THE NEED FOR TEST SPECIFICATION LANGUAGES

Ina Schieferdecker

SOPHIST DAYS 2017, Nuremberg, October 24, 2017

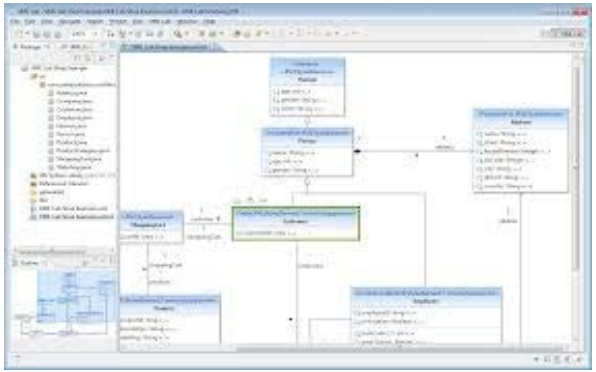
SOPHIST DAYS 2017

Fraunhofer FOKUS

OR: WHERE IS THE TRUTH?

In the code ?

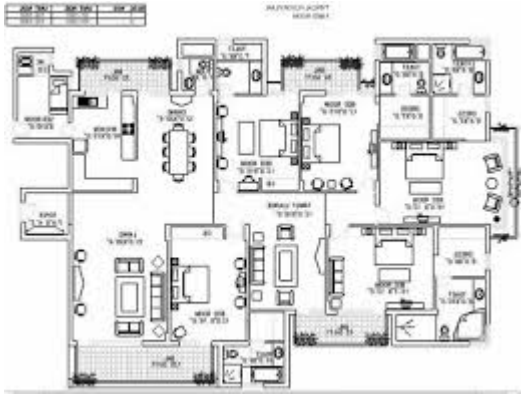
In the model ?



HOW IS IT IN OTHER DISCIPLINES?

In the house ?

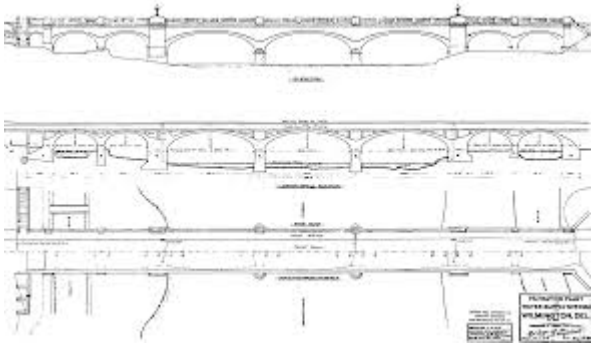
In the model ?



HOW IS IT IN OTHER DISCIPLINES?

In the bridge ?

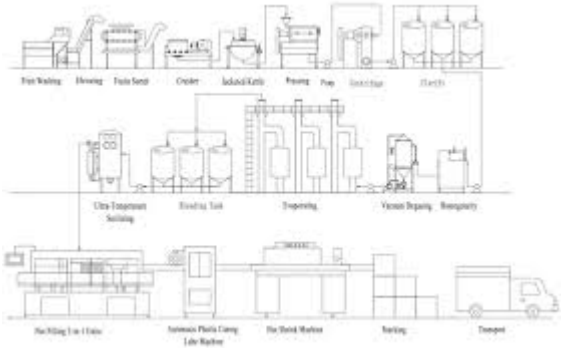
In the model ?



HOW IS IT IN OTHER DISCIPLINES?

In the production line ?

In the model ?



THE SOFTWARE MODEL …

is the digital twin of software!

OUTLINE

1. About me

2. Do we need test specification languages ?

3. What is needed in addition to a test specification language ?

4. Where should we aim at next ?

ABOUT ME: TELECOMMUNICATION

1. Interoperability

   Interoperability is the ability of making systems and organizations to work together (inter-operate). While the term was initially defined for information technology or systems engineering services to allow for information exchange, … [Wikipedia]

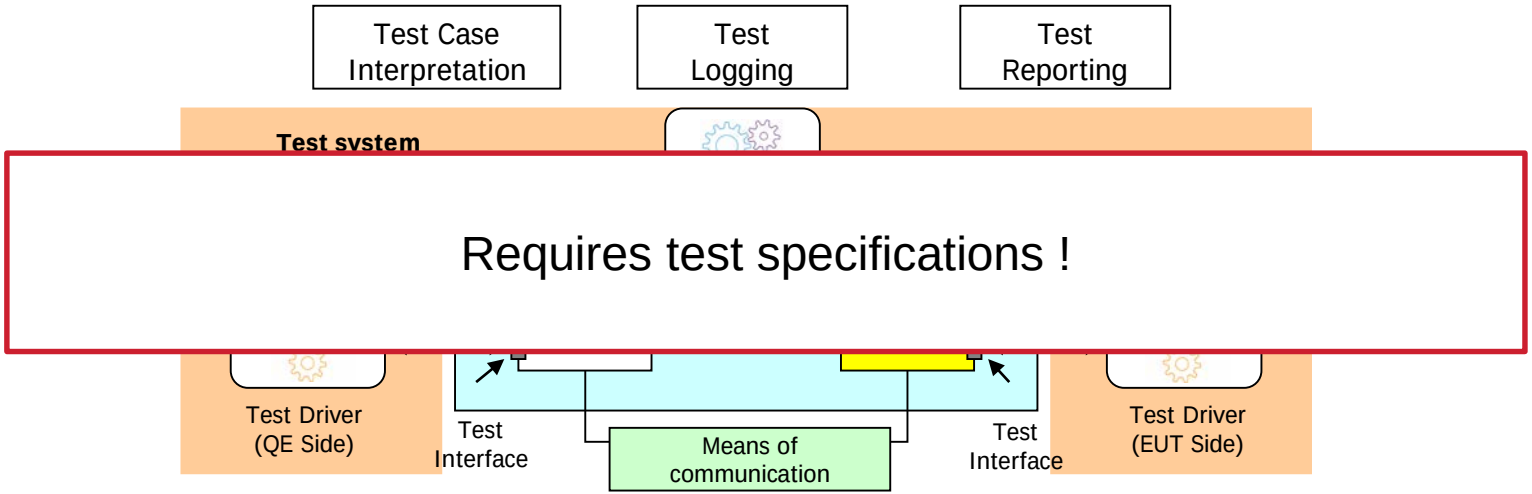
2. Conformance

   Confirmation that a good, service, or conduct meets the requirements of legislation, accepted practices, prescribed rules and regulations, specified standards, or terms of a contract. [Business Dictionary]

Interoperability is a precondition for the increasing integration and networking of systems and components.
Conformance supports interoperability.

ABOUT ME: INTEROPERABILITY TESTING

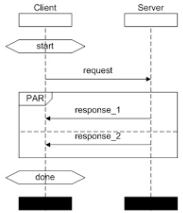
- A dynamic testing method
- Complements conformance testing



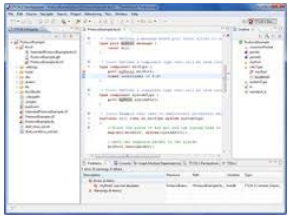
1. QE = Qualified Equipment (previously tested)
2. EUT = Equipment under Test (such as gateway, protocol layer, software component)

ABOUT ME: WORKING ON TEST SPECIFICATION LANGUAGES

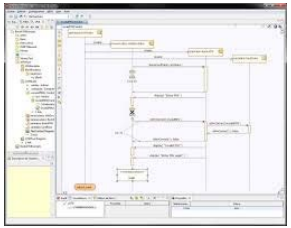
Message Sequence Charts



Testing and Test Control Notation



UML Testing Profile



LOOKING BACK AND FORWARD

Do we need … standardized … test specification languages ?

