SysML – a modeling language for Systems Engineering

IDA - Dansk Selskab for Datateknik
Ingeniørhuset i København, 15. Marts 2010
Finn Overgaard Hansen
foh@iha.dk
Ingeniørhøjskolen i Århus

Agenda

- Systems Engineering and SysML
- SE processes
- What is SysML?
- New SysML concepts and diagrams
  1. SysML Requirements
  2. SysML Structure
  3. SysML Behaviour
  4. SysML Parametric
- Perspectives for SysML
Systems Engineering

According to INCOSE:
“Systems Engineering is an **engineering discipline** whose responsibility is creating and executing an **interdisciplinary process** to **ensure** that the **customer and stakeholder’s needs are satisfied** in a **high quality, trustworthy, cost efficient and schedule compliant manner** throughout a system’s entire life cycle”

INCOSE: The International Council on Systems Engineering
Founded in 1990, 6720 members in dec. 2008

Model Based Systems Engineering (MBSE)

- From **document-based** to **model-based** approach
- A model-based approach requires modeling concepts and tools
- MBSE: producing and controlling a coherent **System Model**
- SysML is created to realize an MBSE approach based on a System model of the wanted system
- **SysML** is a modeling language not a **System Engineering (SE) process**
The **SIMILAR (SE) Process**

Customer Needs

- State the Problem
- Investigate Alternatives
- Model the System
- Integrate
- Launch the System
- Access Performance

Re-evaluate
Re-evaluate
Re-evaluate
Re-evaluate
Re-evaluate
Re-evaluate

Note: All functions are performed in a parallel and iterative manner.

The Harmony (SE) Process (IBM)

- The Harmony process facilitates a seamless transition from **Systems Engineering** to **Software Engineering**
  - It uses SysML exclusively for system representation and specification.
- Harmony process characteristics:
  - a scenario-driven and iterative development process
  - promotes reuse of test scenarios throughout system development
The Harmony process benefits

- The Harmony process models allow systems engineers to **find design errors** early in the development
- **Customer requests** can be more efficiently assessed, incorporated, and given timely feedback
- However, the **greatest benefit** of a model-driven process is **improved communication**
  - between engineering disciplines
  - between technical and non-technical parties
  - using different levels of abstraction
  - avoids information overload

What is SysML?

- A graphical modeling language created in response to the UML for Systems Engineering RFP developed by the **OMG and INCOSE**.
  - a UML Profile that represents a subset of UML 2 with important 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**
SysML Specification History and Status

- Nov. 1997: UML V1.1 launched by OMG
- March 2003: The UML for Systems Engineering RFP (Request for Proposal) was developed jointly by OMG and INCOSE
  - The SysML specification was developed in response to these requirements by the diverse group of tool vendors, end users, academia, and government representatives
- Sept. 2007: OMG SysML v.1.0
- Nov. 2008: OMG SysML v1.1
  - (doc.id: formal/2008-11-02, 256 pages)

System Engineering Technical Processes
System Model and SW/HW Components

Comparison of SysML and UML
SysML Diagram Taxonomy

Major Extensions to UML 2.x

- New Diagram Types
  - Requirement Diagram (req)
  - Parametric Diagram (par)
- Structure Diagrams
  - Block Definition Diagram (bdd)
  - Internal Block Diagrams (ibd)
- Activity Diagrams
  - extensions for continuous flow modeling
  - extensions to support control operators
The 4 Pillars of SysML

1. Requirements
2. Structure
3. Behavior
4. Parametrics

Project activities using SysML

1. Capture and analyze black box system requirements
   - System Context & System Use Cases, Requirement diagrams
2. Develop one or more candidate system architectures
   - Block Definition & Internal Block diagrams
3. Perform engineering trade-off analysis to evaluate and select the optimal architecture
   - Parametric Diagrams
4. Specify component requirements and their traceability to system requirements
   - Requirement diagram
5. Verify the system design by executing system-level test cases
1. SysML Requirements

- Requirement Diagram – a NEW diagram type
- Graphical visualization of requirements
  - Functional
  - Non-functional
- Requirements can graphical be related to:
  - Other requirements
  - Design elements
  - Test Cases
- Standard stereotypes:
  - derive, satisfy, verify, refine, trace and copy
  - Used for requirement traceability

Requirement Diagram Example
2. SysML Structure

- UMLs class concept is replaced with the **Block** concept
- A Block connect to other blocks via **Ports**
- Class diagrams are replaced with **Block Definition Diagrams (bdd)**
- Each Block has an **Internal Block Diagram (ibd)** where the internal parts are connected via **ports**
  - a replacement for class composite diagrams
- **Ports** can connect **discrete** as well as **continuous flows of material or information**
Blocks are Basic Structural Elements

- Provides a unifying concept to describe the structure of an element or system
  - Hardware
  - Software
  - Data
  - Procedure
  - Facility
  - Person

