Towards scalable, smart SysML-based Systems Engineering

Pitch
9.10.2024

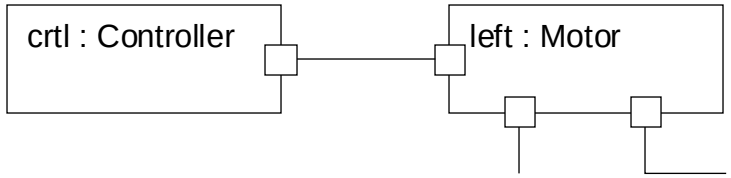
Bernhard Rumpe
Software Engineering
RWTH Aachen
http://www.se-rwth.de/

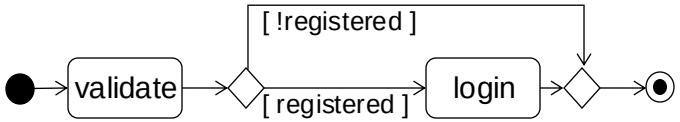
SysML v2

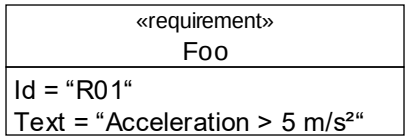
- SysML v2 is about to finish its standardization process

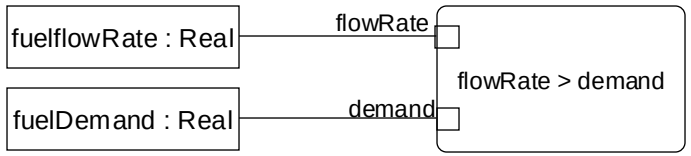
- Four pillars of SysML
  - Structure: Specification of hierarchies, ports, interconnection, model organization
  - Behavior: Specification of sequences of actions, life cycle of a block, message-based behavior
  - Requirements: Specification of requirements and relationships among model elements
  - Parametrics: Constraints, enables integration of engineering analysis and design models

- Interesting new capabilities
  - human readable textual form of the SysML language
  - in the spirit of a modern programming language
  - (and a number of special constructs that resemble modelling concepts)







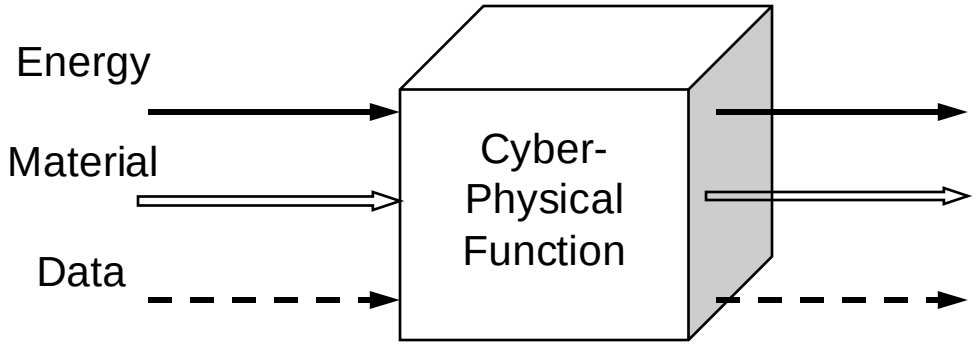


SPES methodology for SysML: System Specification through Functions

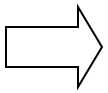
- A system defines a cyber-physical function
  - it encapsulates a physical and computational structure
  - performs data, energetic and physical transformations
  - and is connected to its context through its interfaces.

