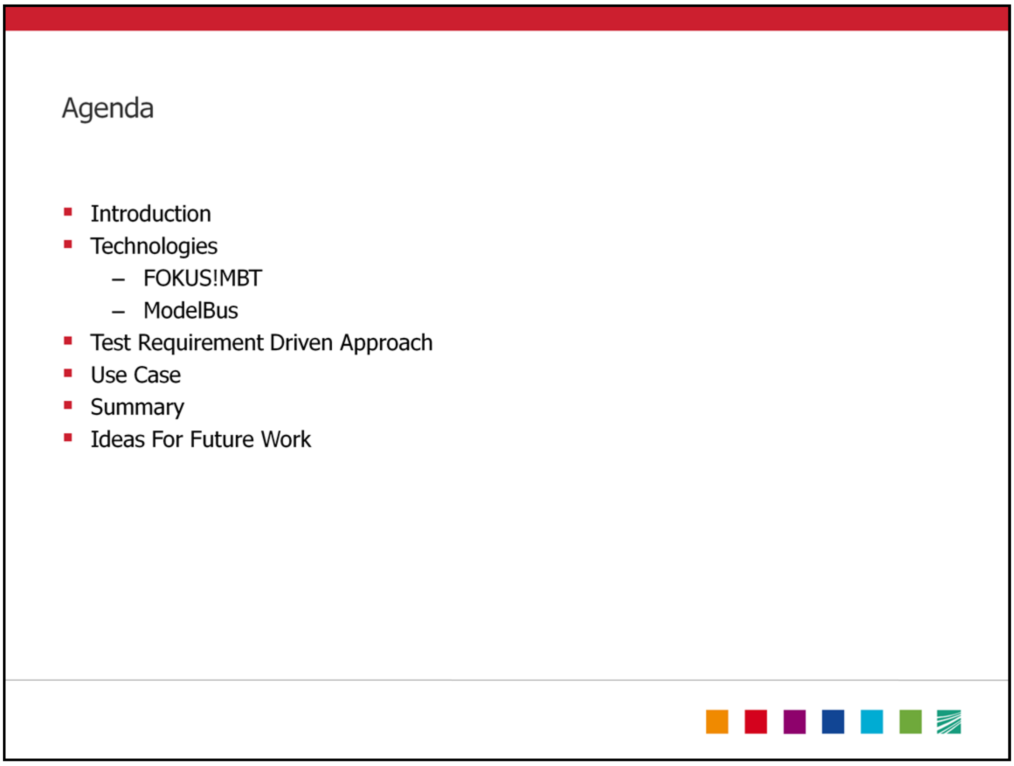
Fraunhofer
FOKUS

Fraunhofer Institute for Open Communication Systems | Kaiserin-Augusta-Allee 31 | 10589 Berlin, Germany

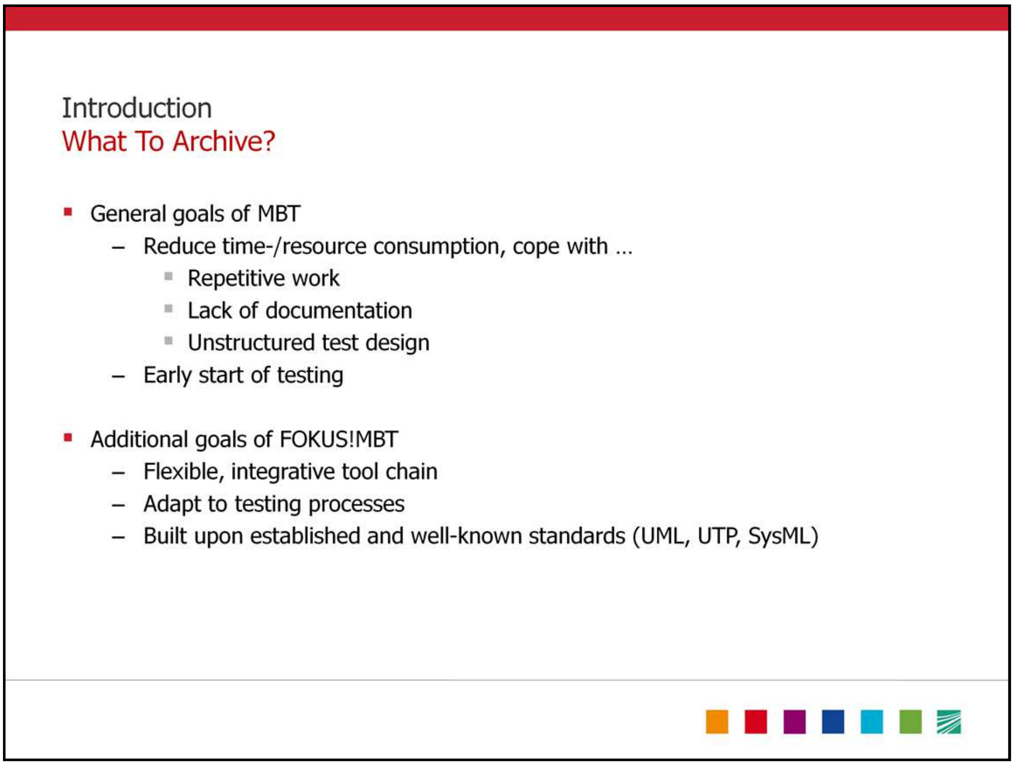
Test Requirements-Driven Testing:
Model-Based Test Automation with Fokus!MBT and ModelBus

