Systems Modeling Language™ (SysML™) Overview

SysML Partners
(www.SysML-Partners.org)

October 24, 2003
Overview

- Introduction
- SysML Requirements & Language Architecture
- SysML Diagram Overview
- AP-233 Alignment
- Plan Forward
- Summary
Introduction
Model Based Approaches Needed For Developing Complex Systems

Example -- The Modern Battlefield

Battlefield Interdiction
Fire Support
Commander JTF
ERGM
TSTAR
NSFS Msl
JSTARS
TLAM D
Logistics Area
TLAM C Bk IV
Small Unit
Choke Point
Hostile Reserve
Battlefield Isolation
Theater Interdiction
Strategic
63 nmi
200 nmi
500 nmi
1000 nmi
Background

- Systems Engineers need a standard language for analyzing, specifying, designing, and verifying and validating systems
- OMG Systems Engineering Domain Special Interest Group
  - chartered by joint INCOSE-OMG initiative in 2001
  - collaborated with UML2 submission teams
  - drafted UML for SE Request for Proposal, issued by the OMG in March 2003 (ad/03-03-41)
- SysML Partners organized in May 2003 to respond to UML for SE RFP
Why UML for Systems Engineering?

- De-facto standard within the software community
- Robust and extensible language to adapt to SE needs
- OMG Infrastructure
  - Broad international and industry representation
  - Defined adoption process to evolve UML
- Availability of tool vendor and training support
Charter

The SysML Partners are collaborating to define a modeling language for systems engineering applications, called Systems Modeling Language™ (SysML™). SysML will customize UML 2.0 to support the specification, analysis, design, verification and validation of complex systems that includes hardware and software components.
SysML Partners

- **Industry**
  - Astrium, BAE, Boeing, Deere & Company, Lockheed Martin, Motorola, Northrop Grumman, Raytheon, Thales

- **Government**
  - NASA/JPL, NIST, DOD/OSD

- **Tool Vendors**
  - Artisan, Ceira, Gentleware, IBM/Rational, I-Logix, Project Technology, Telelogic, Vitech

- **Liaisons**
  - AP-233, EAST, INCOSE, Rosetta
Requirements Summary
Reference: UML for SE RFP ad/03-03-41

- **Structure**
  - e.g., system hierarchy, interconnection

- **Behavior**
  - e.g., function-based behavior, state-based behavior

- **Properties**
  - e.g., parametric models, continuous time variable attributes

- **Requirements**
  - e.g., requirements hierarchy, traceability

- **Verification**
  - e.g., test cases, verification results
Evaluation Criteria
Reference: UML for SE RFP ad/03-03-41

- Ease of use
- Unambiguous
- Precise
- Complete
- Scalable
- Adaptable to different domains
- Capable of complete model interchange
- Evolvable
- Process and method independent
- Compliant with UML metamodel
- Verifiable
Design Principles

- **Reuse**
  - select the subset of UML 2.0 that is reusable for SE applications
  - add new constructs and diagrams needed for SE
  - UML2++-- (compare Java as C++--)

- **Incremental growth**
  - prevent scope and schedule creep
  - take advantage of SE user feedback as language is rolled out in minor and major revisions

- **Architectural alignment**
  - align with evolving AP-233 SE Data Interchange Standard
Extension Mechanisms

- Three extension mechanisms are being considered
  - profiles (stereotypes)
  - metamodel (metaclasses)
  - model libraries (reusable classes)

- Likely that two out of three will be chosen for implementation flexibility
UML 2 Super: Top Level Packages

- UseCases
- Actions
- Activities
- StateMachines
- CommonBehaviors
- Interactions
- CompositeStructures
- Components
- Deployments
- Classes
- Profiles
- AuxiliaryConstructs

SysML Language Architecture

[Reformat and show in multiple slides]
Major SysML Changes to UML 2

- Class Diagram
  - specialize Aggregation (Assembly aggregation)

- Composite Structure Diagram
  - integration & specialization of InformationItem/InformationFlow
  - include Deployment relationship

- Activity Diagram
  - accommodate need for FFBDs
  - extend Actions and ObjectNodes to support continuous functions and inputs/outputs
  - modify control nodes to support disabling actions
Major SysML Changes to UML 2

- New Diagram Types
  - Requirement Diagram
  - Parametric Diagram

- Diagram Usages (for best practices)
  - Concept Diagram
  - Context Diagram
  - HW/SW Mapping Diagram
Major SysML Changes to UML 2

