Towards SysML v2: should you be worried about technical debt?

Software Center Lunch Seminar, 2021-03-22

Antonio Cicchetti

Mälardalen University, IDT, Västerås

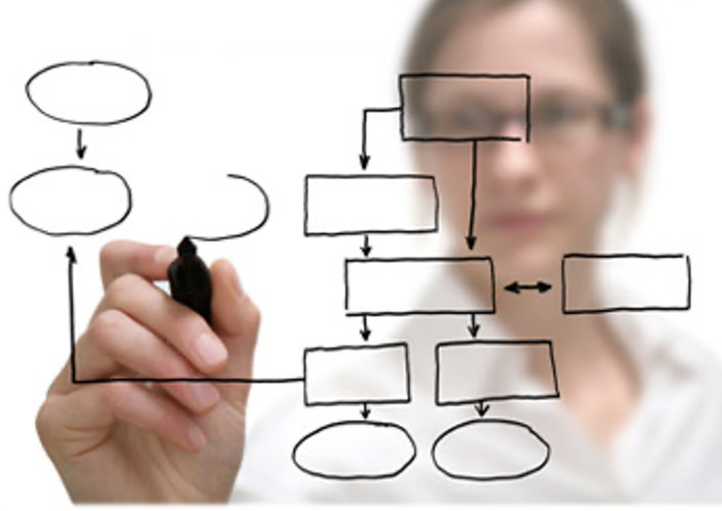
antonio.cicchetti@mdh.se

MÄLARDALENS HÖGSKOLA
ESKILSTUNA VÄSTERÅS

Agenda

- Introduction
- Selected SysML v2 highlights
- The migration problem
- Conclusions and Outlook

Spoiler



- SysML v2 includes a number of interesting features that make its adoption appealing

- The new version is radically different from v1.6 (or 1.4), indeed a proper new language, posing several issues related to legacy (mainly migration)

- Depending on the use you currently make of SysML, upgrading to v2 might introduce different degrees of technical debt

Introduction

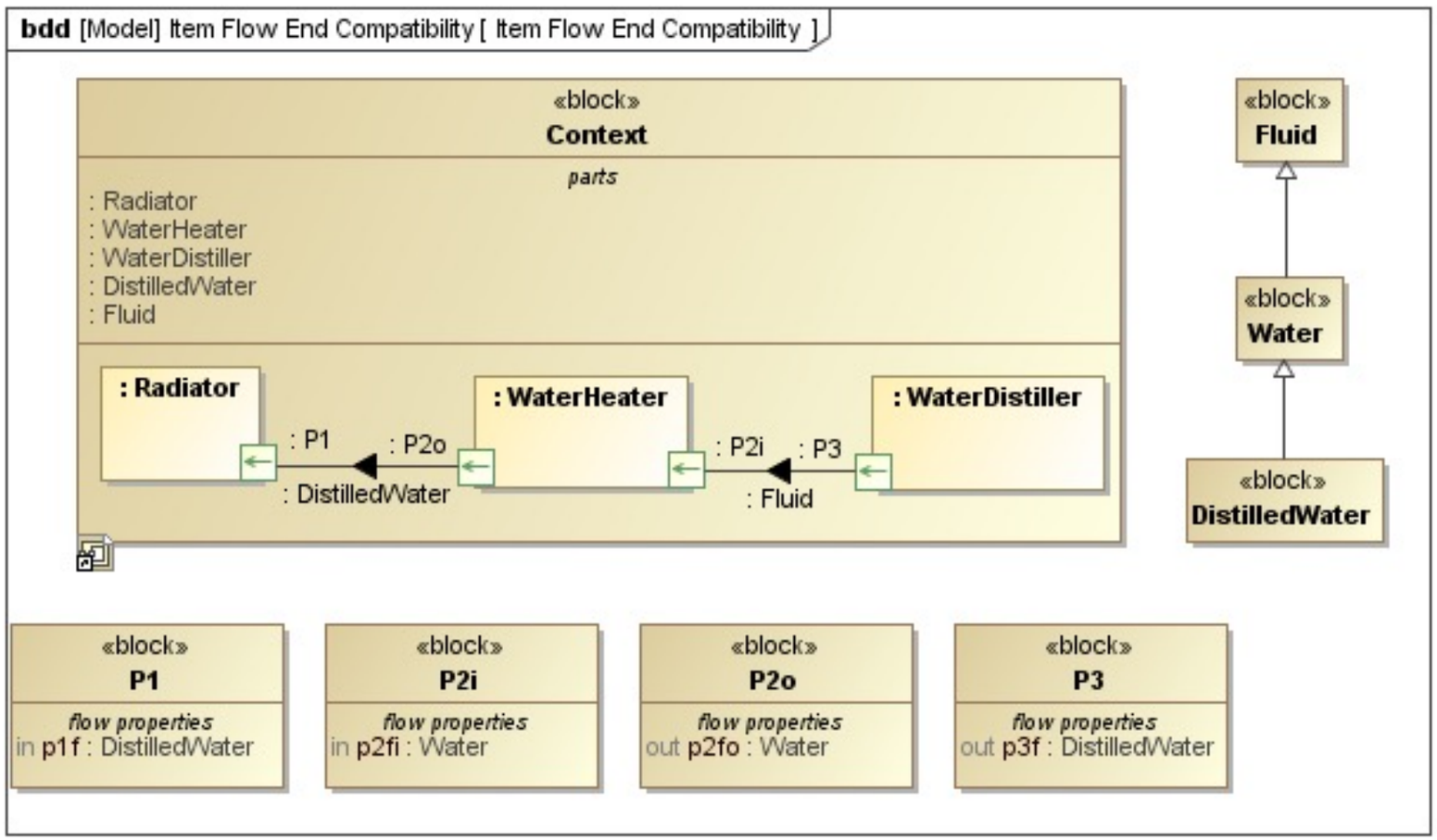
- Growing software ubiquity
  - Also in domains where software traditionally played a smaller role
  - Often cross-cutting legacy development silos
- Continuous Architecting
  - Need for shorter lead-time
  - Pressures for reuse
- Model-based System Engineering
  - Highlights the need for development at higher abstraction levels
  - Use of models for documenting, communicating, analysing and driving software implementation

