Systeme hoher Qualität und Sicherheit
Universität Bremen WS 2015/2016

Lecture 05 (09-11-2015)

High-Level Design with SysML

Christoph Lüth Jan Peleska Dieter Hutter

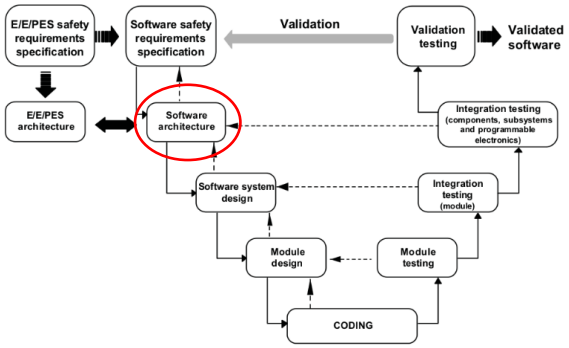
Universität Bremen

Where are we?

Your Daily Menu

- What is high-level design?
  - Describing the **structure** of the system at an abstract level
  - Should fit with **formal model** at lower level
- In which language?
  - Wide-spectrum specification languages such as Z, B, Event-B, CASL, ...
  - Architectural languages
  - Modeling languages such as the UML
  - UML is very software-centred, hence SysML
- Today:
  - Introduction to SysML
  - Structural modeling in SysML

High-Level Design in the Development Cycle



An Introduction to SysML

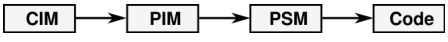
Universität Bremen

What is a model?

- „A model is a representation in a certain medium of something in the same or another medium. The model captures the important aspects of the thing being modelled from a certain point of view and simplifies or omits the rest.“ Rumbaugh, Jacobson, Booch: UML Reference Manual.
- In other words: an **abstract representation of reality**.
- Purposes of models:
  - Analysing requirements
  - Understanding, communicating and capturing the design
  - Organizing information about a large system
  - Analyse design decisions early in the development process

Model-Driven Development (MDD, MDE)

- Recall the idea of MDD:
  - Describe problems on abstract level using *a modelling language* (often a *domain-specific language*), and derive implementation by model transformation or run-time interpretation.
  - Often used with UML (or its DSLs, eg. SysML)



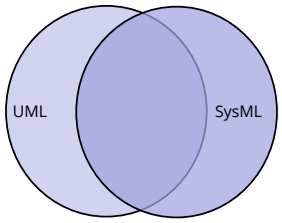
- However, using a modelling language like UML or SysML does not mean one has to employ MDD; in particular, we can still employ V-model-like approaches as required by safety standards.

The Unifed Modeling Language (UML)

- The UML grew out of a wealth of modelling languages in the 1990s, as James Rumbaugh, Grady Booch and Ivar Jacobson all worked at Rational Software.
- It was adopted by the Object Management Group (OMG) in 1997, and approved as ISO standard in 2005.
- UML 2 consists of
  - the superstructure to define diagrams,
  - a core meta-model,
  - the object constraint language (OCL),
  - an interchange format
- UML 2 is not a fixed language, it can be extended and customised using profiles.

The Systems Modeling Language SysML

- SysML is a *modeling language* for **systems engineering**
- Standardised in 2007 by the OMG (Ver. 1.0, now at 1.3)
- SysML Standard available at: http://www.omg.org/spec/SysML/1.3/PDF
- UML vs. SysML:



What for SysML?

- The aim of SysML (much like UML) is to serve as a standardised notation allowing all stakeholders to understand and communicate the salient aspects of the system under development:
  - the requirements,
  - the structure (static aspects), and
  - the behaviour (dynamic aspects).
- Certain aspects (diagrams) of the SysML are **formal**, others are **informal.**
  - Important distinction when developing critical systems
- All diagrams are **views** of one underlying model.

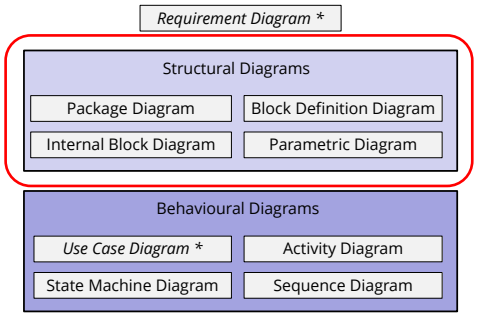
Views in SysML

- Structure:
  - How is the system constructed? How does it decompose?
- Behaviour:
  - What can we observe? Does it have a state?
- Requirements:
  - What are the requirements? Are they met?
- Parametrisation:
  - What are the constraints (physical/design)?
- … and possibly more.