THE WORD WAS IN THE BEGINNING

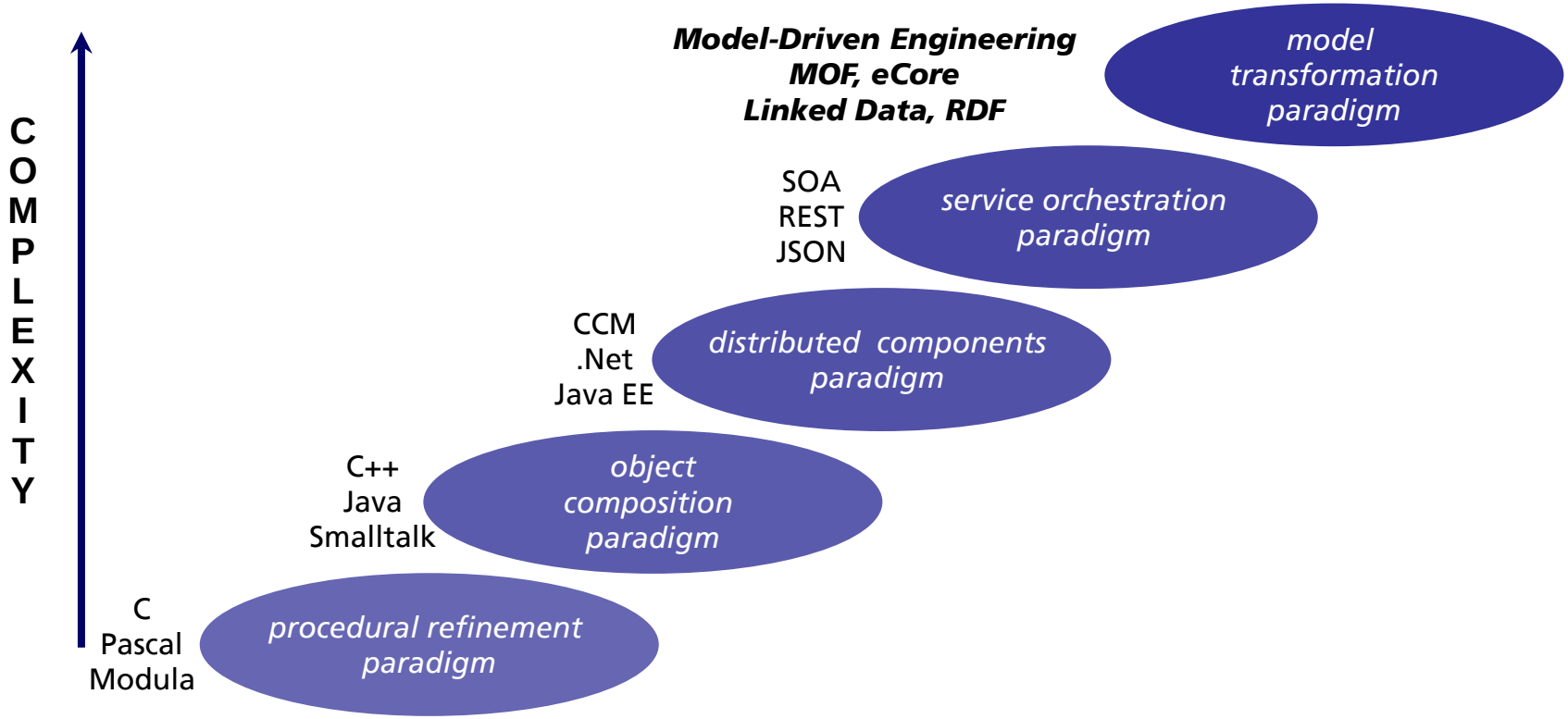
And the word comes with

- A common understanding
  - A language

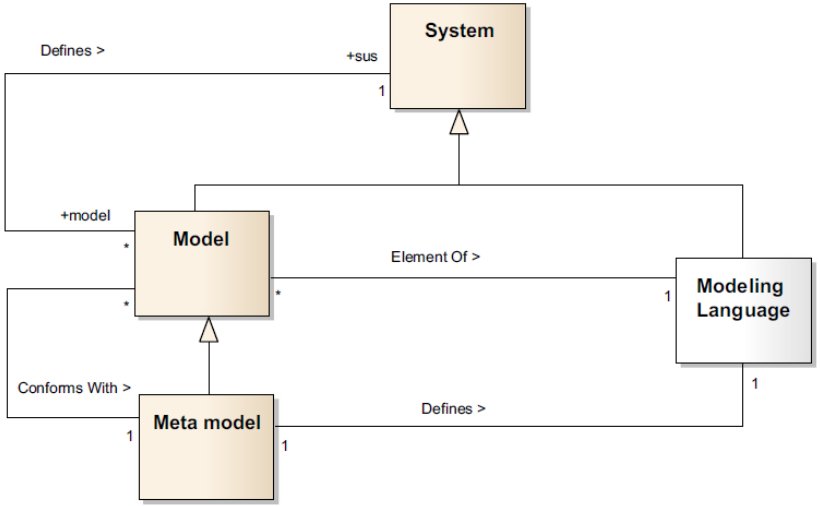
Any software architecture defines a „language“ !

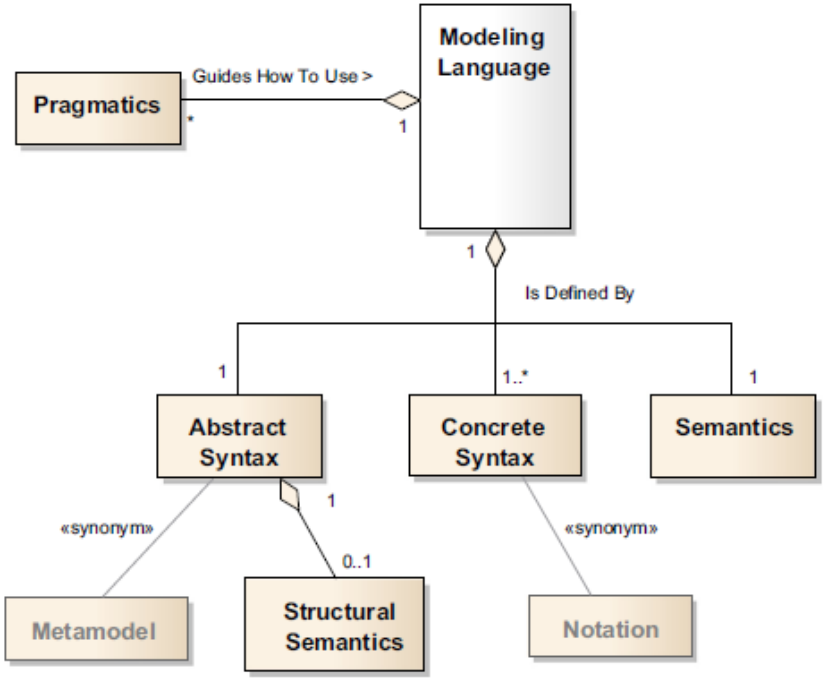
- A mind set
  - A culture
    - A set of methods and processes

