MODEL-BASED SYSTEMS ENGINEERING – A WINDING ROAD

ERICSSON MODELING DAYS
2016-09-13

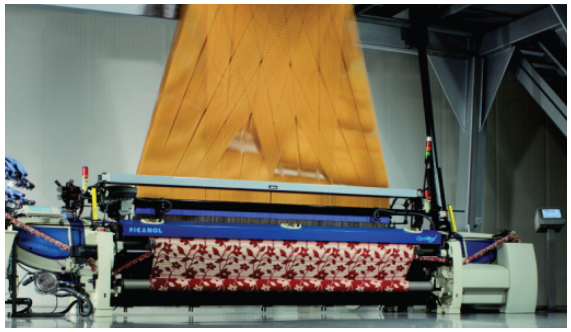
KLAAS GADEYNE, JOHAN VAN NOTEN

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Mission Flanders Make

To strengthen the **long-term international competitiveness** of the Flemish manufacturing industry by carrying out **excellent, industry-driven, pre-competitive research** in the domains of

- ▲ **Mechatronics**
- ▲ **Product development methods**
- ▲ **Advanced manufacturing technologies**



Aiming at product & process innovation
for the **vehicles, machines** and **factories of the future**

Eight research programs within three technology domains

Mechatronics

- RP1 – Clean Energy-Efficient Motion Systems
- RP2 – Smart Monitoring Systems
- RP3 – High-Performance Autonomous Mechatronic Systems

Product Development Methods

- **RP4 – Intelligent Product Design Methods**
- RP5 – Design & Manufacturing of Smart and Lightweight Structures

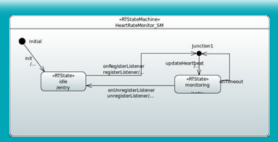
Advanced Manufacturing Technologies

- RP6 – Additive Manufacturing for Serial Production
- RP7 – Manufacturing for High Precision Products
- RP8 – Agile & Human-Centered Production and Robotic Systems

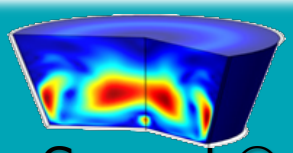
Our partner network

CNH INDUSTRIAL
BOMBARDIER the evolution of mobility
c-mec creative mechanical solutions
Gilbos
achillesdesign
RECTICEL The passion for comfort
Special machine construction VASKON Mould manufacturing & repair
BEKAERT better together
LayerWise Your Partner in Additive Manufacturing
Atlas Copco
PHILIPS
pecco
XEIKON
punch powertrain Gear up for the future
JABIL
DAF
Nikon
ZF
bosal
televic rail
TE connectivity
BEST GETS EVERYTHING SORTED
BLUWAYS
.Ocean nautical innovations
AGFA
Agfa
NiniX Technologies
Materialise innovators you can count on
TENNECO
vest
Test & Measurement Solutions
Hansen industrial gearboxes
LVD
VOLVO VOLVO TRUCKS
SIEMENS
CENAT CONTROL ENGINEERING & AUTOMATIO
Xenics Infrared Solutions
VDL
DAIKIN
TERUMO We keep life flowing
Continental
DEFAWES GEARS AND GEARBOXES
triphase
TRANSPORT & MOBILITY LEUVEN
DANA
Flexible Robotic Solutions - ahead in innovation -
SABCA LIMBURG NV
VAN DE WIELE
OCTINION Engineering the future
SPICER DRIVETRAIN COMPONENTS
oerlikon balzers
VHAR Van Hoecke Automation nv
TOYOTA
MAEX PRECISION-PRODUCTION NV
LEUVEN air bearings
elotte
PICANOL GROUP
GRAMMER
AGORIA no future without technology
sirris driving industry by technology
KU LEUVEN
Universiteit Antwerpen
universiteit hasselt
UNIVERSITEIT GENT
Vrije Universiteit Brussel
Vlaamse overheid

Our model-based context
Example: a CupCake production line

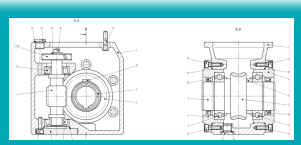


Papyrus-RT
Coordination

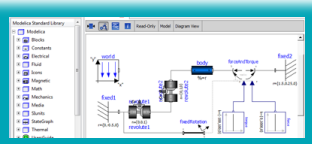


Comsol ®
Thermal analysis

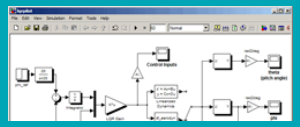
System Engineering =
- Guarantee consistency
- Streamline collaboration
- Make the right decisions
→ Language & tool required



AutoCAD ®
Mechanical design



OpenModelica
Physical modeling



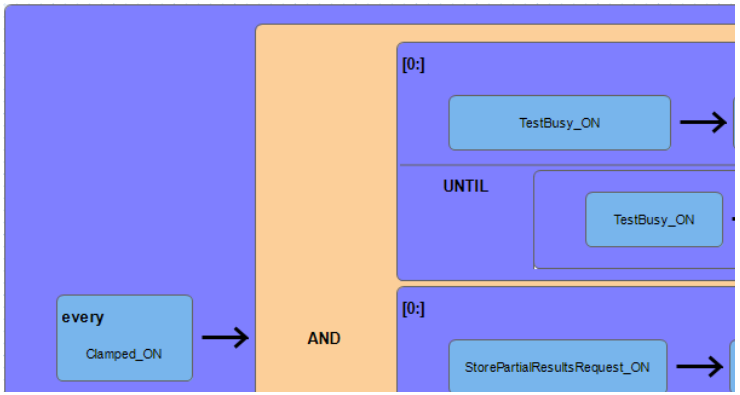
Simulink ®
post-proc. control

What models to use?