Max Bureck, Marc-Florian Wendland | ETSI MBT User Conference 2011 | 10-20-2011 | Berlin |



Agenda

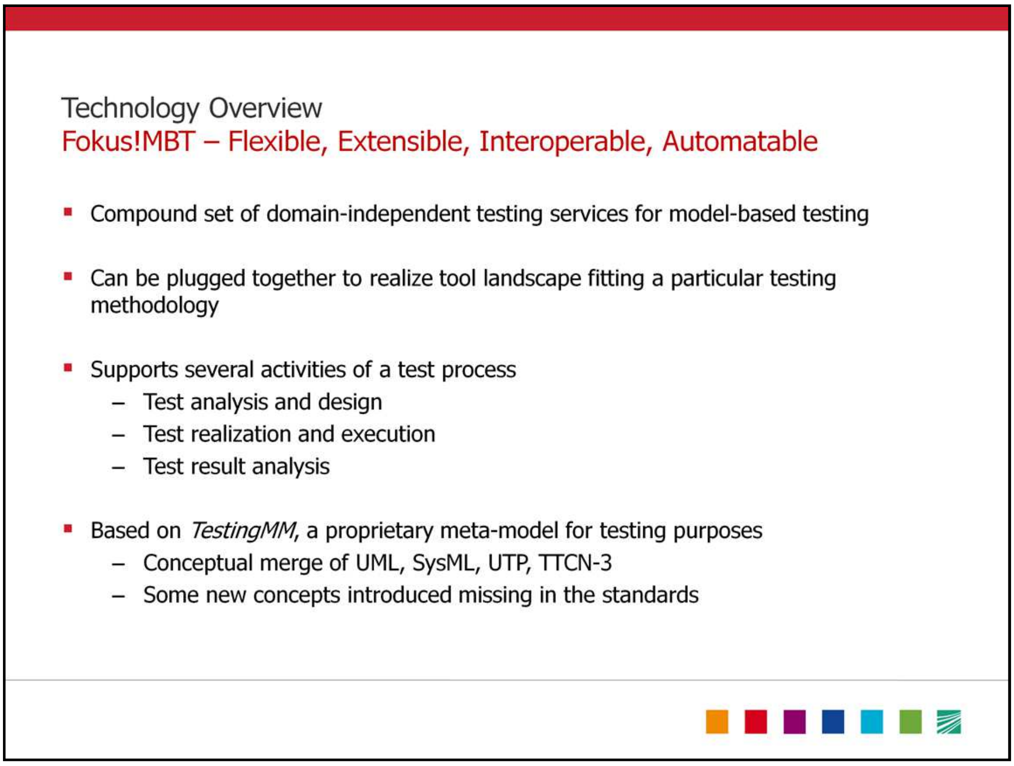
- Introduction
- Technologies
  - FOKUS!MBT
  - ModelBus
- Test Requirement Driven Approach
- Use Case
- Summary
- Ideas For Future Work



Introduction
What To Archive?

