SysML Model-Based Testing

Fabrice Bouquet

fabrice.bouquet@univ-fcomte.fr
OMG SYMSL - Modelica Group

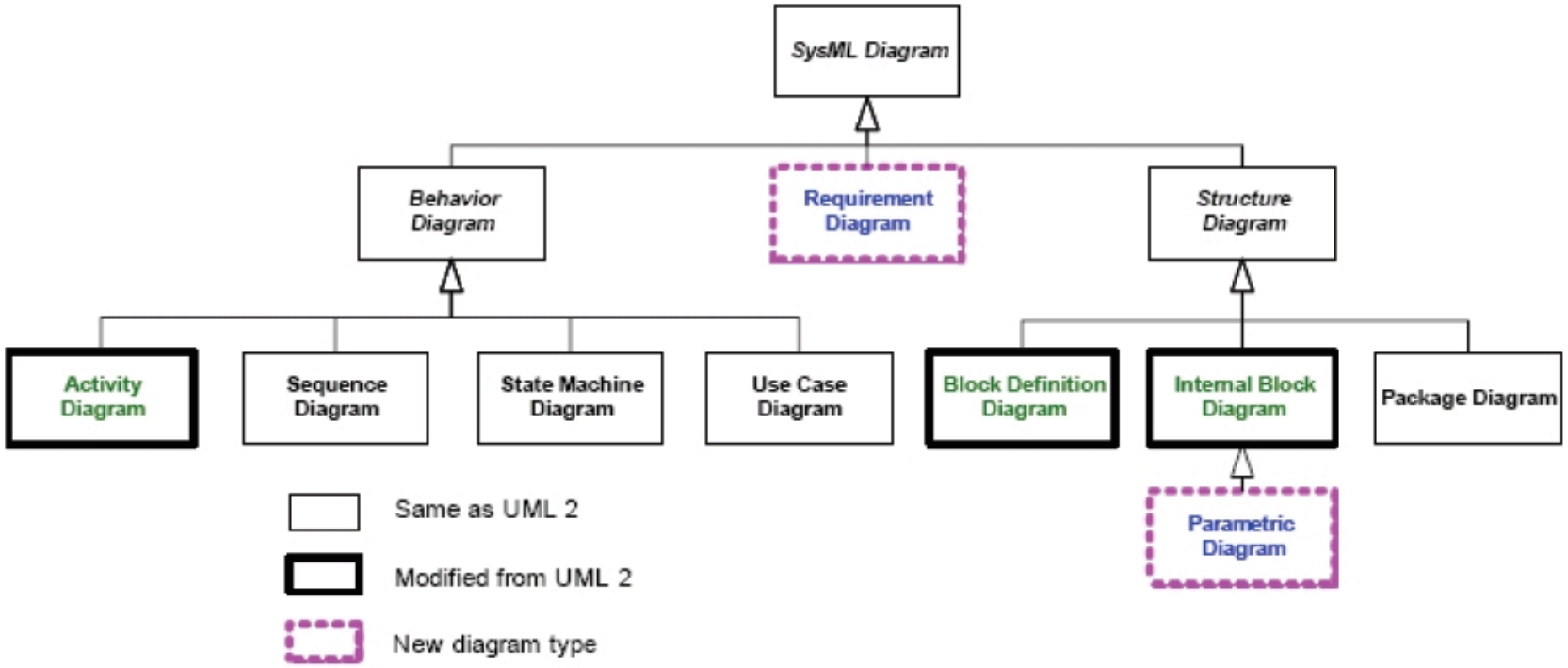
SysML / ALF / OCL / Modelica
March 20th 2013

Outline

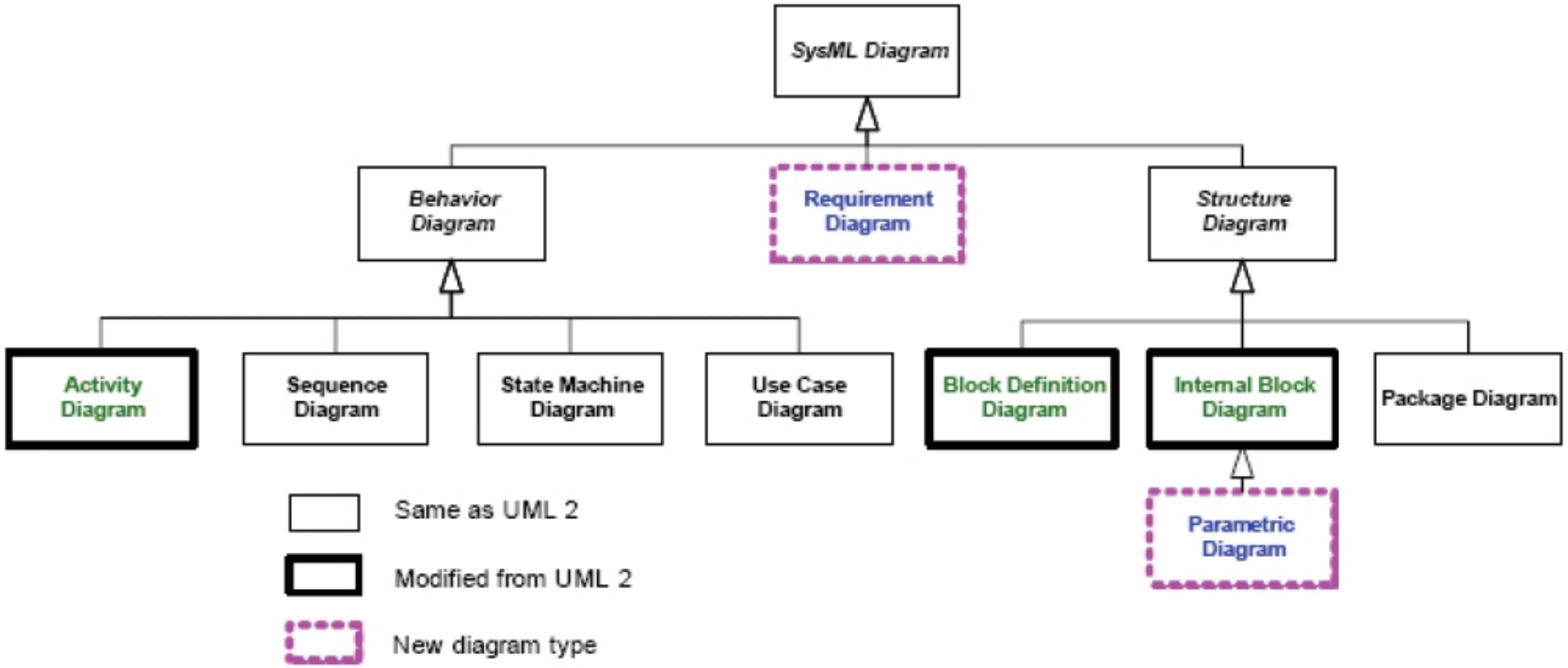
Research Team

- 11 People (2 full Prof., 3 Lecturers, 4 PhD, 2 developers)
- Model-Based Testing (MBT) Domains:
  - Security and Safety i.r. functional
  - IT (UML) and Embedded System (SysML)
- Test Coverage Criteria:
  - Classical:
    - Flow control,
    - Data control,
    - Definition/use of variables
  - Dedicated to SysML:
    - Communication (Signal) Coverage
    - Equation coverage
    - SysML + Marte: time coverage