Very domain specific models

- Tools such as Ecore, Xtext, Graphiti

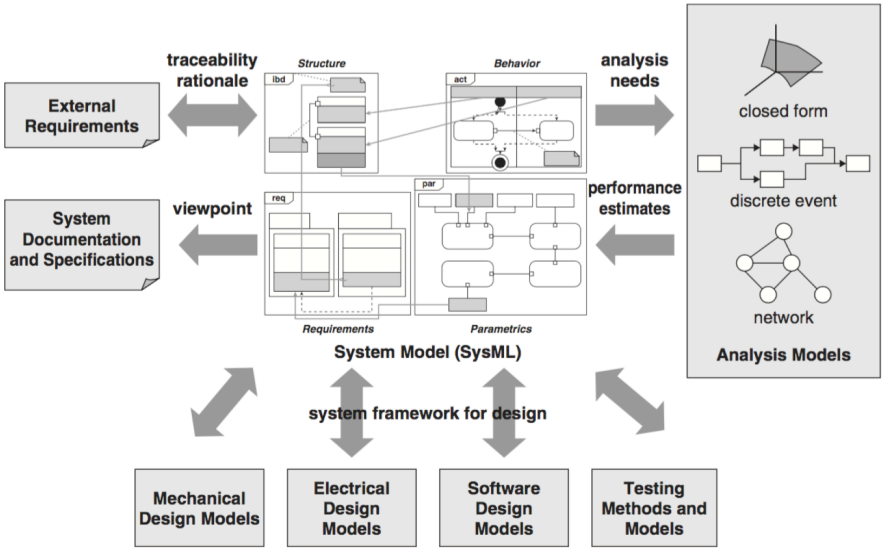
- E.g. a graphical language
  to produce stream processing expressions



Domain of Systems / Mechatronics / CPS

- Tools: Papyrus UML + SysML + Profiles

- E.g. SysML as a pivot model as shown in
  "A Practical guide to SysML", Friedenthal, e.a.

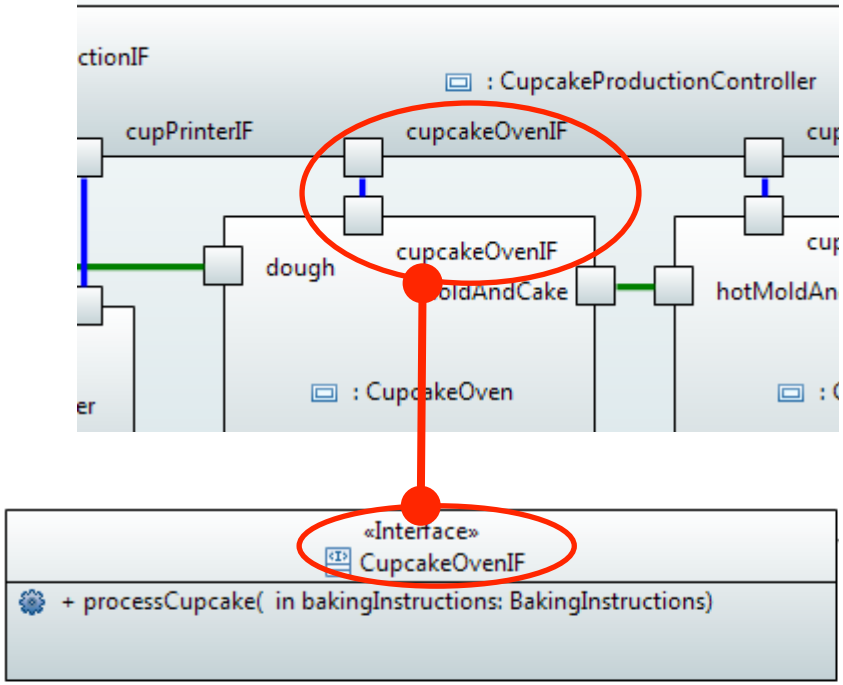


Examples
Describing system architecture

- Required features
  - Documenting interfaces
  - Describe behavior & structure

- How?
  - Standard SysML
    - Activity diagrams
    - Block diagrams
    - Internal block diagrams
  - Tooling such as Papyrus

→ Useful & simple

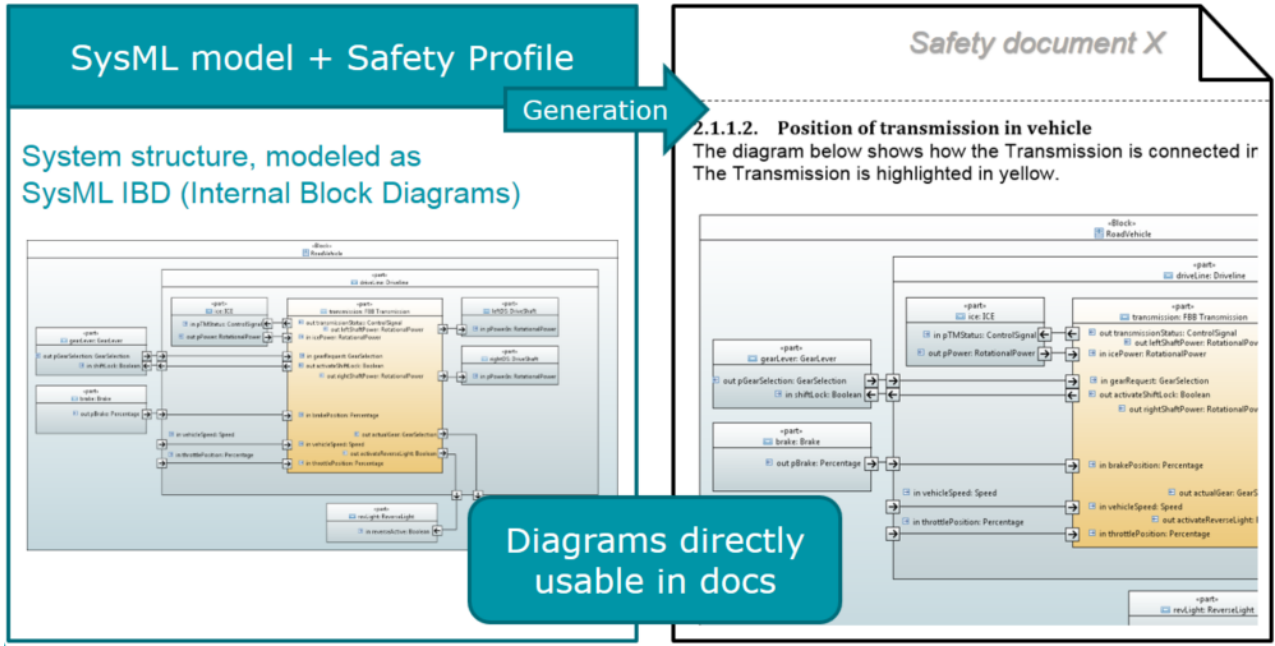


Examples
Generating documentation

- Required features
  - Generate documents / websites based on modeled information

- How?
  - Not provided by SysML
  - Additional tooling: GenDoc (or similar)

→ Useful & relatively simple



Model-based SE: one clear choice

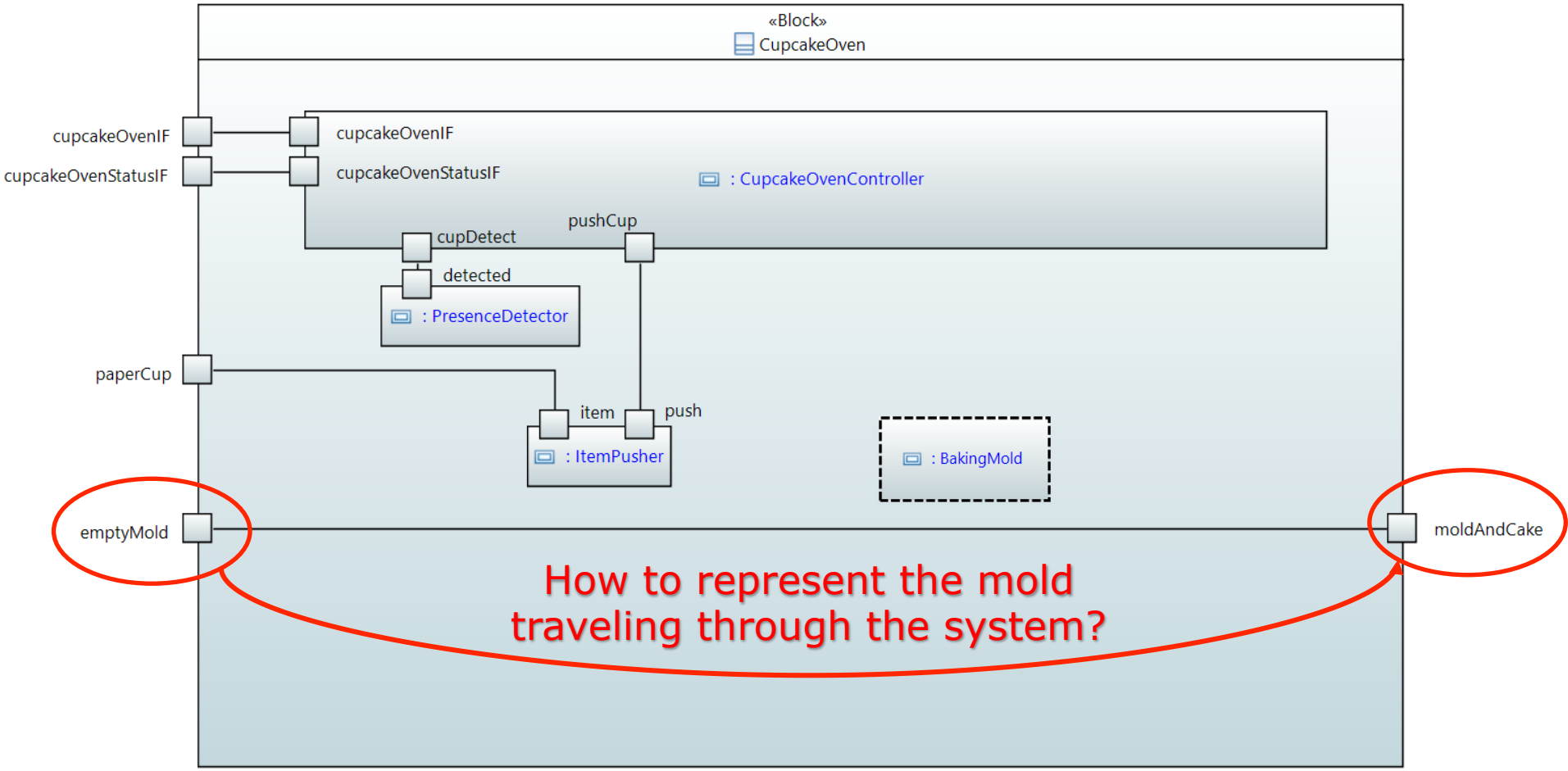
System Engineering? Let's go for

SysML solves everything!

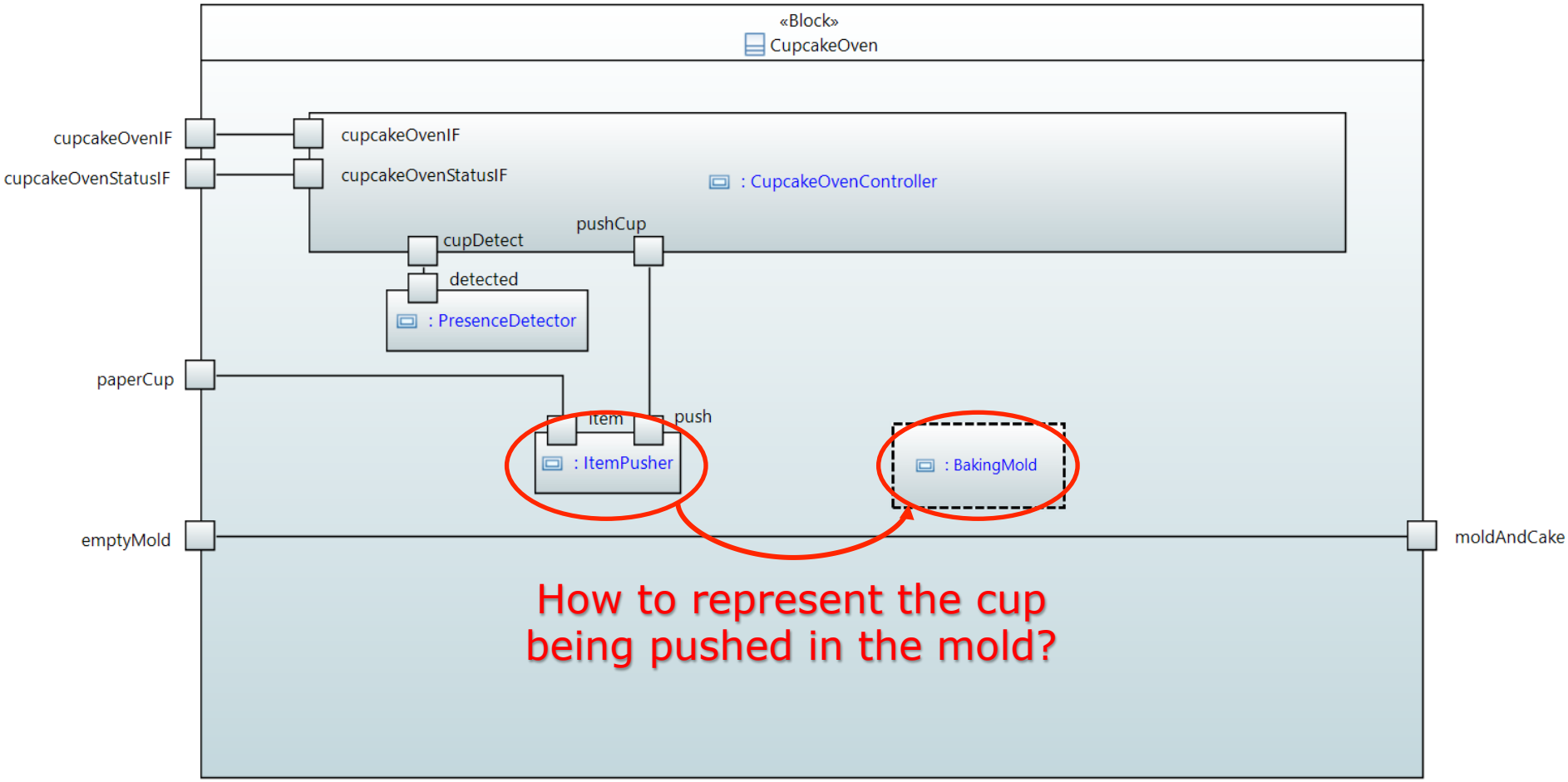
Examples
Easy real-live representation

- Required features of the CupCake oven
  - PaperCup, baking mold & dough enter the oven
  - Cup is put in a baking mold
  - Dough is applied to the cup
  - Cup + baked cake leave the oven
- How?
  - Anybody knows how to represent the structure (not the process) in SysML?
  - Should be clear representation for all team members!

