“Tool Vendor Perspectives – SysML Thus Far”

Hans-Peter Hoffmann, Ph.D
Chief Systems Methodologist
Telelogic, Systems & Software Modeling Business Unit
Peter.Hoffmann@telelogic.com
Tool Vendor Perspectives – SysML Thus Far

- SysML defines the standardized “vocabulary” of the language for model-based systems engineering. As a standard, this vocabulary needs to cover all possible applications.
- SysML does not specify, how to apply these “words”.
- Systems engineering is strongly communication-driven. Systems engineers have to communicate with stakeholders from different domains, e.g.
  - mechanical engineers
  - electrical engineers
  - software engineers,
  - test engineers, and not to forget
  - customers, who not necessarily have an engineering background.
- In such an environment it is paramount to keep the language domain independent and easy understandable.
- Compliance to a standard does not mean that all elements of this standard have to be applied.
- It is recommended to **standardize** the usage of the SysML within the organization, if a company wants to deploy SysML-based systems engineering.
Tool Vendor Perspectives – SysML Thus Far

- In many cases, customers - even if they are quite familiar with the SysML - struggle with the usage of the appropriate SysML diagrams because there is an overlap between the different diagrams.

- Key for a successful usage of the SysML is a **systems engineering process** that is an integral part of the model-driven development (MDD) process.

- The systems engineering process should define the essential SysML artifacts that are needed to enable a seamless transition to the subsequent HW/SW development, e.g. **Telelogic Integrated System/Embedded Software Development Process Harmony**.

- Our recommendation is, that the OMG, INCOSE, and associated groups should address this issue in respective forums.
Key requirement for model-based systems engineering is *model execution*.
- In the functional analysis phase, model execution assures that the requirements are correct, complete, and unambiguous.
- In the design synthesis phase, model execution verifies and validates the system architectural design incl. associated interfaces.

The OMG is working on this topic. The Telelogic tool *Rhapsody* already supports model execution. Important is, that model execution does not require that the user has to write code.

With regard to model execution, SysML *parametric diagrams* also need to be executable.

Currently, engineers put a lot of effort in to produce a decent parametric diagram, but it does nothing as a result of all this work. What is gained is an understanding of the mathematical principles which govern a particular problem, but it cannot be taken any further.
Backup Slides
Rhapsody 7.2
Advances in Systems Engineering
Advances in Systems Engineering

• OMG SysML™ 1.0 enhancements to better organize and communicate information effectively
  – Requirements Tables, Allocation Tables, N-2 Matrix
  – Value Types, Dimensions, Units help in trade study analysis
  – XMI 2.1 for SysML
    • Independently certified by NIST (National Institute of Standards and Technology)

• Improved design consistency
  – Improved user interface to pinpoint design errors in model
  – Create customizable checks to ensure compliance to company/project standards

• Integrated graphical panels validate design correctness
Tables and Matrix Views

• Organize large amounts of information concisely
• Requirements tables summarize requirements information
• Allocation tables show key information — how blocks are allocated
• N-2 matrices show how model elements are connected
• Define tables and matrices to organize any desired information
Value Types, Units, Dimensions

- Model physical dimensions and measurement units
- Enable trade study analysis of different designs by comparing units
- Ensure proper units are being used for system integration and parametrics
- Standard SI library includes standard units
Improved Model Consistency Checks

• Check for completeness, model integrity and correctness of the design
• Design quality into the model
• Configure the check model dialog easily
• Navigate directly to model errors
• Include your own check scripts to ensure the design meets your company standards
Graphical Panels

- Create mock ups of interface to effectively communicate intended design behavior to customers
- Easily modify, monitor and analyze data values during simulation to ensure the design is correct early in the process
Integrated System / Embedded Software Development Process *Harmony*
Model-Driven Development of Embedded Systems

Stakeholder Requirements
- Requirements Models
- Use Case Model

- Executable Use Case Models
- Architectural Analysis Model(s)
- Executable System Architecture Model

System Architecture Baseline

Software Implementation Model

System Changes

Requirements Analysis
- Scenarios

Systems Engineering Harmony/SE

Sw Systems Analysis & Design

Sw Analysis & Design

Sw Implementation & Unit Test

Model / Requirements Repository *

(Sub-)System Integration & Test

Module Integration & Test

System Acceptance

* Configuration Controlled Knowledge of the System Under Development:
  - Requirements Documentation
  - Requirements Traceability
  - Design Documentation
  - Test Definitions
Essential SysML Artifacts for Model-Based Systems Engineering
Model-Based Systems Engineering (Harmony/SE)

Artifact Relationships at the Requirements Analysis / Functional Analysis Level

- Use Case Diagram
- Use Case
- Sequence Diagram
- Activity Diagram
- Structure Diagram
- Block Definition Diagram
- Internal Block Diagram
- Block
- Statechart Diagram
- Requirements Diagram

Relationships:
- 1:1
- 1:*
Key Objectives of the Model-Based Systems Engineering Process *Harmony/SE*

- Identify / derive required system functionality
- Identify associated system modes and states
- Allocate system functionality / modes to a physical architecture
In the Requirements Analysis phase, the focus is on the analysis of the process inputs. Customer requirements are translated into a set of requirements that define
- what the system must do (functional requirements) and
- how well it must perform (quality of service requirements).

Once the requirements are sufficiently understood they are grouped into Use Cases.
In the System Functional Analysis phase, the focus is on the translation of the functional requirements into a coherent description of system functions (Operations). Each use case is translated into a model and the underlying requirements verified and validated through model execution.
In the Design Synthesis phase, the focus is on the allocation of system-level operations to a system architecture - optionally elaborated through trade studies - and on the definition of ports / interfaces and state-based behavior at the lowest level of the structural decomposition.