Language

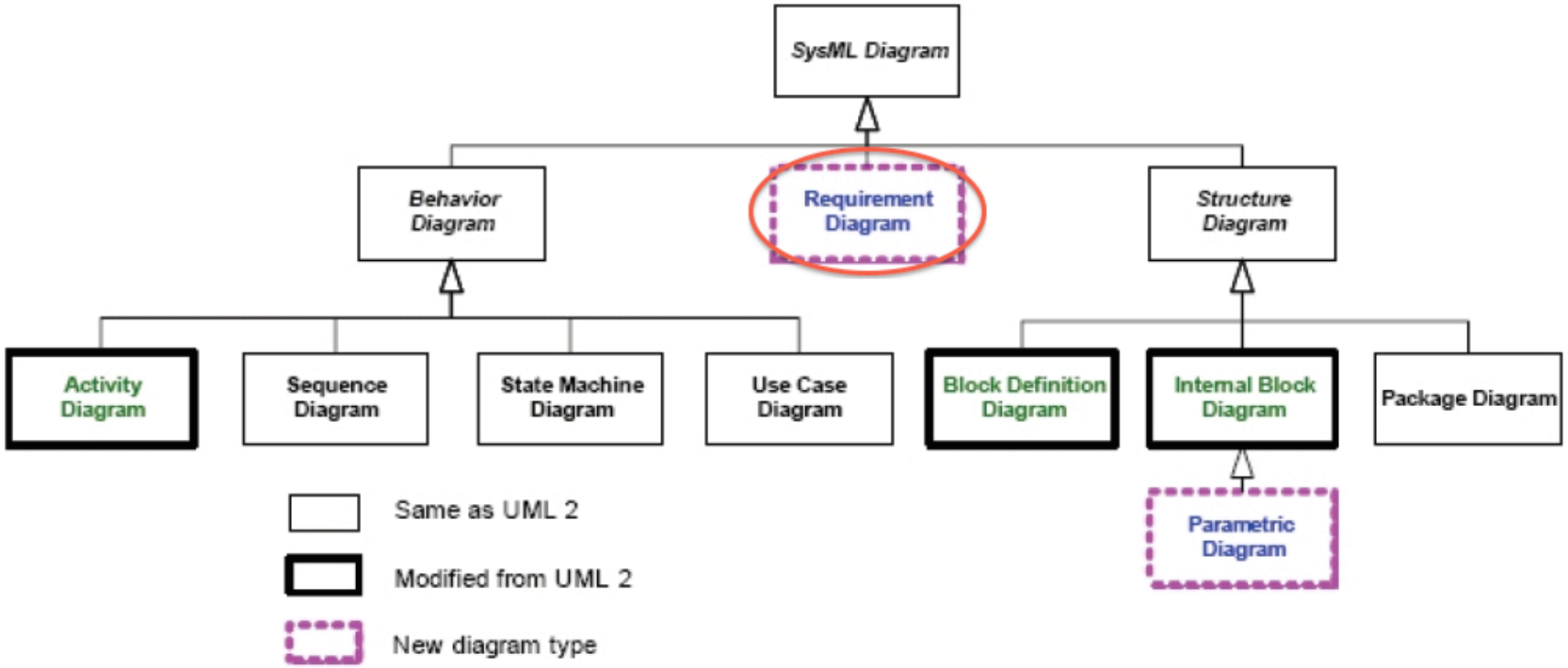

SysML Diagram
Behavior Diagram
Requirement Diagram
Structure Diagram
Activity Diagram
Sequence Diagram
State Machine Diagram
Use Case Diagram
Block Definition Diagram
Internal Block Diagram
Package Diagram
Parametric Diagram
Same as UML 2
Modified from UML 2
New diagram type