EVOLUTION OF SOFTWARE ENGINEERING



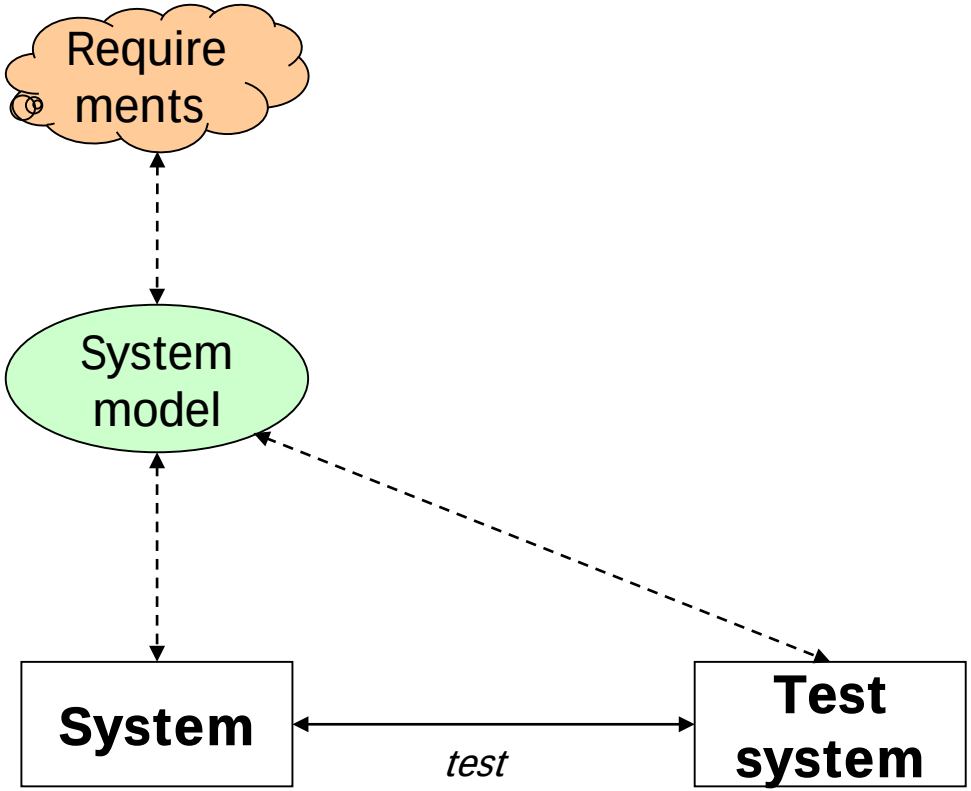
MODELING LANGUAGES



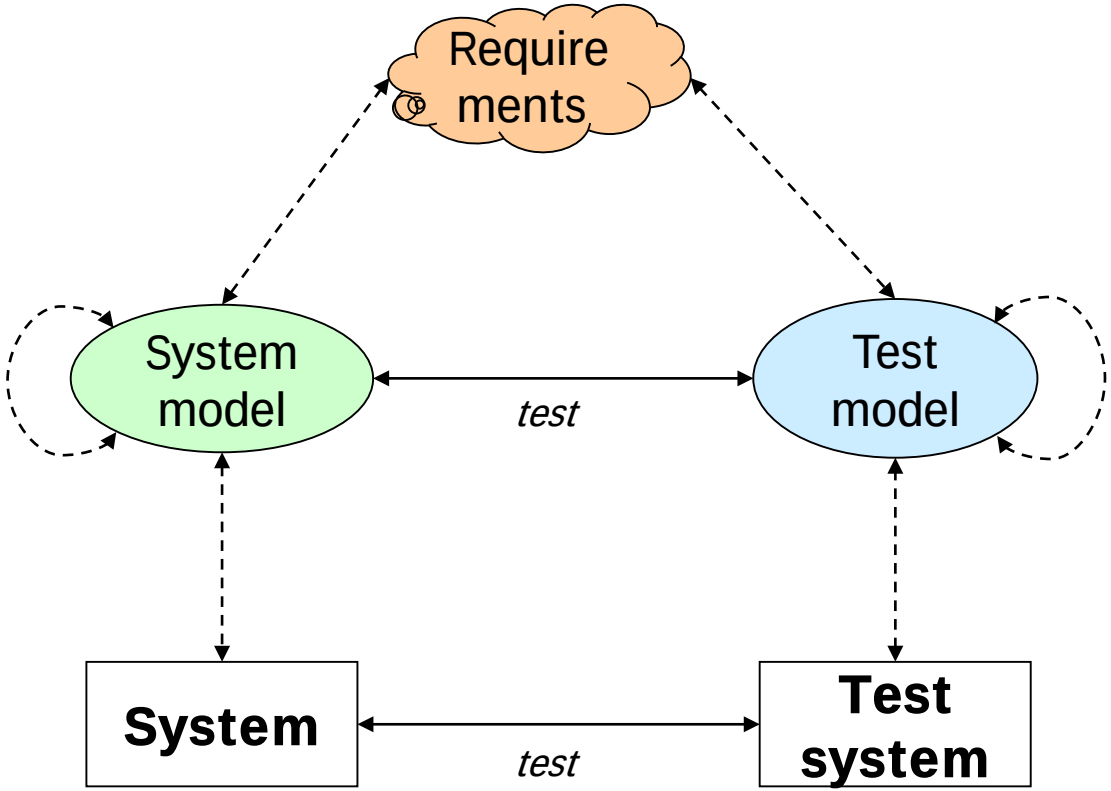


Taken from Alberto Rodrigues da Silva: Model-driven engineering: A survey supported by the unified conceptual model, Computer Languages, Systems & Structures. 43(2015)139–155.