- General goals of MBT
  - Reduce time-/resource consumption, cope with …
    - Repetitive work
    - Lack of documentation
    - Unstructured test design
  - Early start of testing

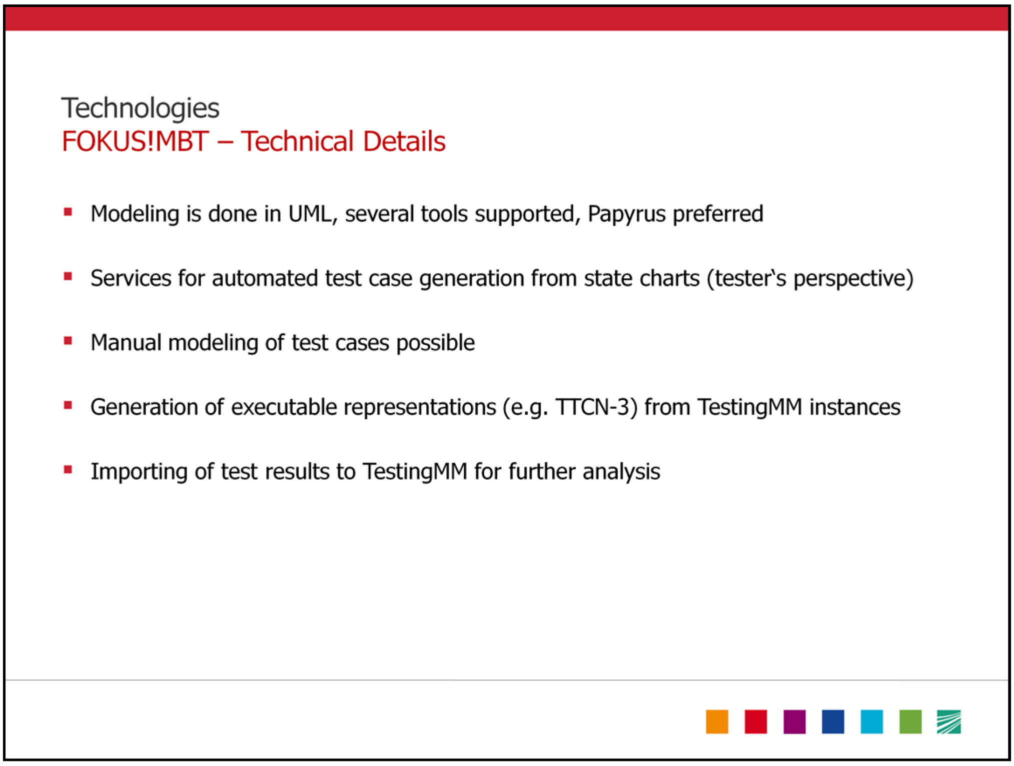
- Additional goals of FOKUS!MBT
  - Flexible, integrative tool chain
  - Adapt to testing processes
  - Built upon established and well-known standards (UML, UTP, SysML)



Technology Overview
Fokus!MBT – Flexible, Extensible, Interoperable, Automatable

- Compound set of domain-independent testing services for model-based testing
- Can be plugged together to realize tool landscape fitting a particular testing methodology
- Supports several activities of a test process
  - Test analysis and design
  - Test realization and execution
  - Test result analysis
- Based on *TestingMM*, a proprietary meta-model for testing purposes
  - Conceptual merge of UML, SysML, UTP, TTCN-3
  - Some new concepts introduced missing in the standards

- There is one ready-to-go process, that can be used out of the box
- One can choose to use the suggested process or adapt tool to own process