Language



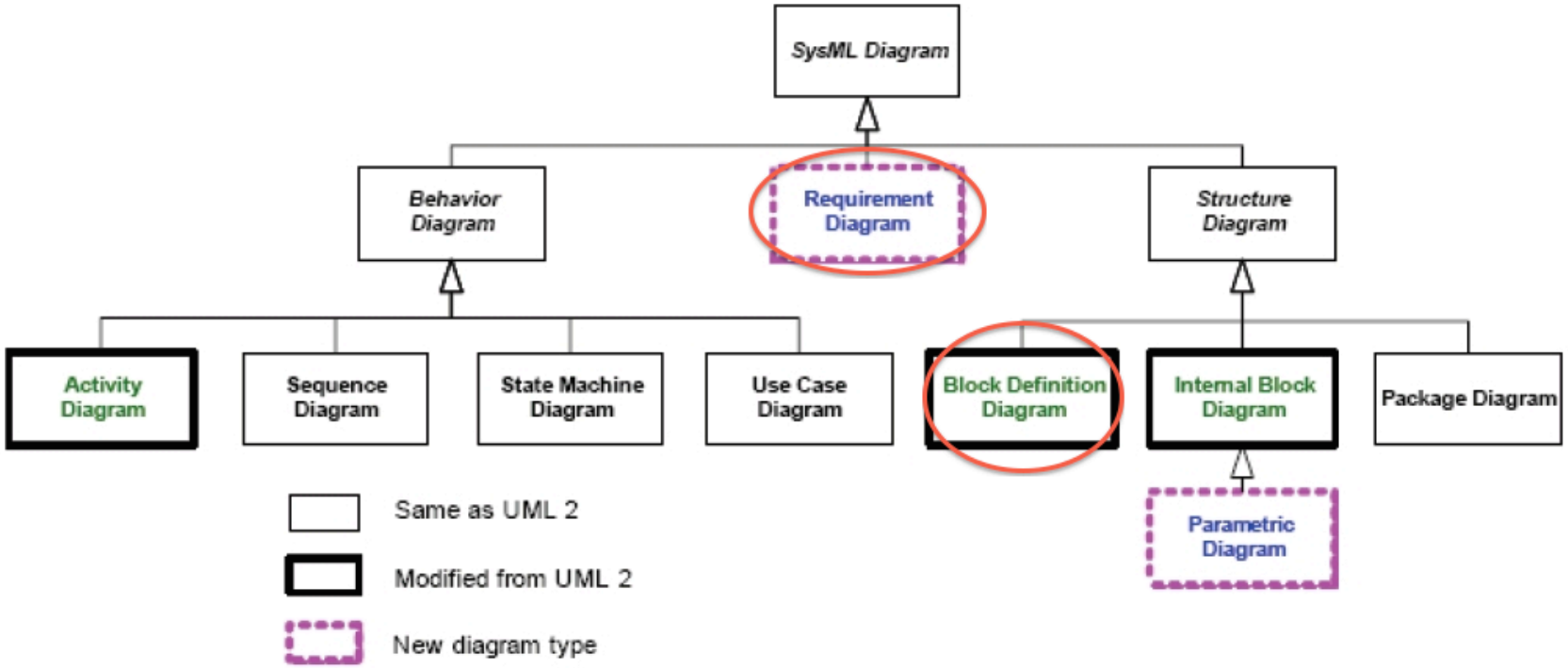
with OCL and ALF

Language



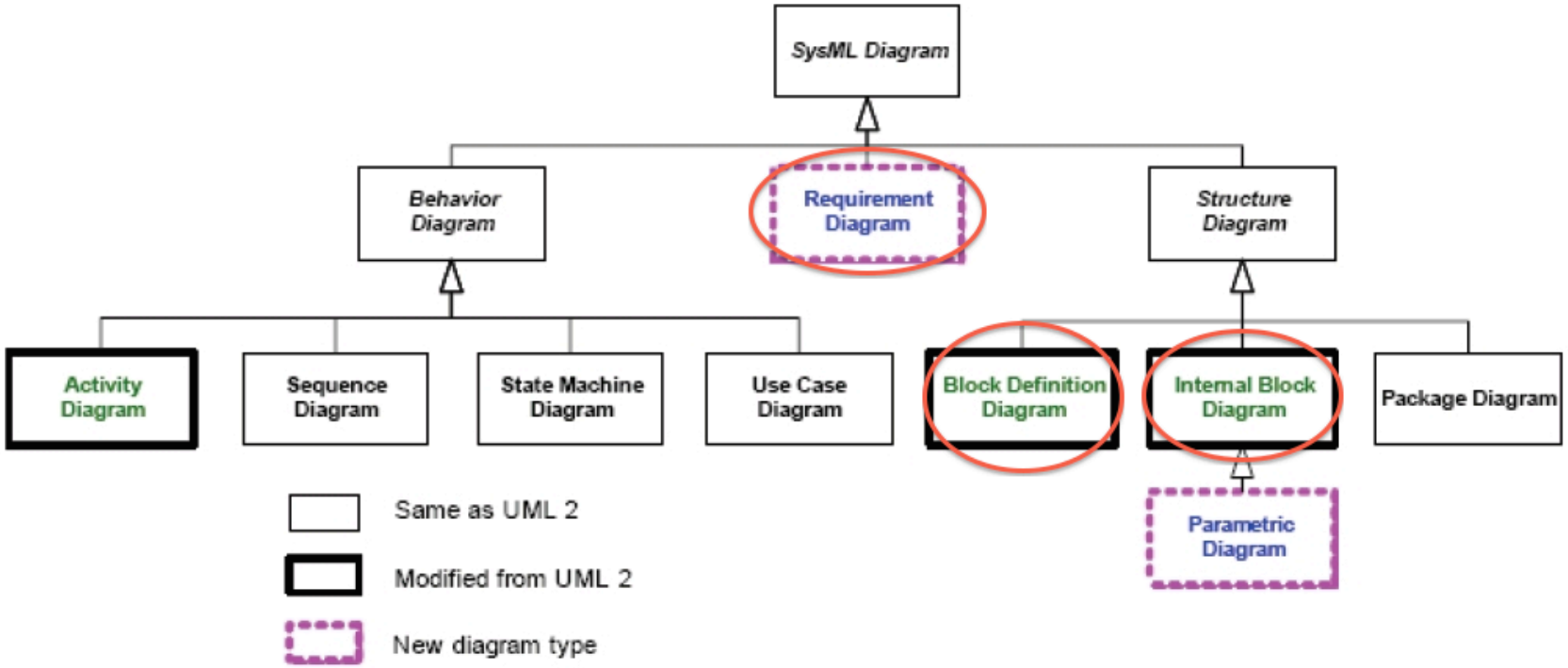
with OCL and AFL

Language



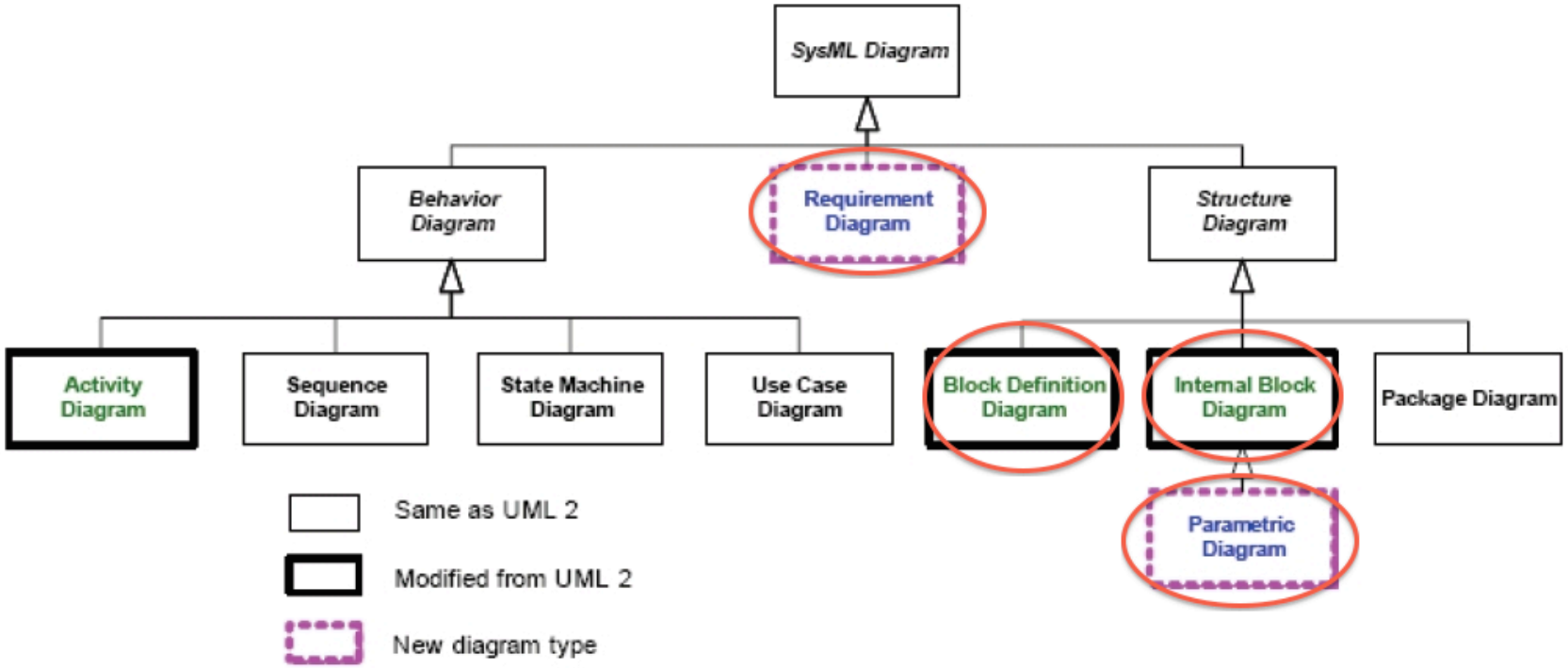
with OCL and AFL

Language



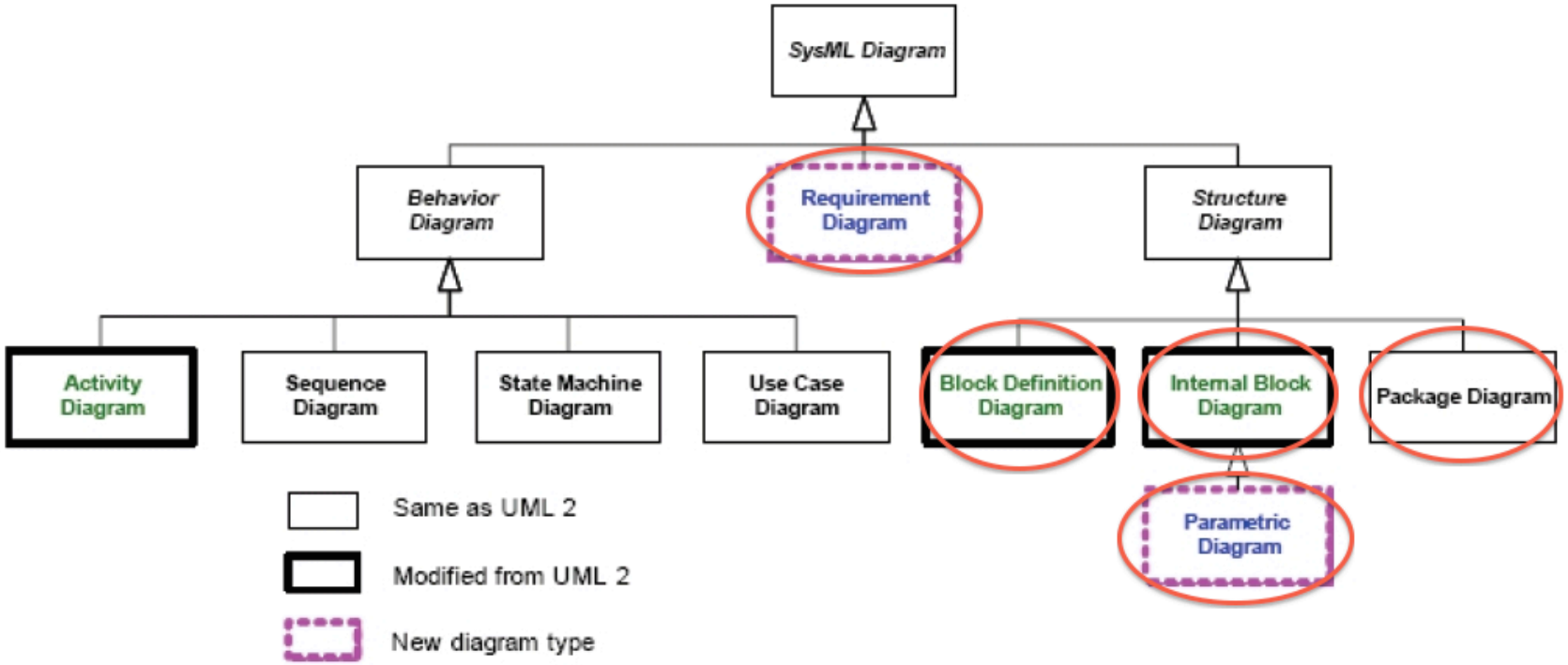
with OCL and AFL

Language



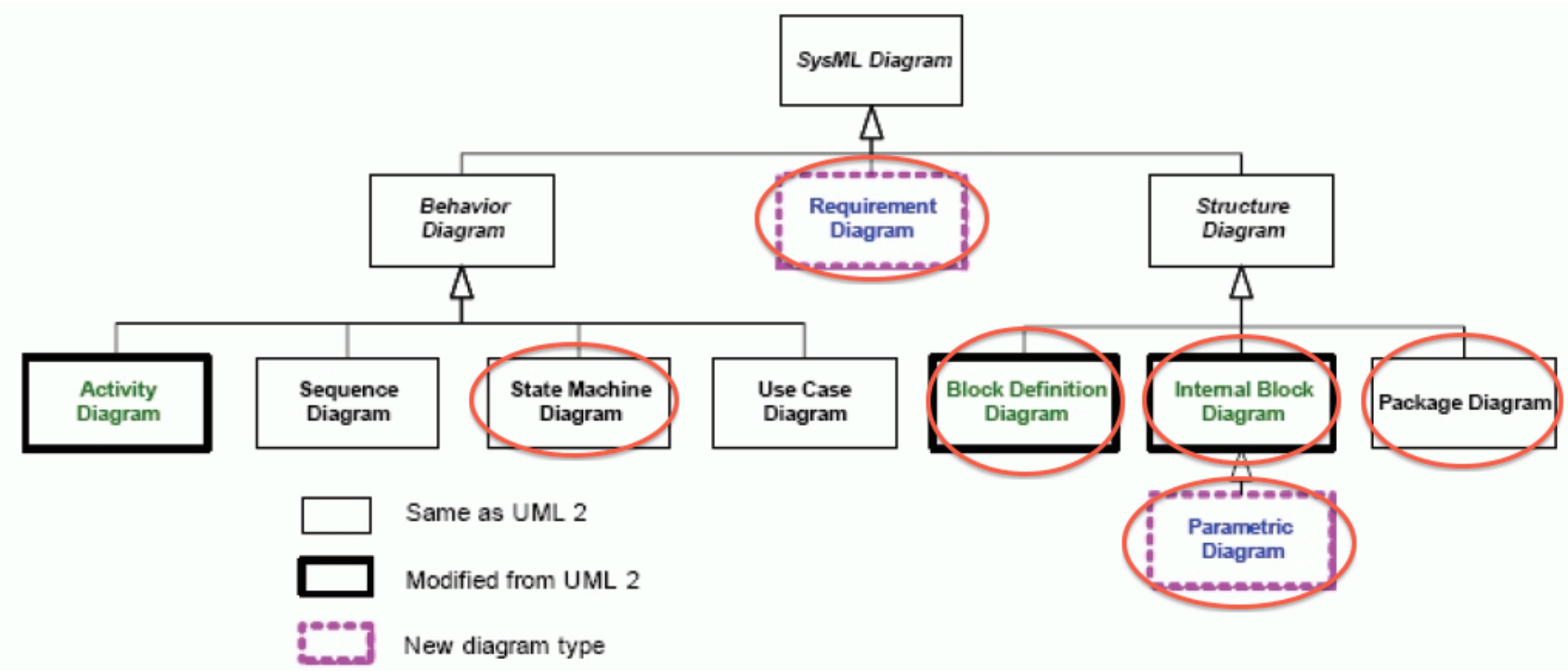
with OCL and AFL

Language



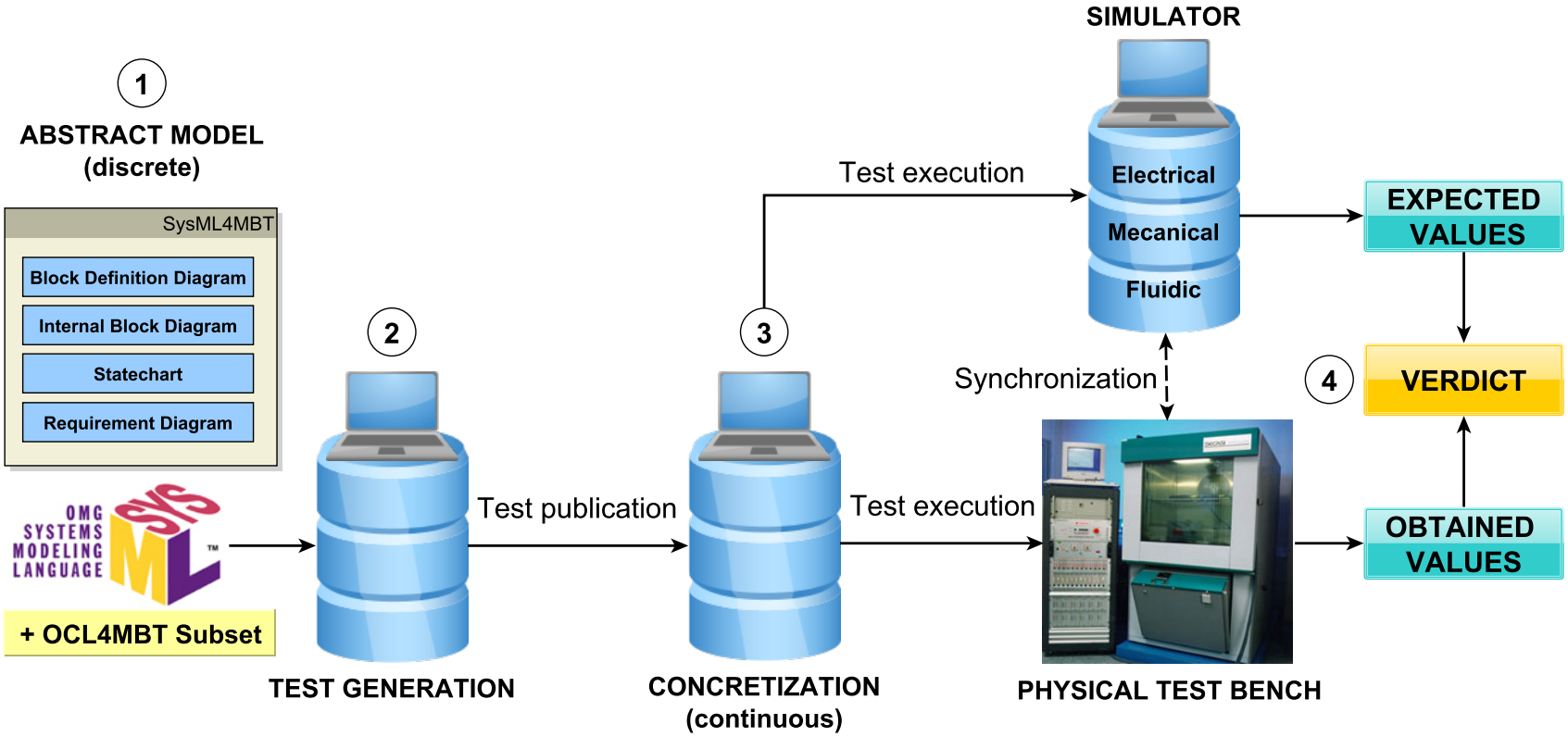
with OCL and AFL

Language



with OCL and AFL

Context: existing MBT tool chain



VETESS Project: `http://lifc.univ-fcomte.fr/vetess`

Context: tool chain analysis

Strengths

- Relevant to generate many test cases
- Ensures an optimal coverage of the model
- Reasonable time of test generation
- Successful feedback from industrial engineers

Weaknesses

- Creation of both SysML model and simulation model
  $\Longrightarrow$ Late functional validation of the SysML model
- Discrete model of the system
  $\Longrightarrow$ Concretization step very costly (+ 50% of time spent during concretization)

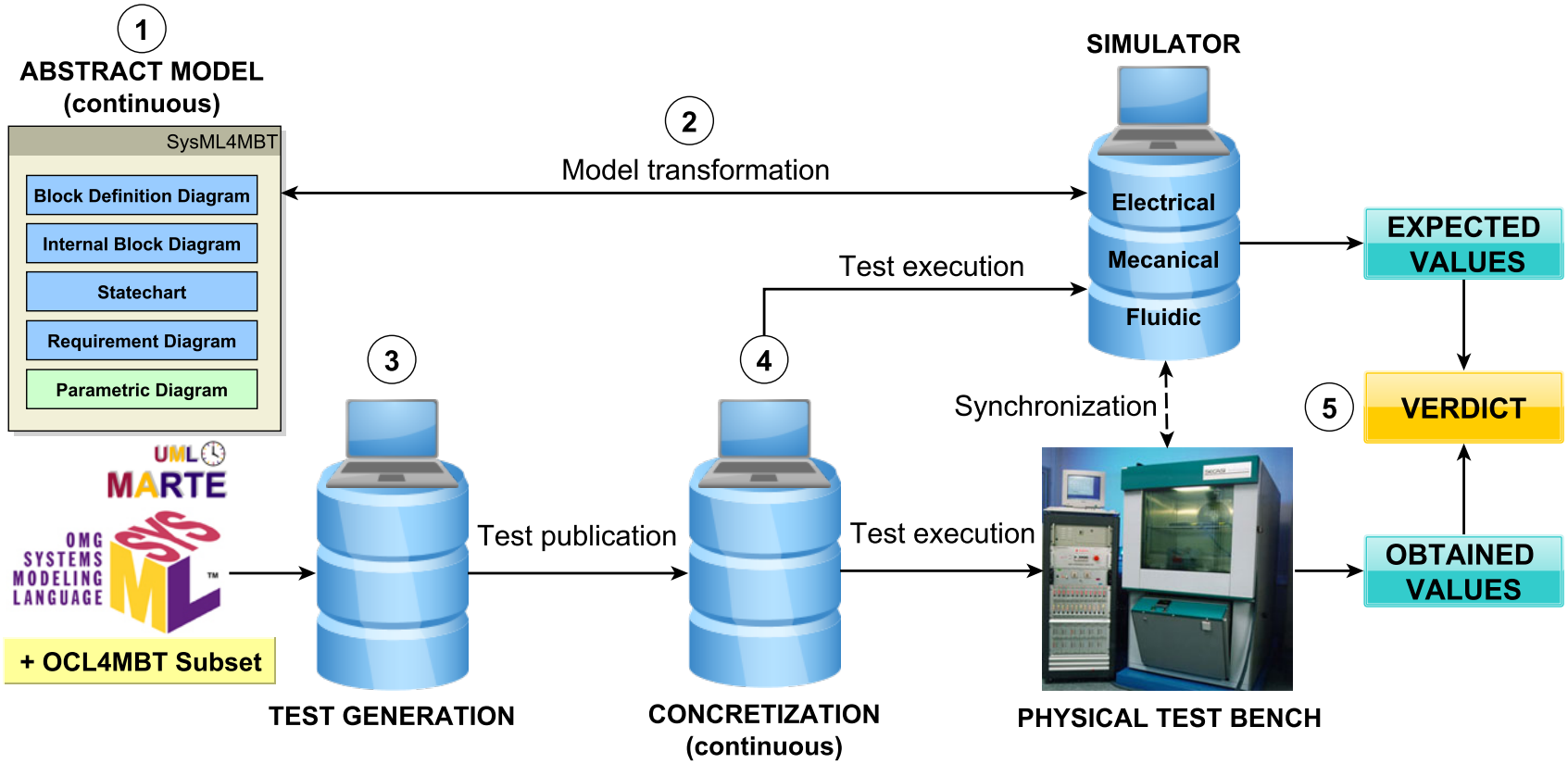
Motivations

Validation of Real-Time and Embedded Systems using MBT

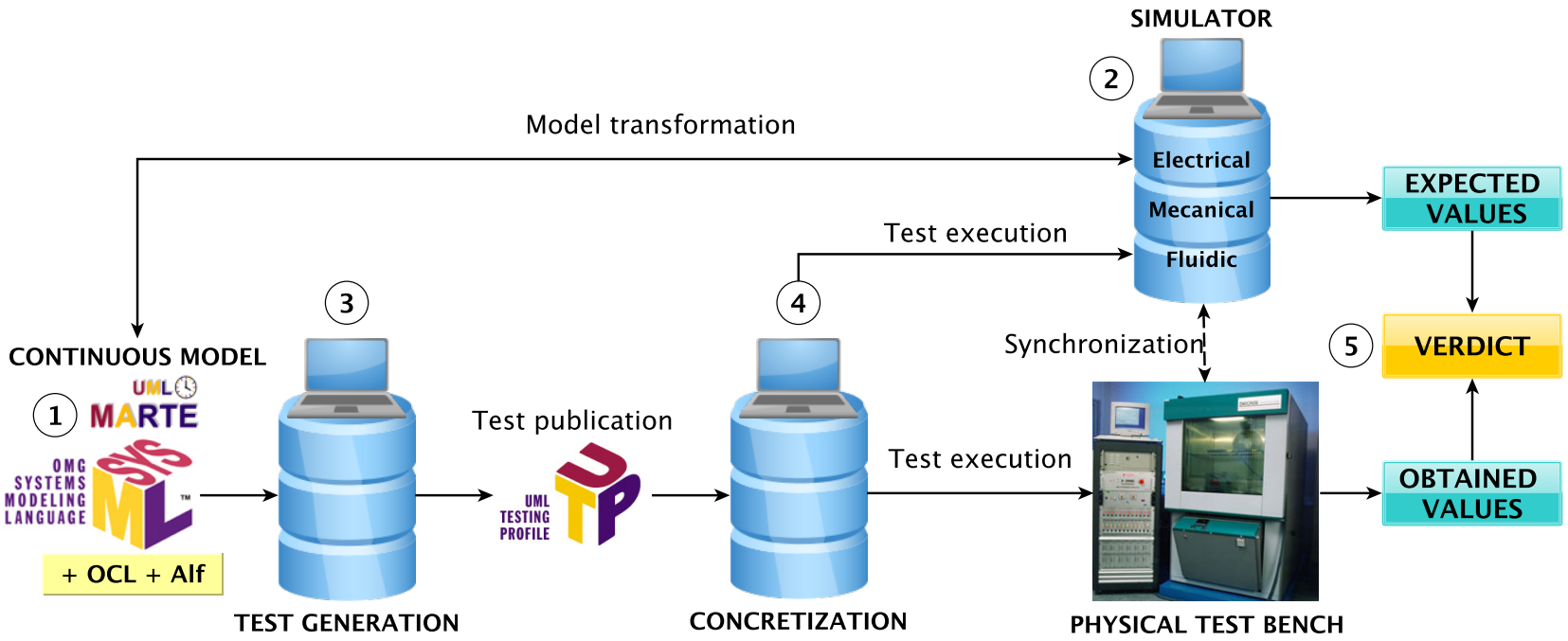
- Improvements of an existing MBT tool chain:
  - Extension for real-time and embedded systems
  - Use of simulation to validate the test model
  - Automation of the end-to-end MBT process
- Use of standardized languages and MDE techniques:
  - SysML/MARTE profiles
  - Generate element for simulation code (VHDL-AMS, Modelica)
  - UTP as a pivot language

Outline

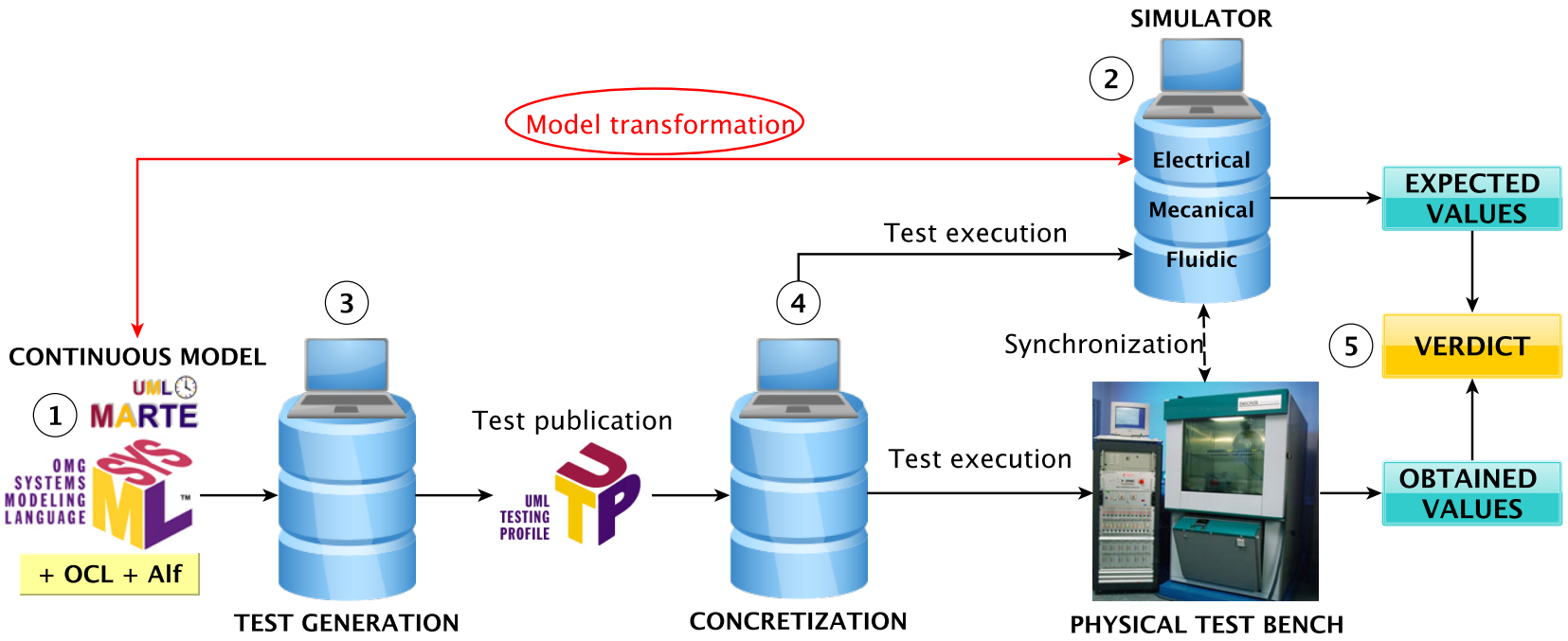
Current Works



Current Works

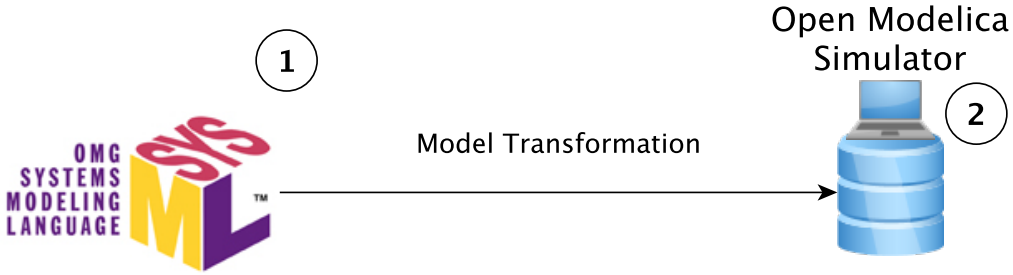


Current Works



Outline

Modeling Consistency: contributions

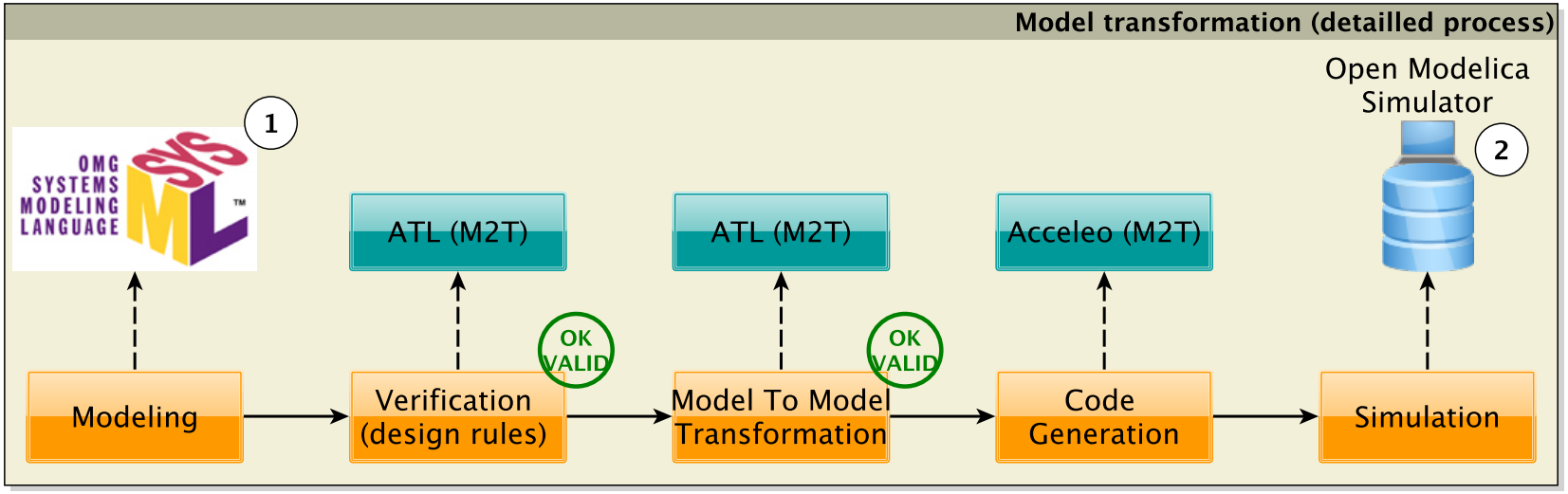


- How to generate correct Modelica code?
- How to provide a SysML modeling guideline?
- How to automate these issues?

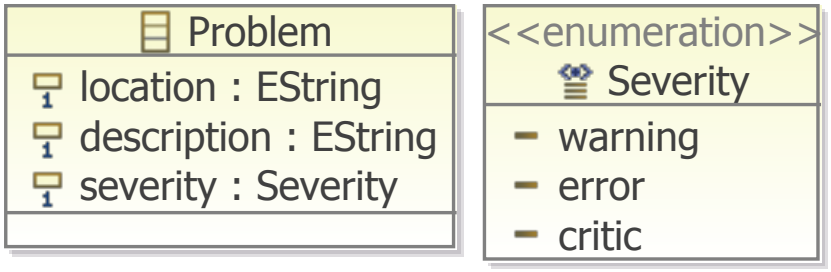
⟹ Rules definition for SysML model consistency:
  - Syntactic verification
  - Semantic verification

⟹ Automation based on MDE techniques:
  - ATL for Model to Model Transformation
  - Acceleo for code generation

Modeling Consistency



Modeling Consistency Verification



Properties | Problems | Console | Synchronize

6 errors, 26 warnings, 1 other

Description	Resource
Errors (6 items)	
Flow port 'Nout' typed with 'Fluidic' without importing the corresponding package: IEEE::fluidic_systems	Level_0.sysml
Property 'APSInstance' typed with 'C' without importing the corresponding package: SI Definitions::SI Value Types	Level_0.sysml
Property 'Atype' is not typed	Level_0.sysml
The block 'APS' contains parts that are not typed with a block	Level_0.sysml
The connector 'Connector15' link two flow ports that are not typed with the same type	Level_0.sysml
The connector 'Connector20' link two flow ports that are not typed with the same type	Level_0.sysml
Warnings (26 items)	
Infos (1 item)	

ATL Verification Rule

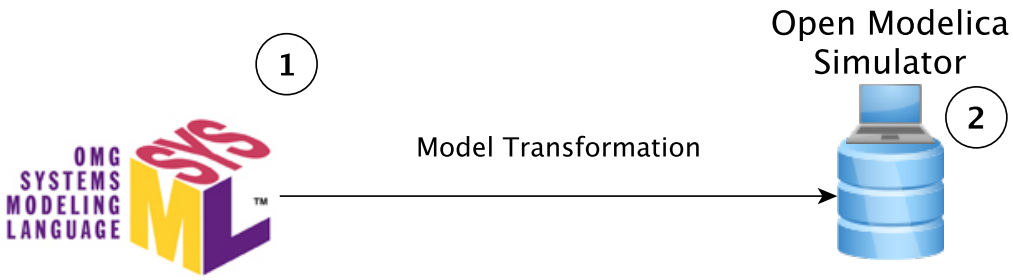
```
helper def: reservedWords: Sequence(String) =
  Sequence{'ABS', 'ACCESS', 'AFTER', 'ALIAS', [...], 'TOLERANCE'};

 helper context MMuml!NamedElement def: isReservedWord() : Boolean =
  if self.name <> OclUndefined then
   thisModule.reservedWords->exists(r | r.toString().toLower() = self.name.toString().toLower())
  else
   false
  endif;

rule uncorrectNameReservedWord{
 from umlElement: MMuml!NamedElement(
  umlElement.isReservedWord()
 )
 to problem: MMproblem!Problem(
  severity <- #error,
  description <- umlElement.name + ' is a reserved word for Modelica',
  location <- 'root::' + umlElement.getQualifiedName()
 )
}
```

Outline

Model Transformation Validation



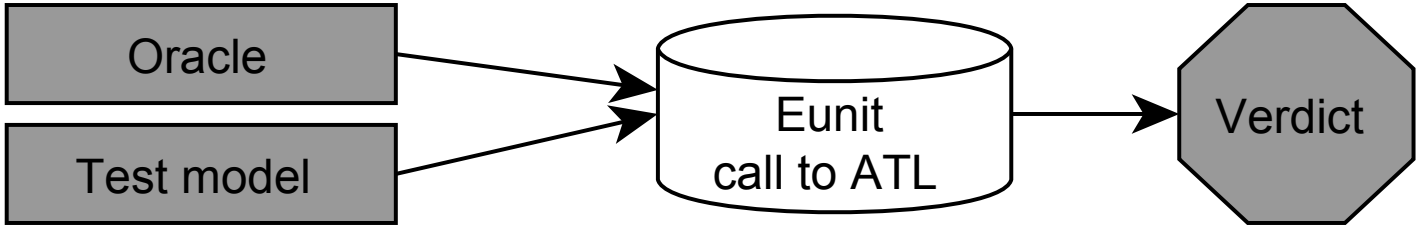
- How to validate the transformation process?
- How to detect bugs during the transformation development?

⟹ Test-Driven Development strategy:

- Use of unit tests
- Use of the EUnit framework

EUnit Framework

- Epsilon Unit Testing Framework
- Based on Eclipse EMF
- Compares transformation result with expected result
- Can be integrated as an automated task