- A system function is described through its input and output signature

- Modularity and encapsulation is key to reuse



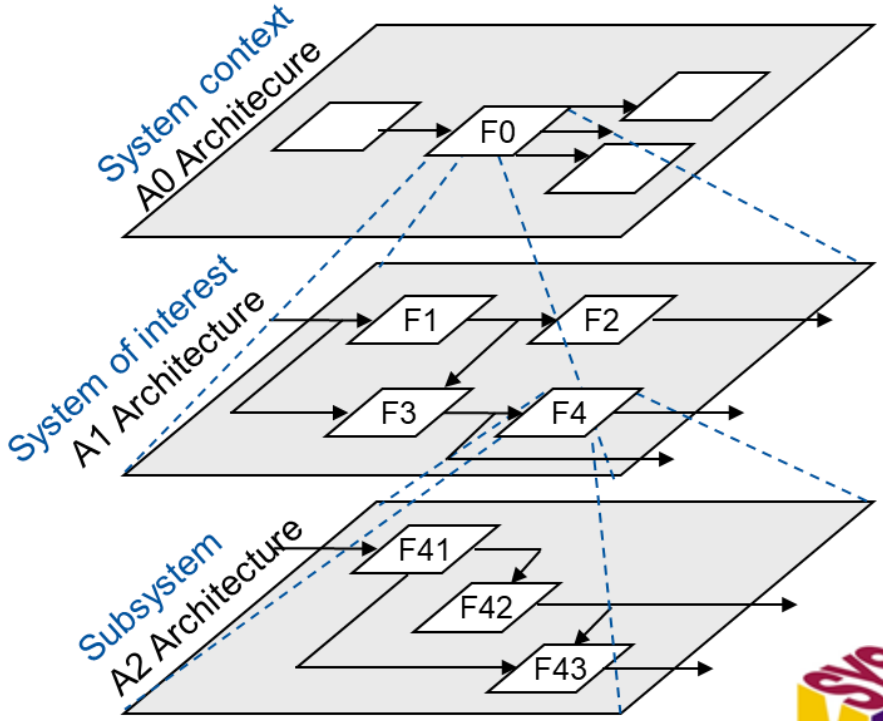

```
standard library package 'Vehicles' {
  import ISQ::TorqueValue;
  import ScalarValues::*;
  part def Automobile;
  alias Car for Automobile;
  alias Torque for ISQ::TorqueValue;
}
```



The concept of function a universal specification and construction principle

Abstraction and Composition

- SysML V2 allows abstraction and composition
- Model signature vs. model “internals”
- Underspecification as abstract behavior description
- Stable standardized interfaces
  (+ conservative extensibility, e.g. new channels)
- Variants of implementations
- (SPES methodology is available for SysML)



⇨ Abstraction and composition are the most important design capabilities

Creating SysML v2 Models with ChatGPT

- User is not required to provide SysML Related instructions
- Capabilities of LLM are used to interpret informal specification and to handle **underspecification**.

Try it yourself

sysml-v2-model-creator

https://chatgpt.com/g/g-3DkGCDlNM-sysml-v2-model-creator

Create a Model for a Dishwasher!

Here's a SysML v2 model for a dishwasher:

```
package DishwasherModel {

    package Definitions {
        part def Dishwasher {
            part washCycle: WashCycle;
            part controlPanel: ControlPanel;
            part waterPump: WaterPump;
            part heatingElement: HeatingElement;
            part detergentDispenser: DetergentDispenser;
            part door: Door;
            port waterInlet: WaterInletPort;
            port waterOutlet: WaterOutletPort;
        }

        part def WashCycle {
```

Use Cases for LLMs in Modelling

Use Cases

1. Transforming informal requirements into SysML v2
2. Working with an existing model
   - “What does this model describe ?”
   - “Does this model contain … ?”
   - “Does the model fit to the given requirement?”
3. Modifying or extending an existing model
4. Refactoring a model for non-functional optimization
5. Explaining a model
6. Creating tests for the models
7. Extract abstraction from an implementation/detailed model

- LLMs like **ChatGPT**, **LLama, Bard**, **Gemini** show amazing conversational skills
- We can ‚**teach**‘ an LLM to use SysML V2 and even specific company-internal libraries when responding
- Imagine what happens, when we give an LLM a homework?
  - (i.e. let it think for a while on a larger set of requirements)