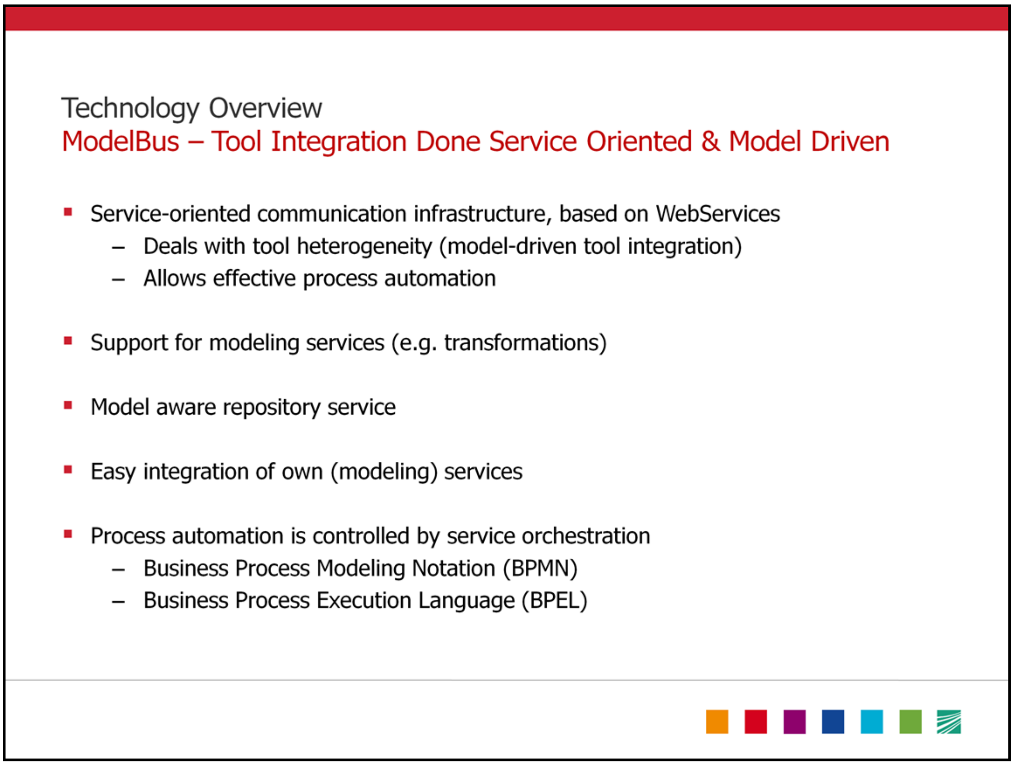


Technologies

FOKUS!MBT – Technical Details

- Modeling is done in UML, several tools supported, Papyrus preferred
- Services for automated test case generation from state charts (tester‛s perspective)
- Manual modeling of test cases possible
- Generation of executable representations (e.g. TTCN-3) from TestingMM instances
- Importing of test results to TestingMM for further analysis

- We can develop support for other tools on request
- Similar tools exist, but don‘t have a flexible service orchestration support
- Tracing from requirement to test executions



Technology Overview

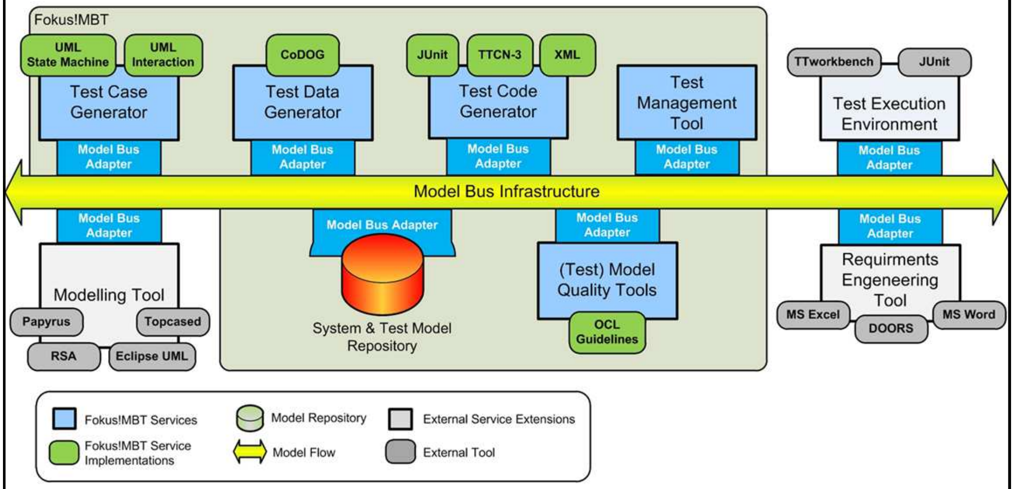
ModelBus – Tool Integration Done Service Oriented & Model Driven