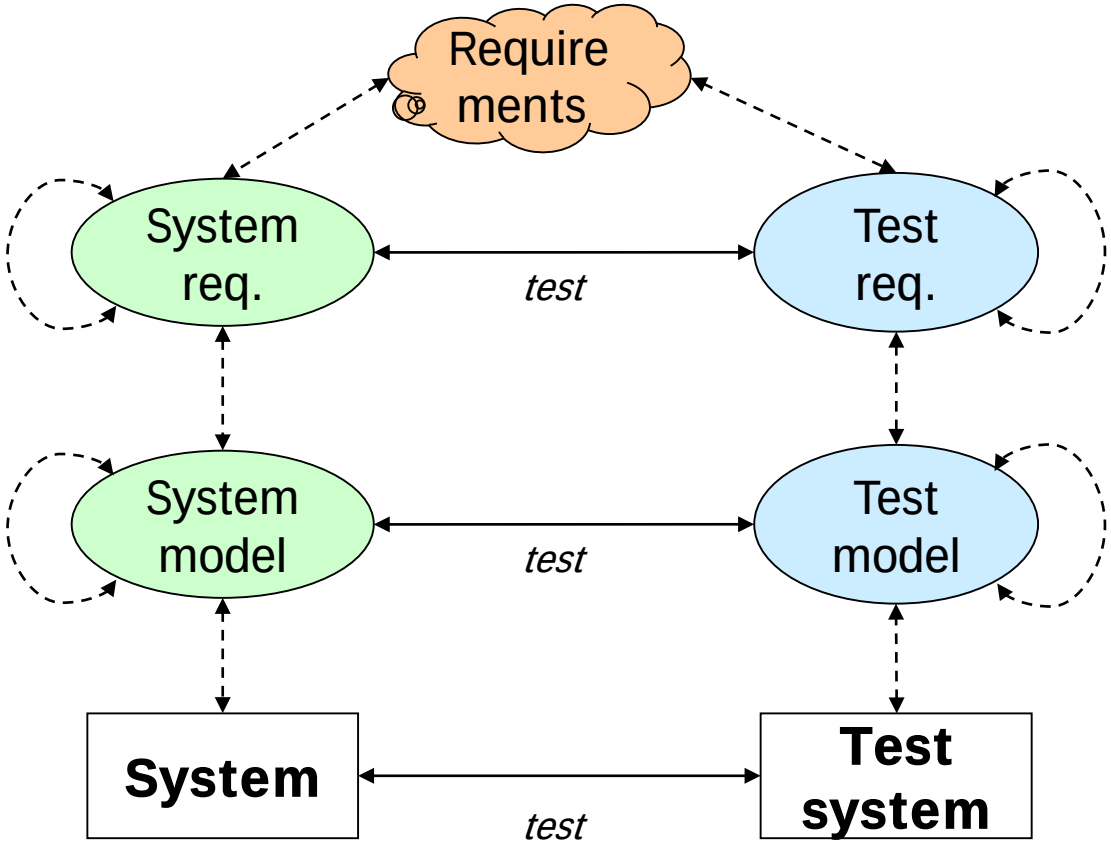
MBT 1.0



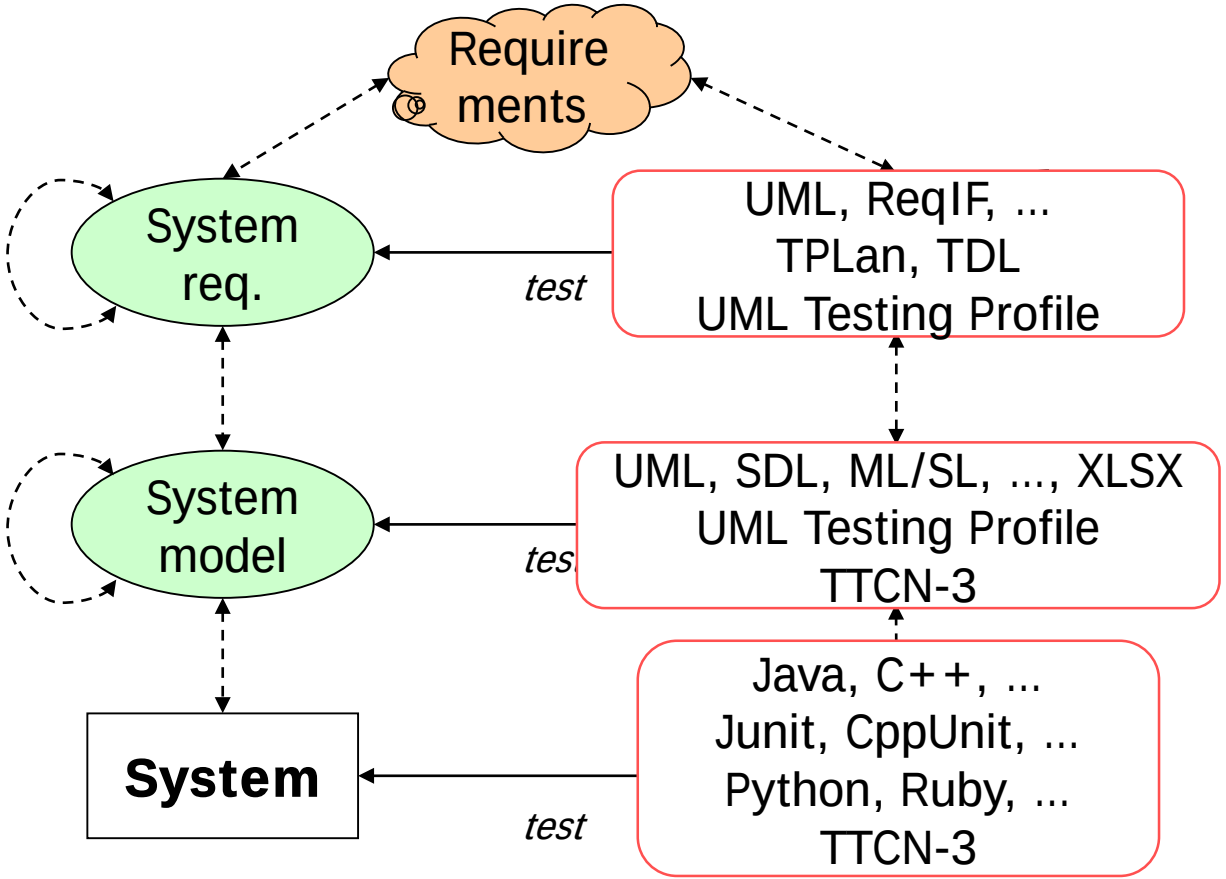
MBT 2.0


Require
ments
System
model
test
Test
model
System
test
Test
system

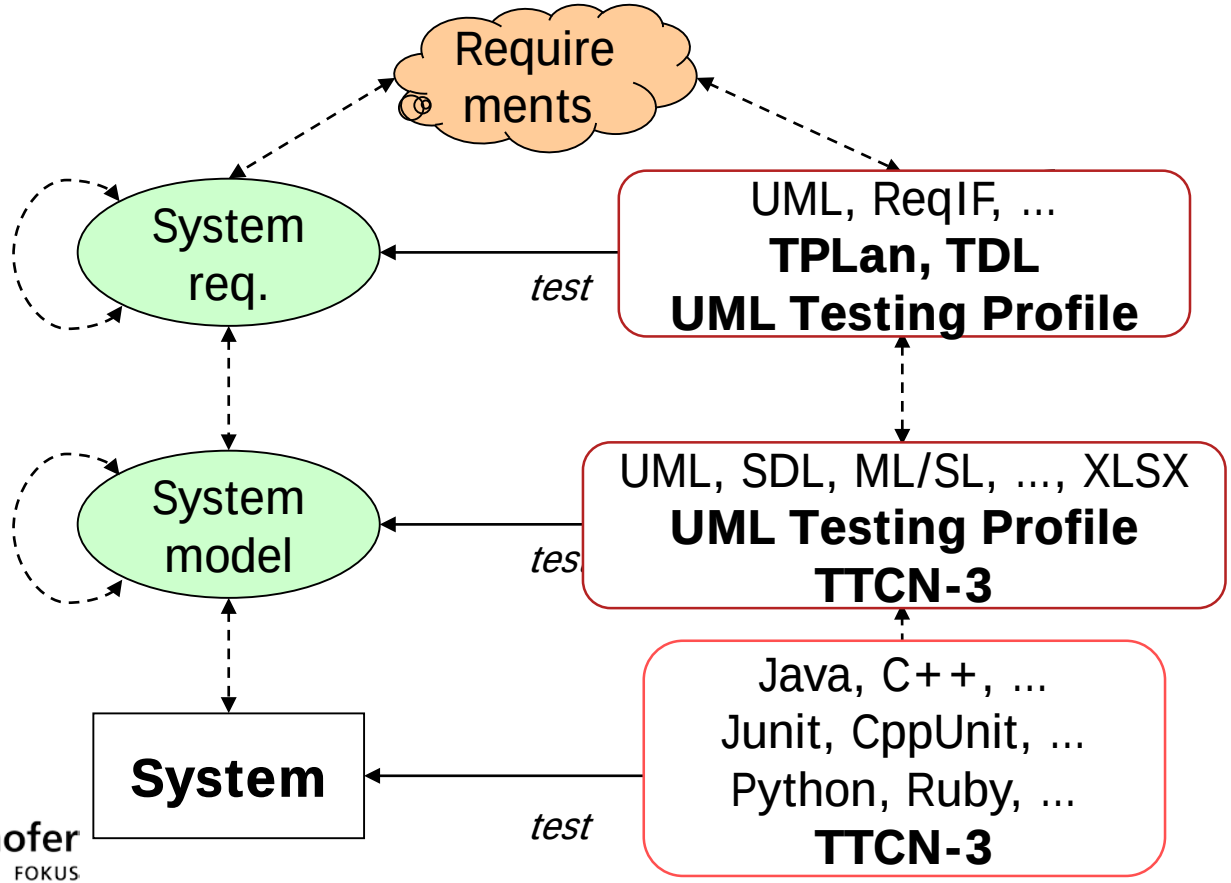
MBT 3.0 ?


Require
ments
System
req.
test
Test
req.
System
model
test
Test
model
System
test
Test
system

TEST SPECIFICATION LANGUAGES



STANDARDIZED TEST SPECIFICATION LANGUAGES



Tester's mind set

Common language

OPEN STANDARD

Fraunhofer FOKUS

LOOKING BACK AND FORWARD

What is needed in addition to a test specification language ?

METHODOLOGIES … AND AUTOMATION … AND TRAINING

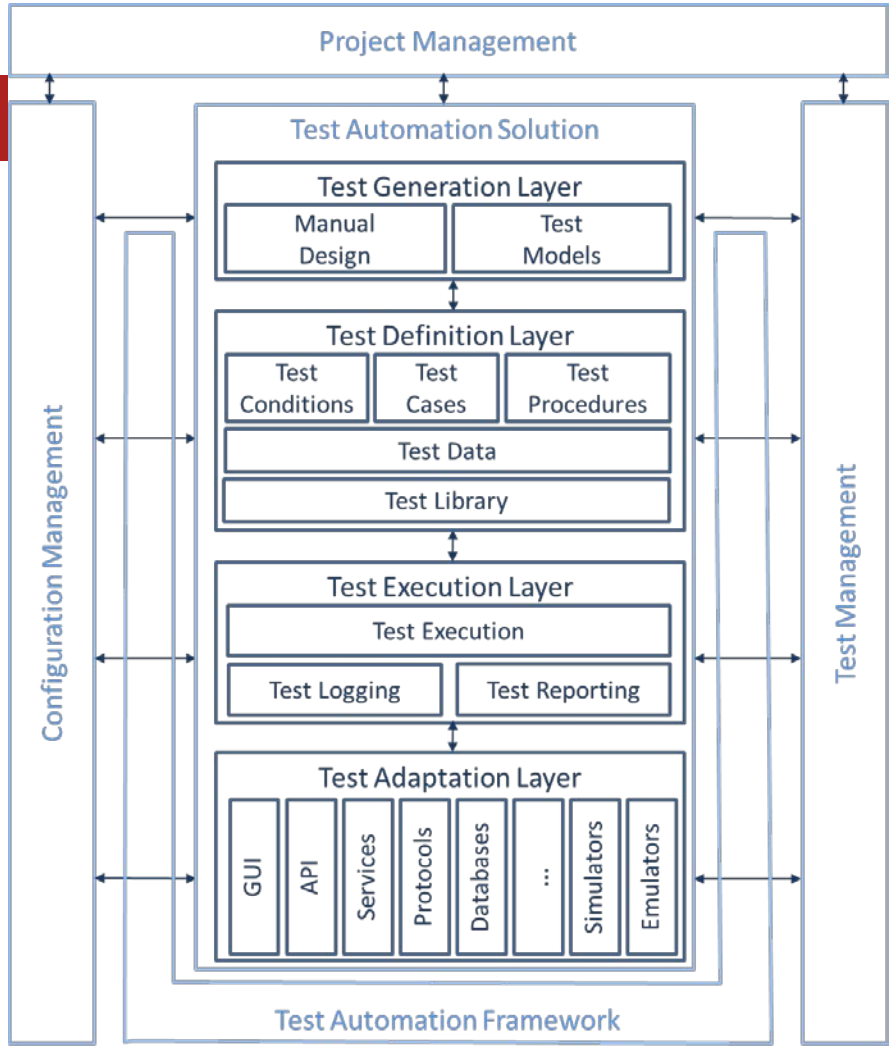
And the methodologies come with

- Tools
  - Examples
  - Guidelines
- Case studies
  - Qualitative / quantitative evaluations
  - Experience reports
- Best practices
  - Pattern / anti-patterns
  - Process (models)