Example: A Cleaning Robot (HooverBot)

- Structure:
  - Has an engine, wheels (or tracks?), a vacuum cleaner, a control computer, a battery…
- Behaviour:
  - General: Starts, then cleans until battery runs out, returns to charging station
  - Cleaning: moves in irregular pattern, avoids obstacles
- Requirements:
  - Must cover floor when possible, battery must last at least six hours, should never run out of battery, …
- Constraints:
  - Can only clean up to 5g, can not drive faster than 1m/s, laws concerning movement and trajectory, …

SysML Diagrams



* Not considered further.

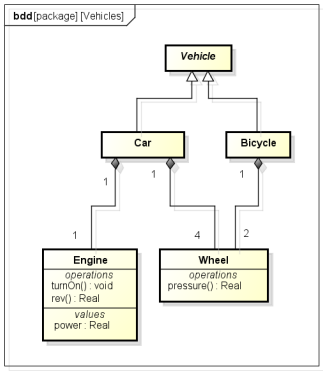
Structural Diagrams in SysML

Block Definition Diagram

- Corresponds to *class diagrams* in the UML
- Blocks are the basic building elements of a model
  - Models are *instances* of blocks
- Block definition diagrams model blocks and their relations:
  - Inheritance
  - Association
- Blocks can also model interface definitions.

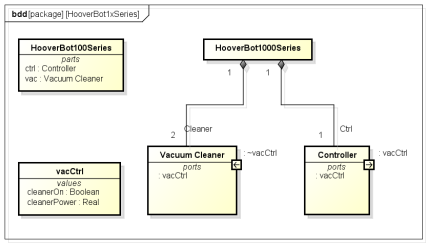
Example 1: Vehicles

- A vehicle can be a car, or a bicycle.
- A car has an engine
- A car has 4 wheels, a bicyle has 2 wheels
- Engines and wheels have operations and values
- In SysML, Engine and Wheel are *parts* of Car and Bicycle.



Example 2: HooverBots

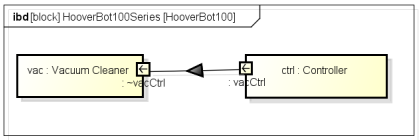
- The hoover bots have a control computer, and a vacuum cleaner.
  - HooverBot 100 has one v/c, Hoover 1000 has two.
  - Two ways to model this (i.e. two views)

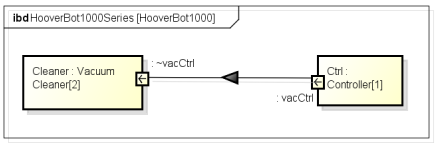


Internal Block Diagrams

- Internal block diagrams decribe instances of blocks.
- Here, instances for HooverBots
- On this level, we can describe connections between ports (flow specifications)
  - Flow specifications have directions.

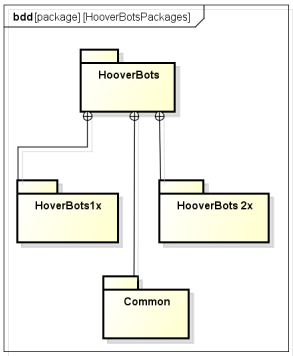
HooverBot 100 and 1000





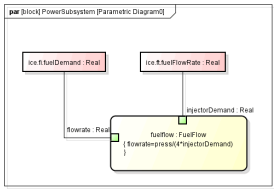
Package Diagrams

- Packages are used to group diagrams, much like directories in the file system.
- Not considered much in the following



Parametric Diagrams

- Parametric diagrams describe constraints between properties and their parameters.
- It can be seen as a restricted form of an internal block diagram, or as equational modeling as in Simulink.



Source:
http://astah.net/tutorials/sysml/parametric

Modeling Tool: Astah-SysML

- Astah-SysML is available at

  http://astah.net/editions/sysml

- A faculty licence is available for FB3 Uni Bremen
  - Non-commercial use only, do not distribute!
- The tool not only helps with the drawing, it also keeps track of the relationship between the diagrams: you edit the model rather than the diagrams.

Summary

- High-level modelling describes the structure of the system at an abstract level.
- SysML is a standardised modelling language for systems engineering, based on the UML.
  - We disregard certain aspects of SysML in this lecture
- SysML structural diagrams describe this structure.
  - Block definition diagrams
  - Internal block definition diagrams
  - Package diagrams
- We may also need to describe formal constraints, or invariants.
- For this: OCL --- next week.