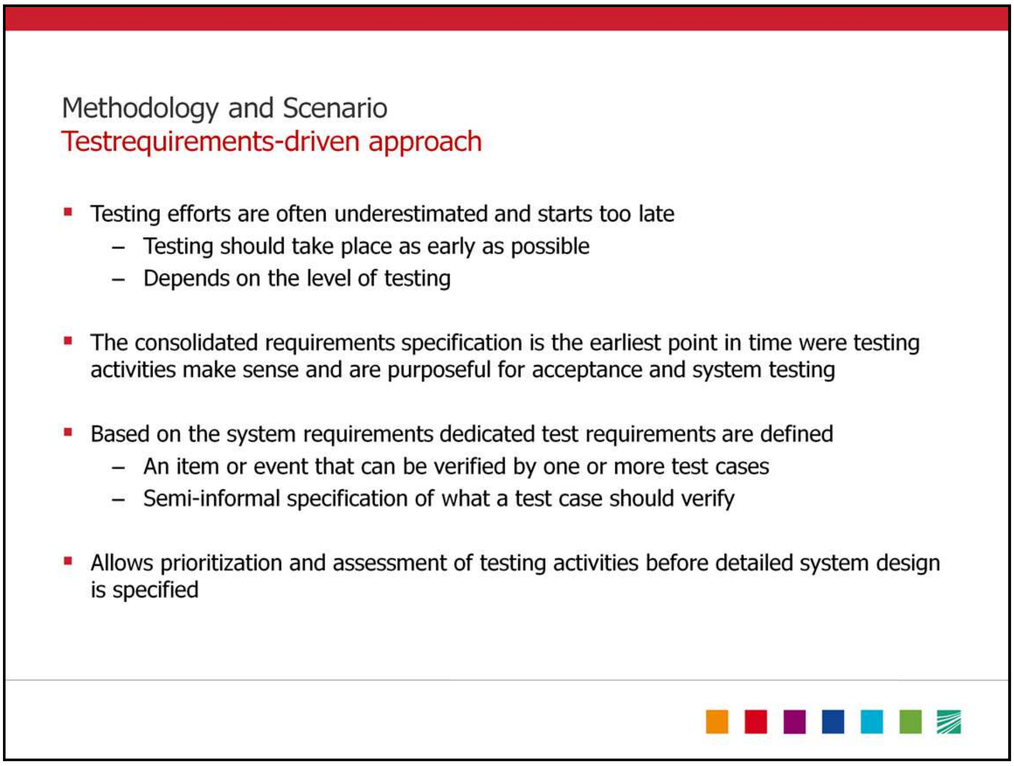
- Service-oriented communication infrastructure, based on WebServices
  - Deals with tool heterogeneity (model-driven tool integration)
  - Allows effective process automation
- Support for modeling services (e.g. transformations)
- Model aware repository service
- Easy integration of own (modeling) services
- Process automation is controlled by service orchestration
  - Business Process Modeling Notation (BPMN)
  - Business Process Execution Language (BPEL)

- Tool integration done over tool adapters, model repository and transformations between tool specific metamodels
- Repository service: like SVN for models
- Other model repositories exist, but no orchestrated modeling services