- Other areas being considered
  - other new diagram types and usages
    - Verification
    - Decision Tree
    - Causal Analysis
  - add communications between objects and states to State Machine diagrams
  - allocation of behavior to structure
Diagram Overview
Changes to UML 2 Diagram Taxonomy include:

- Use existing diagram as-is
- Extensions to existing diagrams
- Some UML 2 diagrams not used and/or subsumed by others
- New SysML Diagrams and usages
SysML Diagram Taxonomy

- **Diagram Description**: `description:text`
- **SysML Diagram**
- **Supports table format**

**Behavior Diagrams**
- **Activity Diagram - M**
- **Sequence Diagram - U**
- **State Machine Diagram - M**
- **Use Case Diagram - U**
- **Class Diagram - U**
- **Structured Class Diagram - M**
- **Collaboration Diagram**

**Structure Diagrams**
- **Requirement Diagram - N**
- **Parametric Diagram - N**

**Interaction Overview Diagram - U**
**Timing Diagram - M**

**U** - UML 2 Diagram used w/o Changes
**M** - Modified UML 2 Diagram with applicable SysML Extensions
**X** - Diagram not explicitly used by SysML
**N** - New SysML Diagram
Diagram Tour

- Extensions to existing diagrams
  - Composite Structure diagram
  - Activity diagram
- UML 2 as-is example
  - Sequence Diagram
- New diagrams
  - Requirements diagram
  - Parametric diagram
class TransportationSystem

abs: ABSBrakeSystem

<table>
<thead>
<tr>
<th>av1: SpeedSensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>pFrequency</td>
</tr>
<tr>
<td>ecu1:Electronic Control Unit</td>
</tr>
<tr>
<td>&lt;&lt;control&gt;&gt; valvePosition:Signal</td>
</tr>
<tr>
<td>mv1: ModulatorValve</td>
</tr>
<tr>
<td>pPositionSignal pHydraulicOut pHydraulicIn</td>
</tr>
<tr>
<td>pHydraulicIn</td>
</tr>
</tbody>
</table>

magneticPlus: Energy

 WheelAssembly O.4

<table>
<thead>
<tr>
<th>wheel1:Wheel</th>
</tr>
</thead>
<tbody>
<tr>
<td>force: HydraulicFluid</td>
</tr>
<tr>
<td>whCy1:WheelCylinder</td>
</tr>
<tr>
<td>loadForce:Mech</td>
</tr>
<tr>
<td>pHydraulic</td>
</tr>
<tr>
<td>pVerticalLoad</td>
</tr>
</tbody>
</table>

tire1:Tire

loadForce:Mech

SysML 24
Monitor Traction

- **<<repeated>> Calculate Traction**
  - Traction Index
  - Loss of Traction

- **<<overwrite>> Calculate Modulation Frequency**
  - Modulation Frequency

- **Optical sensor on wheel**
- **Angular Velocity**
- **Input from Accelerometer**
- **Acceleration**

{stream}
Activity – Example 2/2

SysML

activityDiagram

<<interruptible region>>

Driver

Turn Key On

BrakingSystem

<<ongoing>>

Drive

Brake Pressure

{stream}

Braking

{stream}

Monitor Traction

ABS

Key off

Brake Pressure Modulation Frequency

{stream}

{stream}

{stream}

{stream}
Sequence Diagrams: Combined Fragments – UML V2.0

sd GoHomeInvocation(Time invoc)

:ServiceUser

:Clock

[Now>invoc] InvocationTime

FindLocation

:ServiceBase

TransportSchedule

:ServiceTerminal

loop

alt

[Now>interv+last]

ScheduleIntervalElapsed

FindLocation

TransportSchedule

[pos-lastpos>dist]

GetTransportSchedule

TransportSchedule

TransportSchedule

FetchSchedule

SysML
Requirements - Example

Vehicle Specification

- «requirement» Vehicle performance
  - Reqt # 3.2.1.1
  - Criticality = H
  - Verif Status
  - Text - The system shall ...

- «requirement» Vehicle Comfort
  - Reqt # 3.2.1.2
  - Criticality = H
  - Verif Status
  - Text - The system shall ...

- «Functional Reqt» Provide acceleration
  - «Trace»

- «Functional Reqt» Provide fuel flow

- «Performance Reqt» Acceleration
  - «Trace»
  - «Trace»

- «Physical Reqt» Vehicle Weight

- «Performance Reqt» Horsepower
  - «Trace»

- «System Comp.» Engine
  - «Phys» - Max Horsepower
  - «Phys» - Location

- «System» Vehicle
  - Weight
  - Position
  - Speed
  - Acceleration
  + Accelerate()
  + Decelerate()

Rationale: a = F/M

[DRAFT]

[Needs to be redone using Hruby template so I can correct notation. -- ck]
Parametrics - Example

R1
\[
\text{netVerticalLoad} = \frac{F(\text{max})}{W(\text{load})}
\]

R2
\[
\text{forceStaticFriction} = \frac{F(\text{slide})}{W(\text{load})}
\]

R3
\[
\text{frictionCoefficient}(\text{static}) = \frac{\text{CoF}(\text{RS}, \text{RC})}{W(\text{load})}
\]

R4
\[
\text{forceDynamicFriction} = \frac{\text{CoF}(\text{slide})}{W(\text{load})}
\]

\text{Tire.Force}

\text{Road.surfaceType}

\text{Road.surfaceCondition}

\text{Vehicle.numbWheels}

\text{Vehicle.grossWeight}

\text{Vehicle.sum(Passenger.Weight)}
AP-233 Alignment
AP-233 Overview

- Part of ISO 10303 Standard for a neutral data interchange standard for systems engineering
- Primary AP-233 focus is tool interoperability
- Current AP-233 status
  - Requirements model developed
  - Future modules under development (behavior, structure, management, ...)
- SysML / AP-233 alignment to support interchange between SysML models and other engineering tools
AP233 Data Exchange

SysML Provides Foundation for SE Modeling Tools

- Systems Engineering
- Electrical CAE
- Mechanical CAD
- SW Dev Environment
- Algorithm Design
- Engineering Analysis
- Planning Tools
- Testing Tools

AP-233 NEUTRAL DATA EXCHANGE FORMAT
SysML Project Plan
SysML Milestones

- UML for SE RFP issued – March 28, 2003
- SysML Kickoff – May 6, 2003
- Initial Submission – Oct 27, 2003
- INCOSE Review – January 26, 2004
- Revised Submission – April 5, 2004
- Final Revision – June 1, 2004
- Adoption – Q3 2004
Priorities for SysML V1.0

- Activity diagrams for function based behavior
  - equivalent capability to functional flow block diagrams
  - additional capability to support continuous behaviors
- Composite structure with I/O capability
- Parametric diagram - basic
- Requirement diagram – basic
- Initial alignment with AP-233
- Improved integration of structure and behavior
- Roadmap for SysML V1.1, 2.0
Summary
Potential Synergy Between CCSDS AWG Needs and SysML

CCSDS Need:

“Define common language and representation so that challenges, requirements, and solutions in the area of space data systems can be readily communicated”

- Enterprise
- Connectivity
- Functional
- Information
- Communications

SysML Charter

Define a modeling language for systems engineering applications … to support the specification, analysis, design, verification and validation of complex systems
Recommended Approach

- Identify CCSDS focal point to interface with SysML Partners
- Identify needs to SysML Partners
- CCSDS validate ability to apply (and further customize) SysML to their application
- Issues provide a basis for follow-up SysML / CCSDS joint efforts
Wrap up

- SysML Partners established with broad representation from industry, tool vendors and government agencies

- SysML addresses scope of UML for SE RFP Requirements

- SysML approach is to architecturally align with the UML superstructure and customize

- Major changes include:
  - Enhancements to composite structure and activity diagrams
  - Two new diagram types and New diagram usages

- SysML will be architecturally aligned with ISO AP-233

- First public draft specification available in April 2004 for submissions

- Establish liaison effort between CCSDS and SysML to validate application to Space Systems and identify issues for future planning
References

- **SysML**
  - [http://www.sysml.org](http://www.sysml.org)
  - Chairs
    - Cris Kobryn ([cris@ieee.org](mailto:cris@ieee.org))
    - Sandy Friedenthal ([sanford.friedenthal@lmco.com](mailto:sanford.friedenthal@lmco.com))

- **SE DSIG**
  - [http://syseng.omg.org](http://syseng.omg.org)

- UML for SE RFP – ad/03-03-41
- UML 2 Superstructure