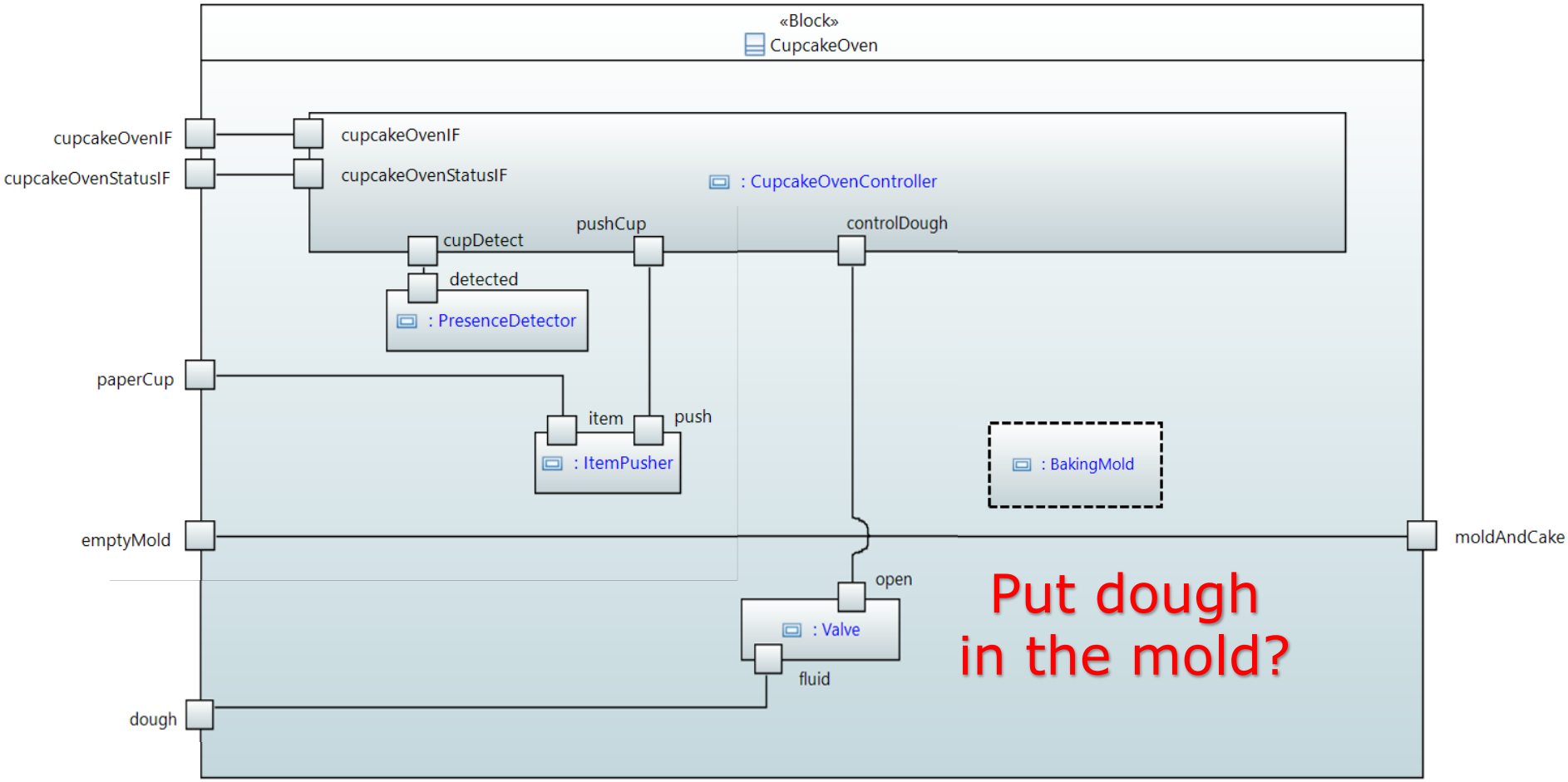
Examples
Easy real-live representation



Examples
Easy real-live representation



Examples
Easy real-live representation



Examples
Easy real-live representation

- SysML approach:
  - Different views
  - Separation of behavior & structure
- Drawback:
  - Domain experts don’t understand the “drawings” anymore
  - Me neither…
- This was exactly one of the required aspects! 

Model-based SE: one clear choice

System Engineering? Let's go for

****!
SysML
is useless!

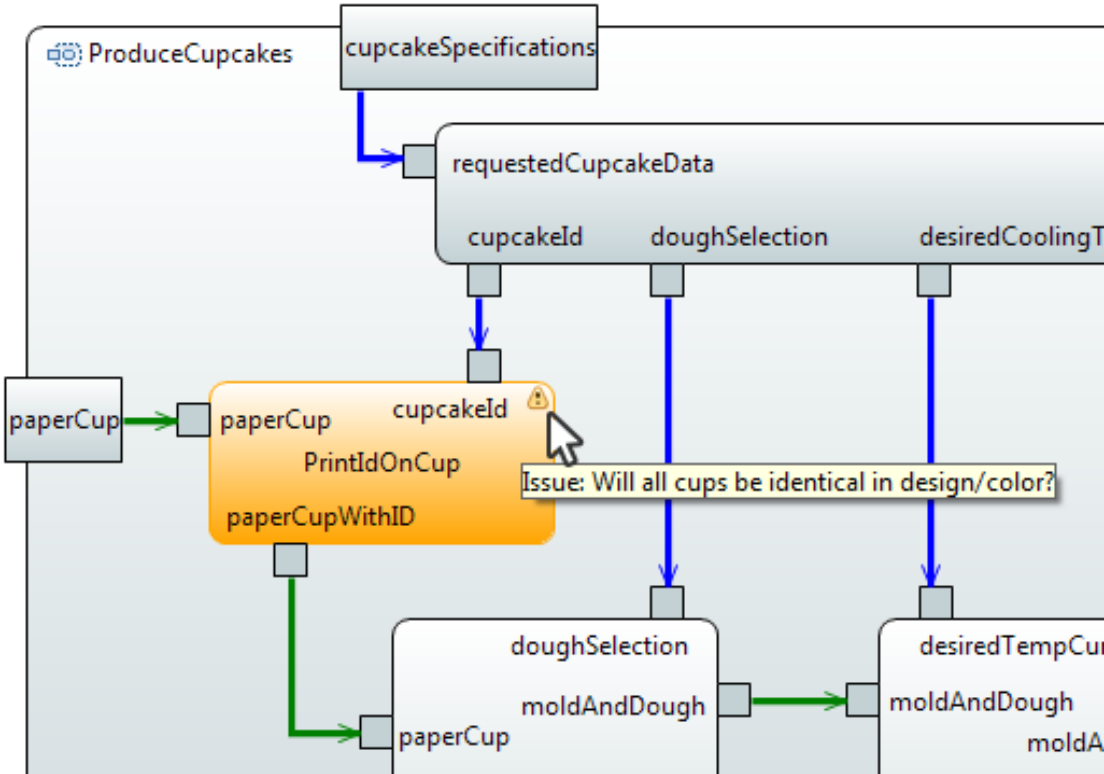
~~SysML solves everything!~~

Examples
Storing issues & decisions

- Required features
  - Shows to-be-discussed elements in orange
  - Attached to each element are issues / decisions / rationales
  - Hover over ⚠ shows attached issues

- How?
  - Not provided by SysML
  - Tool smith needs to:
    - Define Profile
    - Modify CSS
    - Implement validation

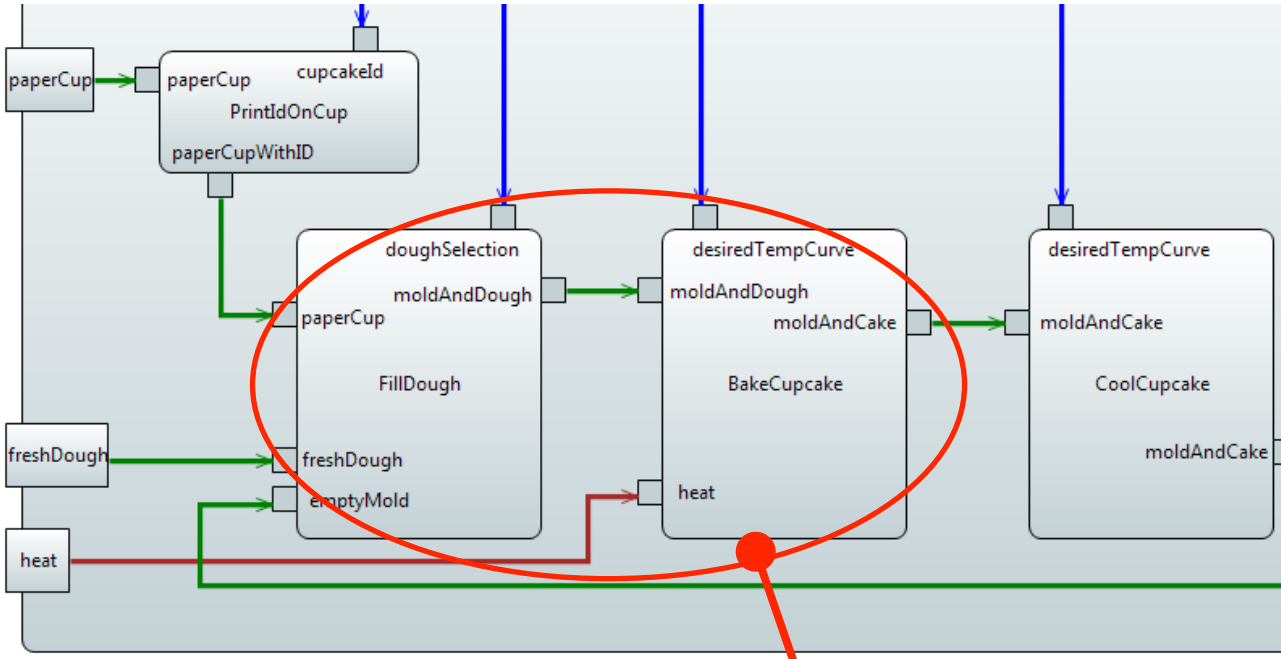
→ Feasible, but not trivial



Examples
Allocation between abstraction levels

- Required features
  - Allocate relationship
  - Tables
  - Automatic sources
- How?
  - Base by SysML
  - Tables by Papyrus
  - Tool smith needs to
    - Define table type
    - Code table population

→ Feasible, but not trivial



	Target
<Abstraction> ACT - CleanBakingMold	moldCleaner
<Abstraction> ACT - DecorateCupcake	cupcakeDecorator
<Abstraction> ACT - CoolCupcake	cupcakeCooler
<Abstraction> ACT - BakeCupcake	cupcakeOven
<Abstraction> ACT - FillDough	cupcakeOven
<Abstraction> ACT - ControlProduction	cupcakeProductionController
<Abstraction> ACT - PrintIdOnCup	cupPrinter

Model-based SE: one clear choice

System Engineering? Let's go for OMG SYSTEMS MODELING LANGUAGE SysML™

SysML is a solid base!