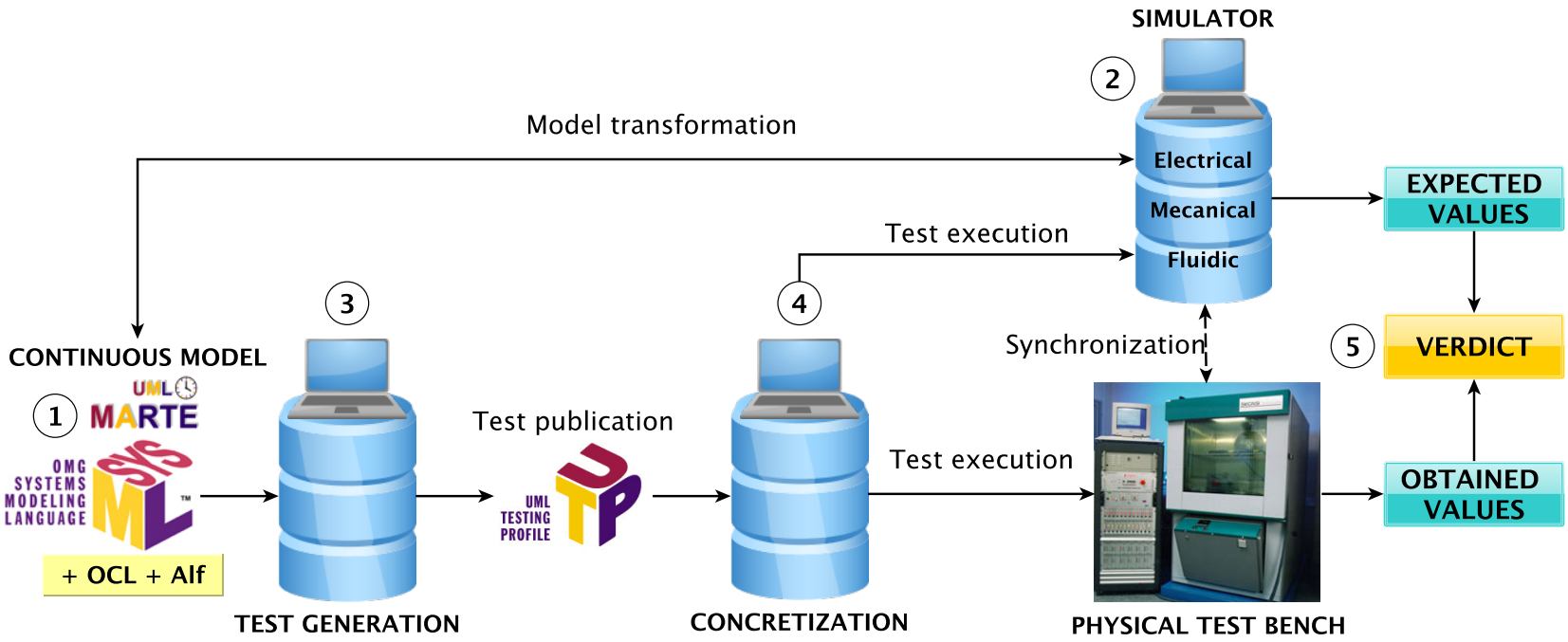


Results

- SysML Consistency verification:
  - Detects Modelica syntactic/semantic errors
  - Allows SysML modeling guideline for Modelica generation
    - 31 ATL rules
    - 24 ATL helpers
- Model Transformation Validation:
  - *SysML2Problem* transformation $\Longrightarrow$ 59 unit tests
  - *SysML2Modelica* transformation $\Longrightarrow$ 41 unit tests:
    - 34 ATL rules
    - 23 ATL helpers

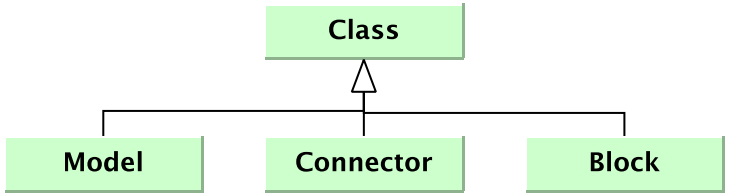
Outline

Overview


SIMULATOR
2
Model transformation
Electrical
Mecanical
Fluidic
EXPECTED VALUES
Test execution
3
4
Synchronization
5
VERDICT
CONTINUOUS MODEL
1
MARTE
OMG SYSTEMS MODELING LANGUAGE
+ OCL + Alf
Test publication
UML TESTING PROFILE
Test execution
OBTAINED VALUES
TEST GENERATION
CONCRETIZATION
PHYSICAL TEST BENCH

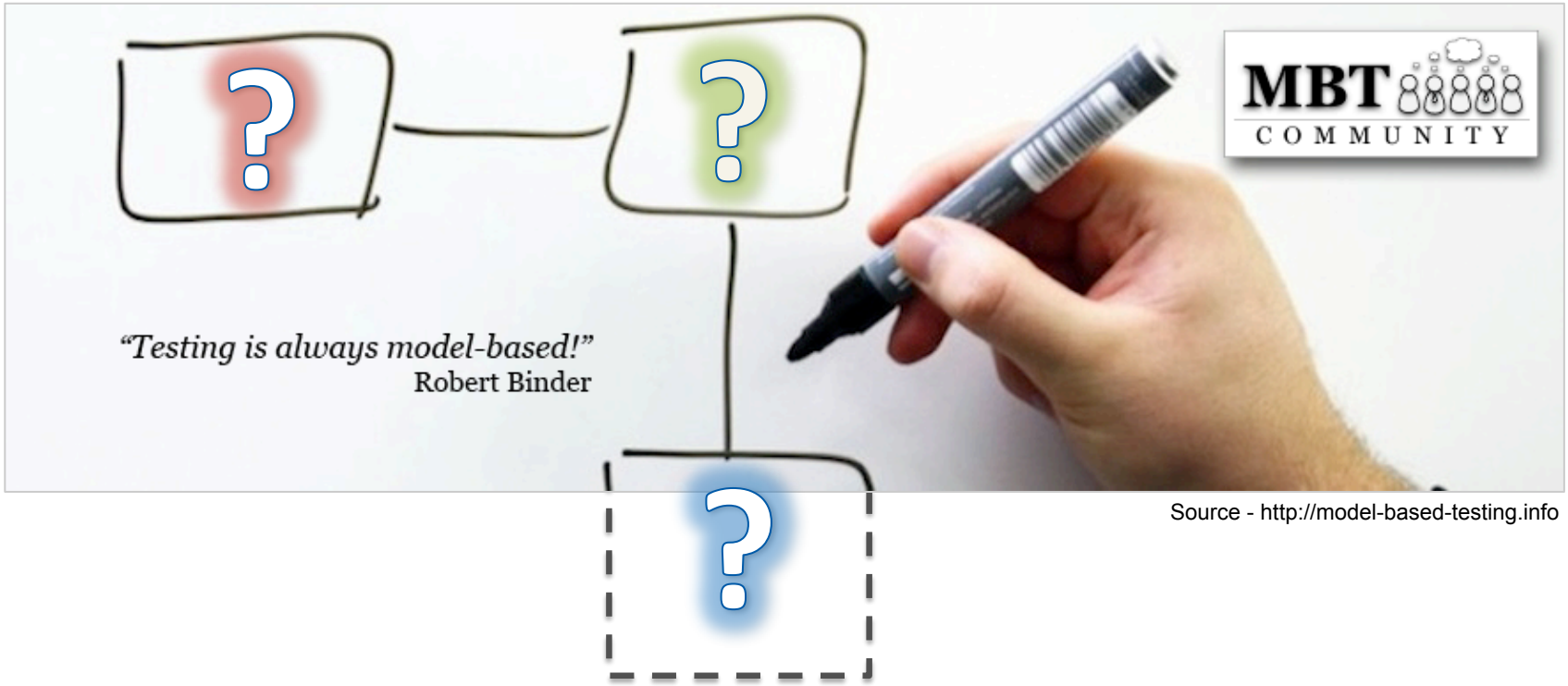
Conception Choice

- SysML and (Modelica) Connector?
  - Connector as Stereotype for Block Vs Flow Specification
  - Use Stereotype on flow properties
- For a subset with only "Model" can be optional?



- Tool for simulation of Modelica?
  - Interactive simulation
  - documented API

Thank you for your attention



Source - http://model-based-testing.info

Research supported by:

- SyVAD Project: `http://syvad.univ-fcomte.fr/`
- Smart Blocks Project: `http://smartblocks.univ-fcomte.fr/`
- Labex Action: `http://www.labex-action.fr`