Introduction

- What is SysML
  - A standard general-purpose language for system modelling
  - (Originally) a language extension/profile of UML

Introduction

- What is SysML
  - A standard general-purpose language for system modelling
  - A language extension/profile of UML



From No Magic documentation web site:
https://docs.nomagic.com/display/SYSMLP190/SysML+Block+Definition+Diagram

Introduction

- What is SysML
  - A standard general-purpose language for system modelling
  - A language extension/profile of UML

- *Supports the specification, analysis, design, and verification and validation of complex systems that may include hardware, software, information, processes, personnel, and facilities*[1]

- Design approach
  - “Onion-skin” system specification
  - Recursive structure to allow arbitrary levels of detail

1. Ed Seidewitz, Introduction to the OMG Systems Modeling Language (SysML), Version 2. October 2020. From Slideshare: https://www.slideshare.net/seidewitz/sys-ml-v2-201016-models-sysml-v2-tutorial-238904153

Why a v2 for SysML

- Formal foundation
- A new metamodel
- Textual notation
- Standard API

Why a v2 for SysML

- Formal foundation
  - Recursively built on top of a core (KerML) derived from formal logic
  - Automation of (semantics) validation checks
- A new metamodel
- Textual notation
- Standard API

Why a v2 for SysML

- Formal foundation
  - Recursively built on top of a core (KerML) derived from formal logic
  - Automation of (semantics) validation checks
- A new metamodel
  - Defined as a proper DSL based on the OMG MOF metamodel
  - The language is released from constraints inherited by UML (e.g. instance specific values, feature diagrams/variants)
- Textual notation
- Standard API

Why a v2 for SysML

- Formal foundation
  - Recursively built on top of a core (KerML) derived from formal logic
  - Automation of (semantics) validation checks
- A new metamodel
  - Defined as a proper DSL based on the OMG MOF metamodel
  - The language is released from constraints inherited by UML (e.g. instance specific values, feature diagrams/variants)
- Textual notation
  - The “default” language notation is textual
  - Starting from text it is possible to derive multiple notations for the same models, e.g. diagrams, tables, etc.
- Standard API

Why a v2 for SysML

- Formal foundation
  - Recursively built on top of a core (KerML) derived from formal logic
  - Automation of (semantics) validation checks
- A new metamodel
  - Defined as a proper DSL based on the OMG MOF metamodel
  - The language is released from constraints inherited by UML (e.g. instance specific values, feature diagrams/variants)
- Textual notation
  - The “default” language notation is textual
  - Starting from text it is possible to derive multiple notations for the same models, e.g. diagrams, tables, etc., including import/export between tools
- Standard API
  - Platform independent service and operation definition (logical API model)
  - Enhanced tool chaining opportunities

What a SysML v2 model looks like

```
import ISQ::*;
import SI::*;
import ScalarFunctions::*;

constraint def MassConstraint (
   partMasses : MassValue[0..*],
   massLimit : MassValue) {

   sum(partMasses) <= massLimit
}

part def Vehicle {
   assert constraint massConstraint : MassConstraint (
      partMasses = {chassisMass, engine::mass, transmission::mass},
      massLimit = 2500@[kg]);

   attribute chassisMass : MassValue;

   part engine : Engine {
      attribute mass : MassValue;
   }

   part transmission : Engine {
      attribute mass : MassValue;
   }
}
```

A *constraint assertion* asserts that a constraint *must* be true.

ⓘ If an assertion is violated, then the model is *inconsistent*.

Ed Seidewitz, Introduction to the OMG Systems Modeling Language (SysML), Version 2. October 2020. From Slideshare: https://www.slideshare.net/seidewitz/sys-ml-v2-201016-models-sysml-v2-tutorial-238904153

The migration problem

- When moving from older versions to SysML v2, previous SysML models become invalid

- The Object Management Group (OMG) is already planning guidelines and support for the transition

- Possible migration scenarios
  - Non-breaking changes (new elements in v2, e.g. the support for variants)
  - Breaking and resolvable changes (elements in v2 with a corresponding old element, e.g. part def <-> block)
  - Breaking and unresolvable changes (ports are no more defined as part of a block, rather a block exposes an interface conforming to a port definition)

The migration problem

- What about technical debt?
  - The standard will be released (probably) by the end of 2021
  - Tool providers will (most probably) provide a profile based implementation of the standard

- In the short term
  - You will be able to open “old” models
  - You will not be able to take advantage of most of the features due to v2 being a DSL per se

Analysing the potential technical debt

- In the long term
  - The current use of SysML in an organization has a fundamental impact on the technical debt
  - “What” (which concepts/diagrams?) is used will impact how wide/deep the effects will be
  - “How” (documentation/communication/development?) is used will impact how to manage the debt

Analysing the potential technical debt

- Project proposal in the upcoming SwC sprint (end of Spring 2021)
  - A. Martini and myself
  - Assessing the potential technical debt due to SysML v2 and evaluating a migration strategy
  - The main goal of the first sprint would be to understand the “what” and “how” of SysML usage at interested companies

- Link to the current SysML v2 specification draft (Github repo):

https://github.com/Systems-Modeling/SysML-v2Release/tree/master/doc