~~SysML solves everything!~~

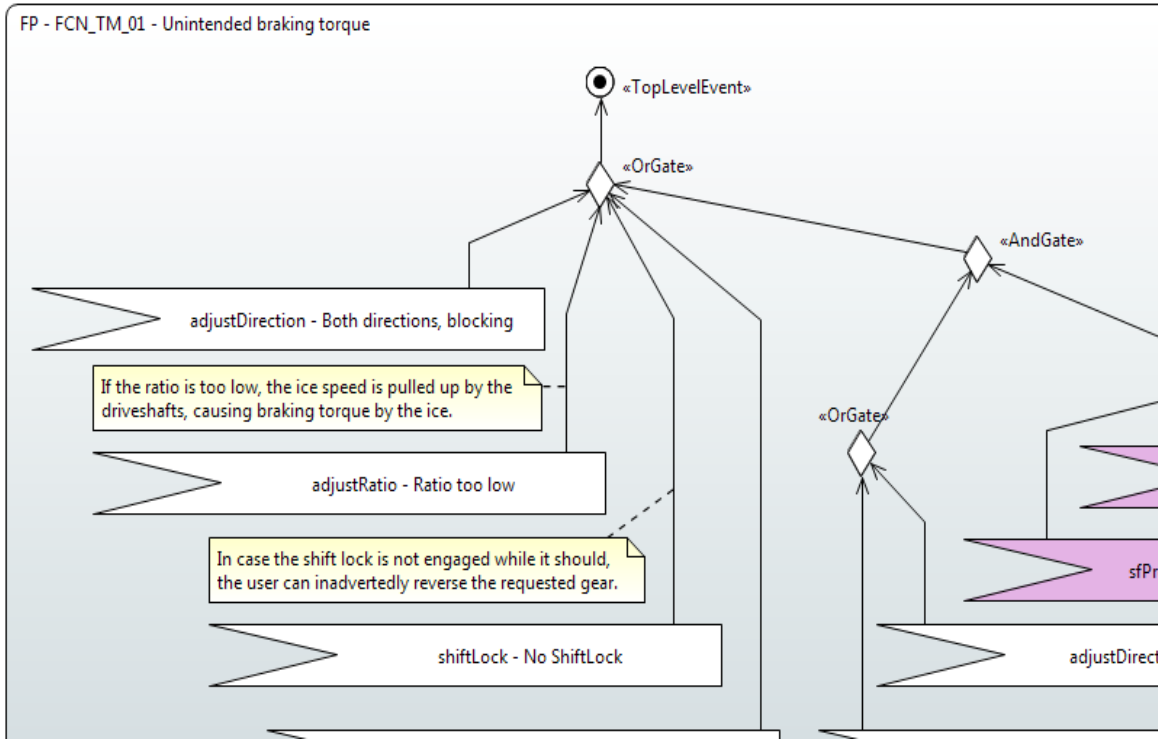
~~****! SysML is useless!~~

Examples
Functional Safety Failures & Propagation

- Required features
  - Function definition
  - Failure definition
  - Failure propagation through system architecture

- How?
  - Not available in SysML
  - Tool smith needs to define
    - Profile
    - New or modified diagram type
    - New user interactions
    - Validations
    - Exploitation for FMEA analysis

→ Coding, maintenance…



Model-based SE: one clear choice

System Engineering? Let's go for OMG SYSTEMS MODELING LANGUAGE SysML™

SysML customizations heavy on creation & maintenance!

~~SysML is a solid base!~~

****! ~~SysML is useless!~~

~~SysML solves everything!~~

A winding road…

General solution available?

- Similar observations for other topics
  - Validation of requirements
  - Design space exploration
  - Design Concept comparison / what-if

- In all cases
  - SysML offers a base
  - Additional steps → additional tooling

- Solution
  - Provide a tool that covers all Systems Engineering functionality
  - Everybody happy….

This is a lie…

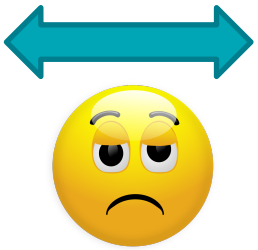
General solution available?

- Observation:
  - SysML is a generic SE language
  - Papyrus is a generic tool supporting that generic SE language
  - Additional tooling adds value
    - **Most valuable tools include a method**
    - **Most valuable tools make assumptions about your model structure**

  E.g.