TEST AUTOMATION ENGINEERING

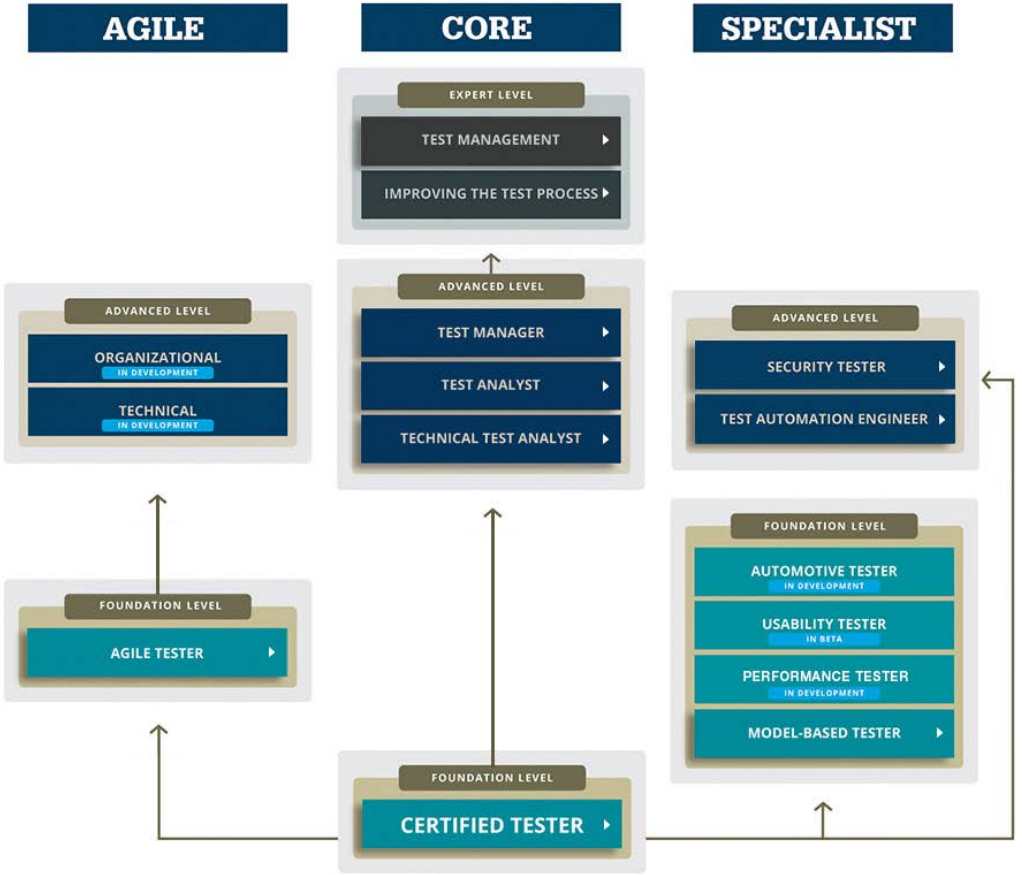
Generic Test Automation Architecture

Test Automation Engineer Syllabus



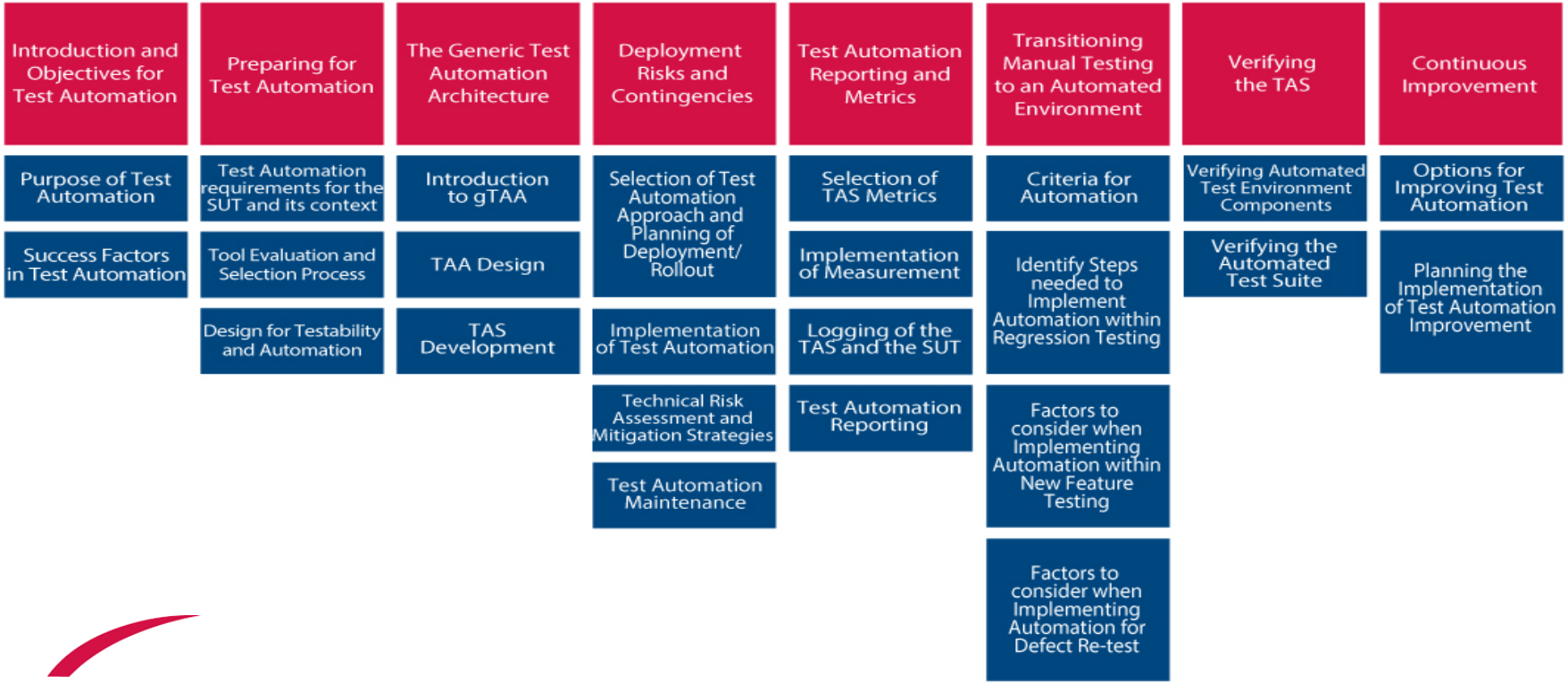
ISTQB®
International Software Testing Qualifications Board

SOFTWARE TESTING TRAINING



ISTQB®
International Software Testing Qualifications Board

CERTIFIED TEST AUTOMATION ENGINEER



ISTQB®
International Software Testing Qualifications Board

CERTIFIED MODEL-BASED TESTER

Introduction to Model-Based Testing	MBT Modeling	Selection Criteria for Test Case Generation	MBT Test Execution	Evaluating and Deploying an MBT Approach
Objectives and Motivations for MBT	MBT Modeling activities	Classification of MBT Test Selection Criteria	Specifics of MBT Test Generation and Execution	Evaluate an MBT Deployment
MBT Process	Languages for MBT Models	Applying Test Selection Criteria	MBT Test Adaptation	Manage and Monitor the Deployment of an MBT Approach
Integrating MBT into the Software Development Lifecycles	Good Practices for MBT Modeling Activities			

ISTQB®
International Software Testing Qualifications Board

TOOLING MAKES LANGUAGES AND METHODOLOGIES LIVE

They need to be practicable, usable and efficient

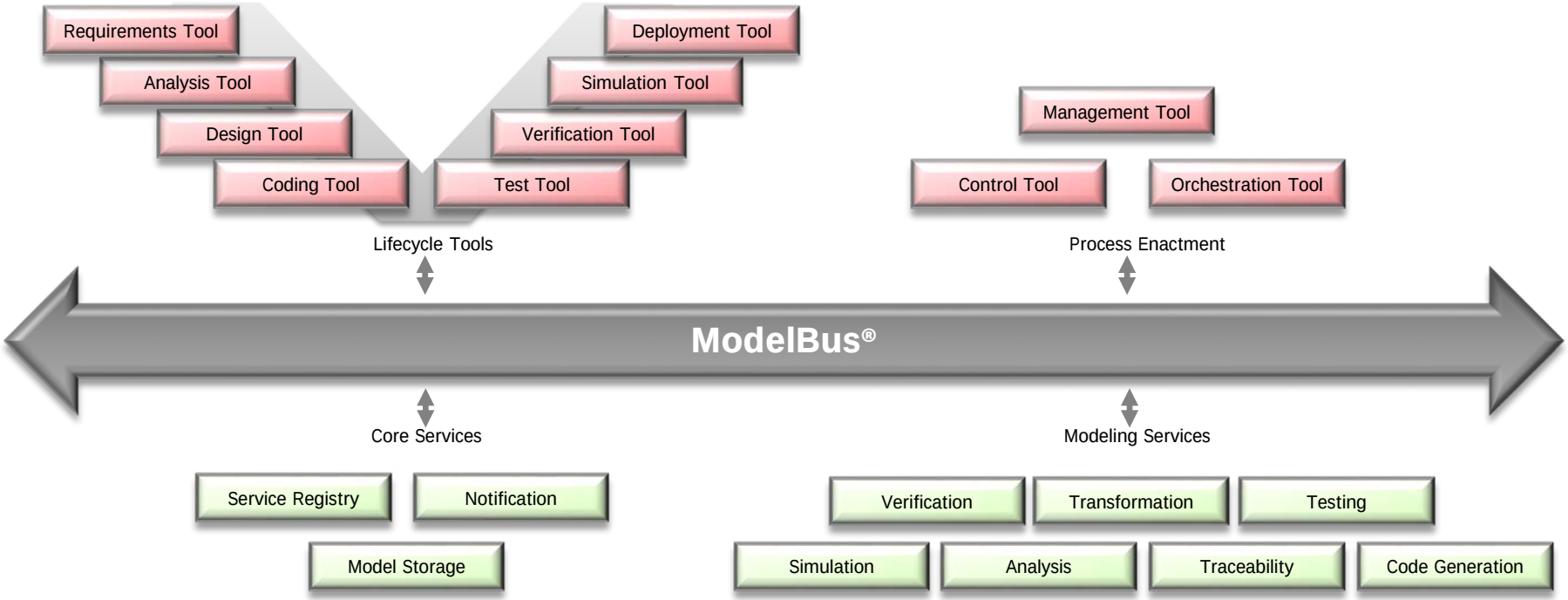
- Within established tool and process environments
- Should respect huge amounts of legacy data – being (also) models !
- Reuse the expertise of the people and teams

TOOL FRAMEWORK MODELBUS®

1. Is a model-driven tool integration framework which allows you to build seamlessly integrated tool environments for your development process.
2. Connects tools – commercial off the shelf or in-house tools
3. Helps automating development and quality assurance processes
4. Uses SOA principles and well-established standards

See http://www.modelbus.org/

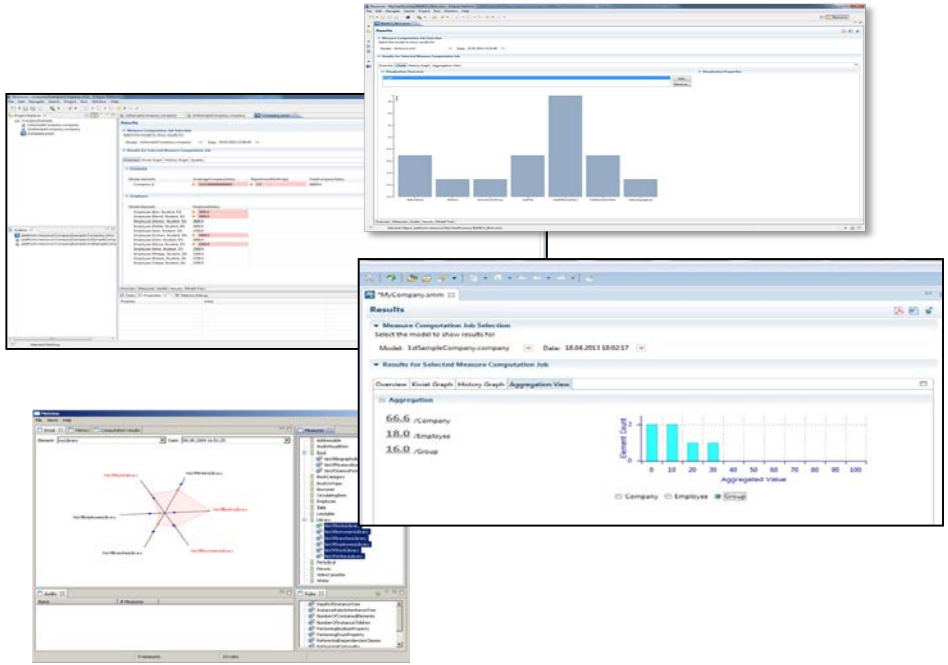
MODELBUS® - GENERAL CONCEPT



SELECTION OF CONNECTED TOOLS

- Eclipse-based Tools
- Topcased, Papyrus, ProR, …
- Rational Software Architect
- Doors
- Rhapsody
- Simulink
- Microsoft Office (Word, Excel)
- Sparx Enterprise Architect
- AVL InMotion
- …

Model-driven metric definition and computation



Challenge

Identifying properties and quality of models

Definition of metrics for complex models

Approach

Model-driven handling of quality attributes and properties

Definition of Metric Generation rules

Usage of OMGs SMM for management of metrics

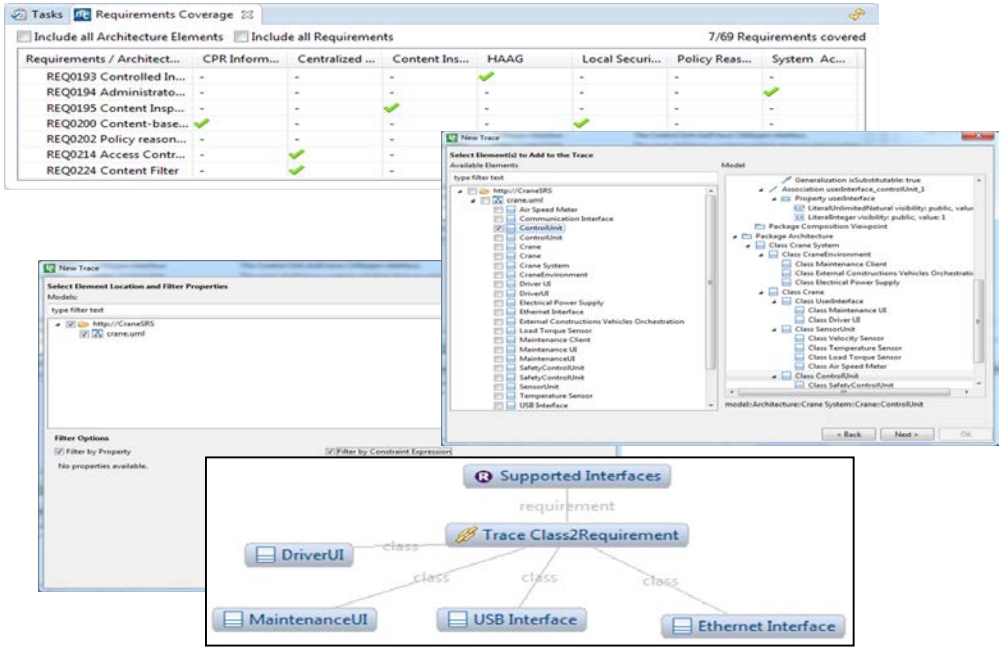
Solution

Tool front-end allows the definition and management of metrics

definition of thresholds, grouping of metrics

Visualization of Metric computation results in tabular way and kiviat graphs

Tool-independent traceability framework



Challenge

- Linking of development artefacts
- Definition of individual complex traceability models

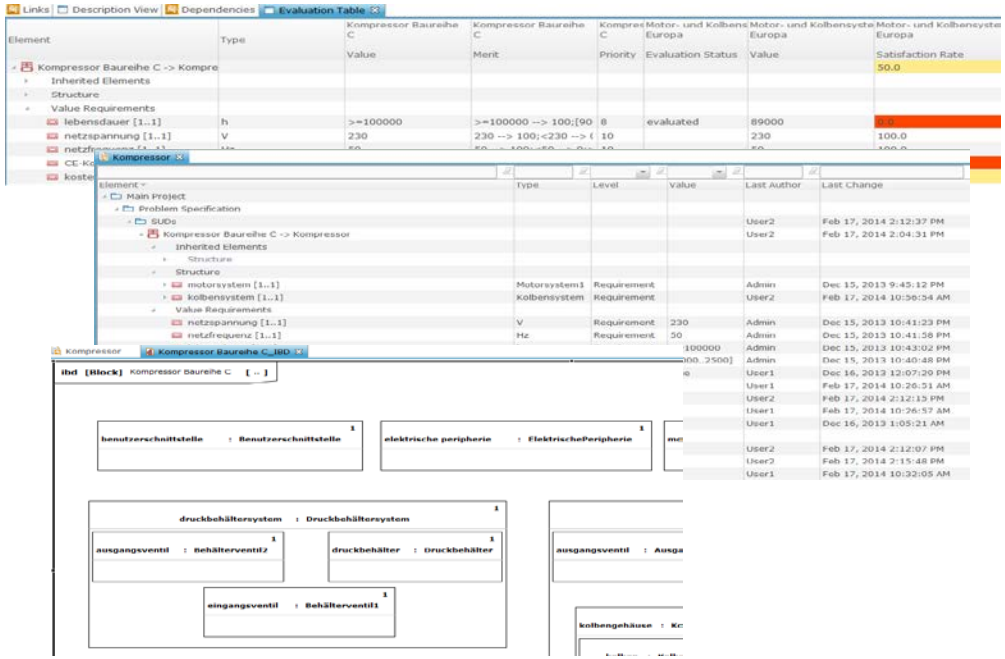
Approach

- Model-driven handling of traceability information
- Based on ModelBus tool integration approach to easily extend existing tools with traceability functionality

Solution

- Definition and utilization of type safe case specific traceability
- Navigation and quering through trace info
- Common Look and Feel UI (Web and Desktop)
- Graphical visualization of traces

Model-driven requirements engineering



Challenge

- Structured Requirements during the whole system development
- Design Space Exploration

Approach

- Model-driven handling of requirements
- Graphical refinement of requirements
- Based on SysML/UML
- Based on ModelBus (Versioning, Notification, Locking, Fragmentation, etc.)

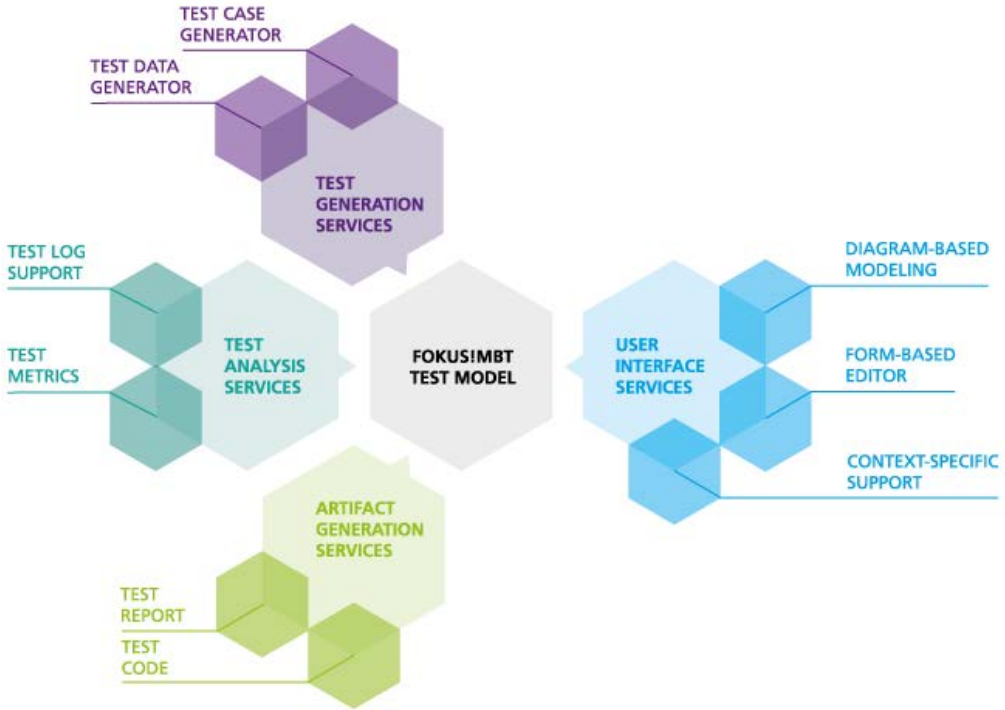
Solution

- Structured Definition of Requirements
- Refinement through structural/behavioral diagrams
- Evaluation of different concepts
- Common Look and Feel UI (Web and Desktop)

AND YES …

We also have our MBT tool

Fokus! MBT



Fraunhofer FOKUS

LOOKING FORWARD

Where should we aim at next ?

LOOKING FORWARD

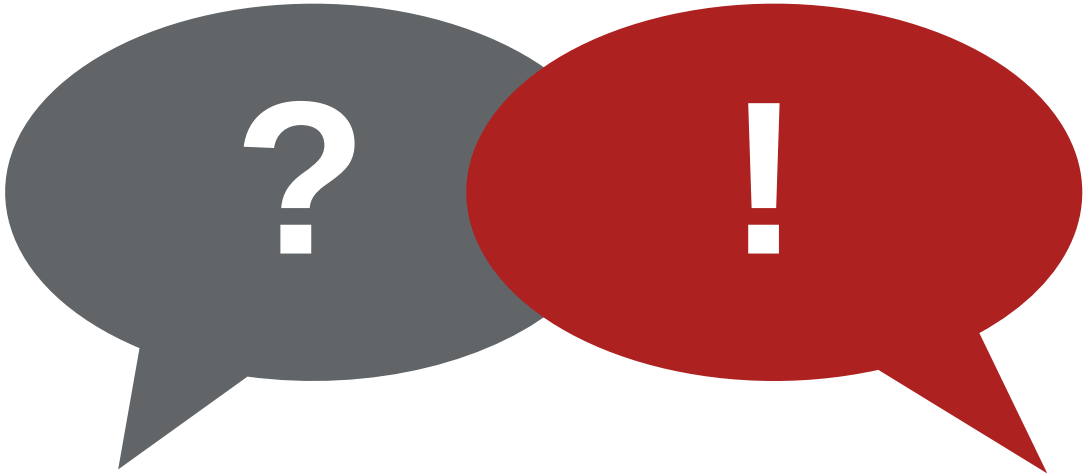
Where should we aim at next ?

- Test features as original part of software components
- Automated test framework generation
- M
- D
- Self-awareness and self-adaptation via models

Look for existing (test) specification languages

or develop your own one explicitly !

Some of which may require language extensions or even new languages …



CONTACT

Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS

Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

info@fokus.fraunhofer.de

www.fokus.fraunhofer.de

Director

Prof. Dr.-Ing. Ina Schieferdecker
Tel. +49 (30) 34 63 -7241
ina.schieferdecker@fokus.fraunhofer.de