- Multiple compartments can describe the block characteristics
  - Properties (parts, references, values)
  - Operations
  - Constraints
  - Allocations to the block (e.g., activities)
  - Requirements the block satisfies

Blocks and Atomic Flow Ports

- A flow port describes an interaction point for items flowing in or out of a block

- An atomic flow port specifies only a single type of input or output

 optical image: Light  
external light: Light  
electrical image: Image  
Optical image: Image

Imaging Assembly

Optical Assembly
Blocks and Nonatomic Flow Ports

An interaction point with a complex interface is modeled as a **Nonatomic Port**

```
Camera <> «block» camera i/o: Camera Interface

Monitoring Station <> «block» station i/o: Camera Interface
```

**A Conjugate flow port**

Connector and Ports

```
```

```
:Light
```

```
:Connector
```

```
:Electronic Assembly
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```

```
:Image
```
Delegation Ports

Ibd [Block] Camera [Nested flow]

Camera Module

- :Optical Assembly
- :Imaging Assembly

Electronic Assembly

- :MPEG Converter
- :Image Processor

Delegation port

Standard (service based) ports

Camera Control

- operations
  - getCameraStatus(in cameraId: Integer, in cameraStatus: String)
  - testCamera()
  - panCamera(in strength: Integer)
  - tiltCamera(in strength: Integer)

Monitoring Station

Camera Control «interface»

Provided interface

Required interface
Block Definition Diagram Example

Internal Block Diagram for Automobile Domain
Block Definition Diagram Example

Internal Block Diagram Example
3. SysML - Behavior

- Activity diagrams are enhanced with new concepts
- Flows can be **continuous** and model **information** as well as **material flow**
- Control flows are introduced
- Activities can have pins

![Activity Diagram with parameter nodes]

**act** Operate Camera [Activity Frame]

- «optional»
  - currentImage : Light[0..1]
    (stream, direction = in )
- «optional»
  - MPEG output : MPEG4[0..1]
    (stream, direction = out )

- config : ConfigurationData
  (direction = in )

- «optional»
  - composite out : Composite[0..1]
    (stream, direction = out )
**Activity Diagram - decomposed**

```
act Operate Camera [Object Flow]
    \rightarrow CollectImages
        \rightarrow captured image \{stream\}

Object flow \rightarrow CaptureVideo
    \rightarrow video out \{stream\}

Outputs
    \rightarrow MPEG output \{stream\}

\rightarrow Composite out \{stream\}
```

**Activity Diagram Notation**

Flows can be discrete, streaming or control.
4. SysML Parametric

- Parametric Diagram – a NEW diagram type
- Used to express constraints (equations) between value properties
  - Provides support for engineering analysis (e.g., performance, reliability)
- Constraint block captures equations shown on a bdd
  - 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 $F=\text{m} \times a$)
- Parametric enable integration of engineering analysis with design models
BDD Parametric Constraint Blocks

Parametric Diagram - Example
Combining Model-Driven (MDD) and Model Based Design (MBD) in Industrial Machine Control

MDD: Model Driven Development in Rhapsody (IBM)  
MBD: Model Based Design in Simulink (Mathworks)

<table>
<thead>
<tr>
<th>Tool Environment Capabilities</th>
<th>Rhapsody</th>
<th>Simulink</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems and Software Development Environment</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mathematical Block Diagram Environment</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>UML 2.1, SysML 2.0 Based Code Generation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dynamic Controls and Signal Processing Engineering</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Logical Algorithm Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements Traceability &amp; Documentation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Define Physical, Functional &amp; Software Architecture</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mathematical Algorithm Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SysML/UML Analysis, Simulation &amp; Test</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Physical Plant Behavior Modeling</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1. MDD versus MBD feature comparison
Cross Connecting Model Elements

Requirements - satisfy

Structure - Behavior

Parametrics

SysML - a modeling language for Systems Engineering

Vendors of SysML tools

- ARTiSAN Software Tools
- EmbeddedPlus Engineering (Third party for IBM Rational)
- IBM
  - Rhapsody
  - Tau
- InterCAX
- No Magic
- Papyrus for SysML (open source eclipse modeling tool)
- Software Stencils - Microsoft Visio SysML and UML templates
- Sparx Systems
Perspectives for SysML

- Enable a common modeling language and model across engineering disciplines
- Enable traceability between disciplines
- Enable different kinds of system analysis
- Enable integration of discrete and continuous based modeling tools
- Critical enabler for Model Based System Engineering with tool support

Summary

- SysML a common modeling language for different disciplines e.g. Hardware, Software and Mechanics
- New and important concepts for cross disciplinary analysis of system properties (e.g. parametric)
- Blocks and ports as general modeling elements
- Important enhancement to activity diagrams
- Lot of support for traceability between models and model elements
- Must be supported by an appropriate SE process
References

- OMGs SysML homepage: [www.omgsysml.org](http://www.omgsysml.org)
- INCOSE organization: [www.incose.org](http://www.incose.org)
- IBM Rational Harmony:
- Books: