Sequence diagrams provide key mechanisms for representing complex behavior
  - reference sequences
  - control logic
  - lifeline decomposition
Black Box Interaction (Drive)

UML 2 Sequence Diagram More Scalable by Supporting Control Logic and Reference Sequences
Black Box Sequence (StartVehicle)

**Simple Black Box Interaction**

References Lifeline Decomposition
For White Box Interaction
White Box Sequence (StartVehicle)

Decomposition of Black Box Into White Box Interaction
State Machines

- Supports event based behavior (generally asynchronous)
  - Transition with event, guard, action
  - State with entry, exit and do-activity
  - Can include nested sequential or concurrent states
  - Can send/receive signals to communicate between blocks during state transitions, etc
Operational States (Drive)
 SysML Diagram Taxonomy

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
Blocks

- Used to express structure of system
  - Backbone description of system hierarchy
  - Either white- or black-box description
  - Includes internal relationships and connectivity in addition to system decomposition
Power Subsystem IBD

Internal Block Diagram Used to Specify Interconnection Among Parts in Context of Enclosing Block
Parametrics

- Used to express constraints (equations) between value properties
  - Provides support to engineering analysis (e.g. performance, reliability, etc)

- Constraint block captures equations
  - Expression language can be formal (e.g. MathML, OCL ...) or informal
  - Computational engine is defined by applicable analysis tool and not by SysML

- Parametric diagram represents the usage of the constraints in an analysis context
  - Binding of constraint usage to value properties of blocks (e.g. vehicle mass bound to \( F = m \times a \))
Defining Vehicle Dynamics

Defining Reusable Equations for Parametrics
Evaluating Vehicle Dynamics

Using the Equations in a Parametric Diagram to Constrain the Value Properties
Evaluating Measures of Effectiveness

**MOE’s and objective function provide flexible support for trade study analysis that is fully integrated with parametrics**
Allocations

- Provides general relationship to map one model element to another
- Includes specific subclasses of allocation with constraints on their usage
  - Behavioral
  - Structural
  - Flow
- Explicit allocation of activities to swim lanes (e.g. activity partitions)
- Graphical and/or tabular representations
Different Allocation Representations
(Tabular Representation Not Shown)

Allocate Relationship

Explicit Allocation of Activity to Swim Lane

Compartment Notation

Callout Notation
View/Viewpoint

Viewpoint captures stakeholder concerns
- express concerns as requirements
- construction rules describe view which satisfies reqts

View describes system from a particular viewpoint
SysML in a Standards Framework
Systems Engineering Standards & Architecture Frameworks

- **Process Standards**: EIA 632, ISO 15288, IEEE 1220, CMMI
- **Architecture Frameworks**: FEAF, DoDAF, Zachman FW, MDA
- **Modeling Methods**: HP, OOSE, SADT, Other
- **Modeling & Simulation Standards**: IDEF0, UML/SysML, HLA, MATHML
- **Interchange Standards**: MOF/XMI, STEP/AP233, CADM

Implemented By Tools
### ISO/IEC 15288

#### System Life Cycle Processes

**Enterprise Processes**
- 5.3.2 Enterprise Environment Management Process
- 5.3.3 Investment Management Process
- 5.3.4 System Life Cycle Processes Management
- 5.3.5 Quality Management Process
- 5.3.6 Resource Management Process

**Agreement Processes**
- 5.2.2 Acquisition Process
- 5.2.3 Supply Process

**Project Processes**
- 5.4.2 Project Planning Process
- 5.4.3 Project Assessment Process
- 5.4.4 Project Control Process
- 5.4.5 Decision-Making Process
- 5.4.6 Risk Management Process
- 5.4.7 Configuration Management Process
- 5.4.8 Information Management Process

**Technical Processes**
- 5.5.2 Stakeholder Reqs Definition Process
- 5.5.3 Reqs Analysis Process
- 5.5.4 Architectural Design Process
- 5.5.5 Implementation Process
- 5.5.6 Integration Process
- 5.5.7 Verification Process
- 5.5.8 Transition Process
- 5.5.9 Validation Process
- 5.5.10 Operation Process
- 5.5.11 Maintenance Process
- 5.5.12 Disposal Process
Example Standards-based Tool Integration

Systems Modeling Tool

Other SE Engineering Tools

Model/ Data Interchange

Engineering Repository
Participating SysML Tool Vendors

- Artisan
- EmbeddedPlus
- IBM
- I-Logix
- Sparx Systems
- Telelogic
- Vitech
Current initiative underway to develop standard profile for representing DODAF and MODAF products

- Requirements for profile issued Sept 05
- Final submissions expected Dec ‘06

Multiple vendors and users participating

Should leverage SysML
## Tool Support

### Project Management
- IMP/IMS, Process Modeling (ISDM)

### SoS / DoDAF

### System Modeling
- SysML

### Software Modeling
- UML 2.0

### Hardware Modeling
- VHDL, Verilog

### Risk & Opportunity Management

### Verification & Validation

### Measurement

### Specialty Engineering

|-------|-------------------------|-------------------------|----------------------------------|-------------|-----------------|-------|------------------|----------------------|-----------------------------|--------------------------|-------------|----------------------|
Summary

- SysML sponsored by INCOSE/OMG with broad industry and vendor participation
- SysML provides general purpose modeling language to support specification, analysis, design and verification of complex systems
  - Subset of UML 2 with extensions
- Multiple vendor implementations announced
- Adoption expected to begin in April 2006
- Standards based modeling approach for SE expected to improve communications, tool interoperability, and design quality
References

- UML for SE RFP
  - OMG doc# ad/03-03-41
- SysML specification v0.99
  - http://syseng.omg.org/SysML.htm
- UML 2 Superstructure
  - OMG doc# formal/05-07-04
- UML 2 Infrastructure
  - OMG doc# ptc/04-10-14