  – Papyrus-RT = supposes UML-RT method
  – Safety profile = builds on a method for functional safety
  – Allocation completeness checks = needs to know what you want to allocate
  – Concept comparison = depends on the concept generation process

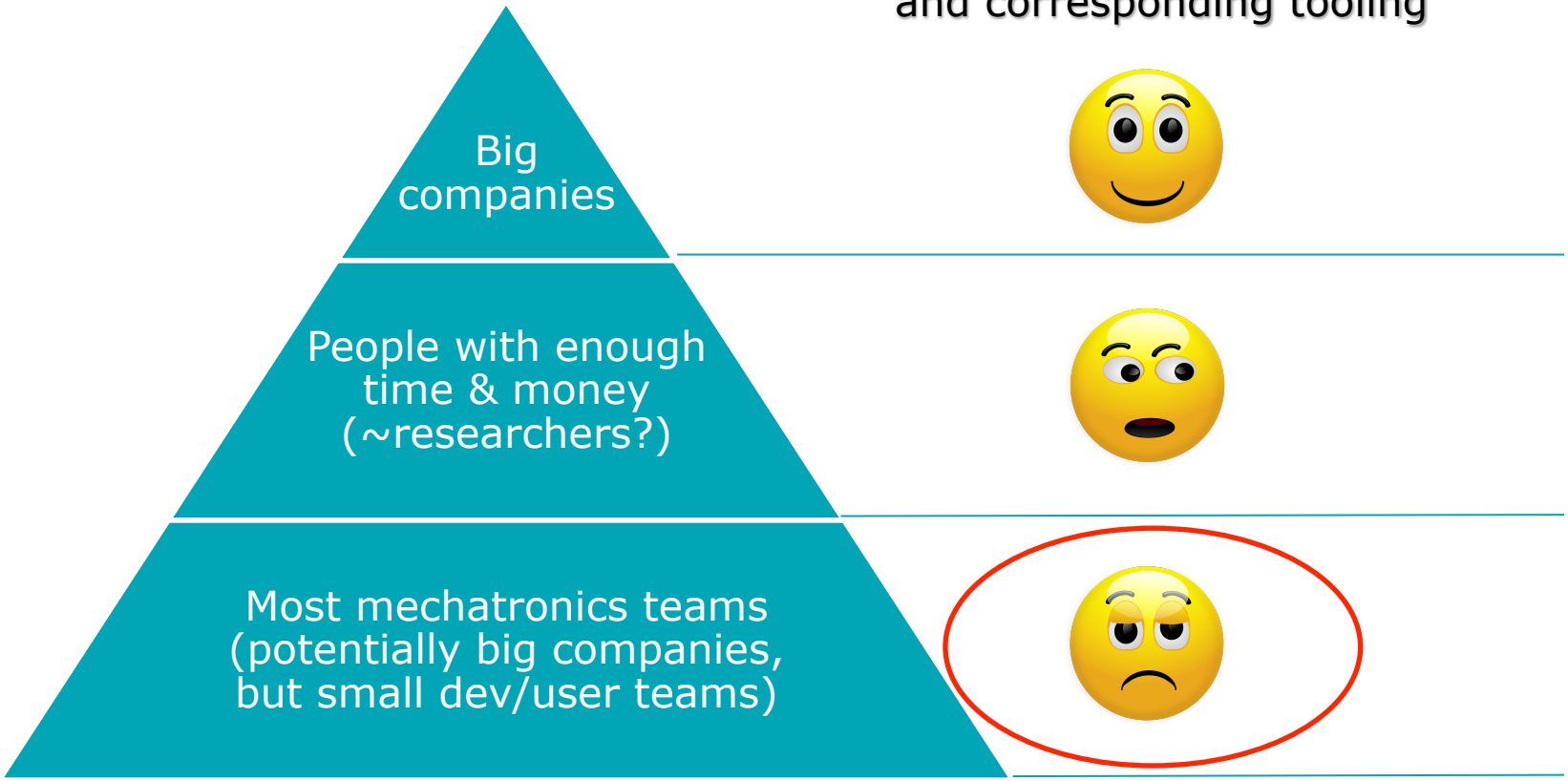
- But… processes/standards are like toothbrushes…
  - No "generally accepted method" exists
  - Each company lives in a different context
  - Most companies struggle with the method (& tool)



1 common SE tooling

Feasibility / adoptability for companies

Ability to create method and corresponding tooling



Challenge…
drowning in complexity of tools / methods

How to straighten the “winding road”?

- SysML?
  - Yeah, good base
  - Lots of unclarity on method / best practices → far from a full solution
- Ease SysML
  - SysML is difficult → simplify it (reduce menus, reduce UML)
  - Ease / streamline typical usage scenarios (~ depends partially on method)
- Introduce reusable method fragments
  - Not a full / strict method
  - Fragments of method + corresponding tooling
    “If you want to …, then … is a good way to do so”
- Allow company to pick an choose
  - Little development
  - Mainly configuration
  - Include guidance

The solution…

- Your options:
  - Wait for a big player to develop method & tool → Swallow
  - Develop method & tool yourself → Drown
  - Collaborate in Papyrus IC on Papyrus for SE → Win

Papyrus IC

	Company 1	Company 2	Company 3	
Feature A	★	★	★	→ Joint development
Feature B	▭	▭	▭	→ Slight adjustment
Feature C	☁	⚡	✹	→ Custom dev
…				

QUESTIONS?

Feel free to contact:
klaas.gadeyne@flandersmake.be
johan.vannoten@flandersmake.be