Technology Overview

Fokus!MBT & ModelBus: Service Integration

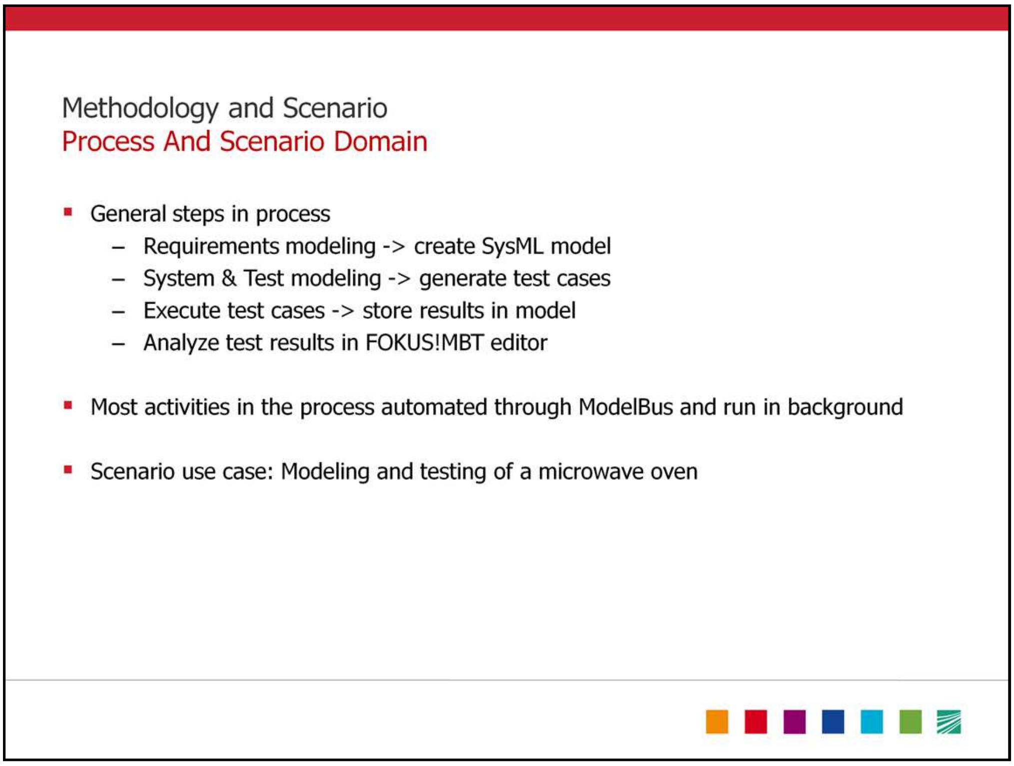




Methodology and Scenario
Testrequirements-driven approach

- Testing efforts are often underestimated and starts too late
  - Testing should take place as early as possible
  - Depends on the level of testing
- The consolidated requirements specification is the earliest point in time were testing activities make sense and are purposeful for acceptance and system testing
- Based on the system requirements dedicated test requirements are defined
  - An item or event that can be verified by one or more test cases
  - Semi-informal specification of what a test case should verify
- Allows prioritization and assessment of testing activities before detailed system design is specified

This is one of many possible methodologies, that can be implemented using FOKUS!MBT

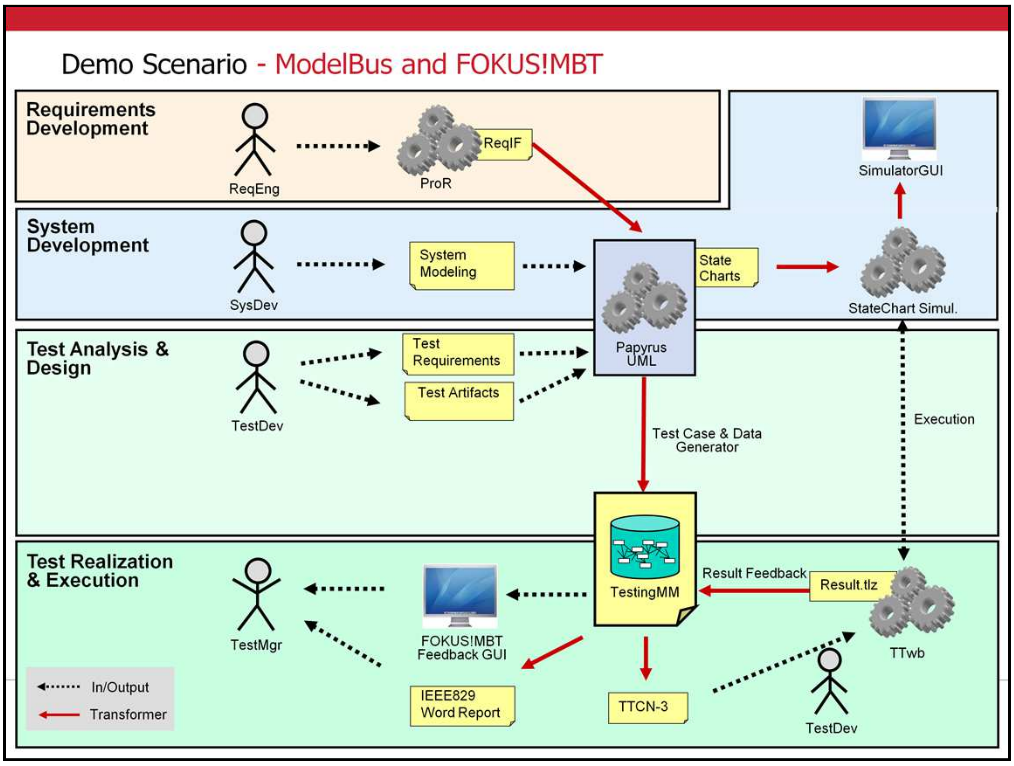


Methodology and Scenario
Process And Scenario Domain

- General steps in process
  - Requirements modeling -> create SysML model
  - System & Test modeling -> generate test cases
  - Execute test cases -> store results in model
  - Analyze test results in FOKUS!MBT editor

- Most activities in the process automated through ModelBus and run in background

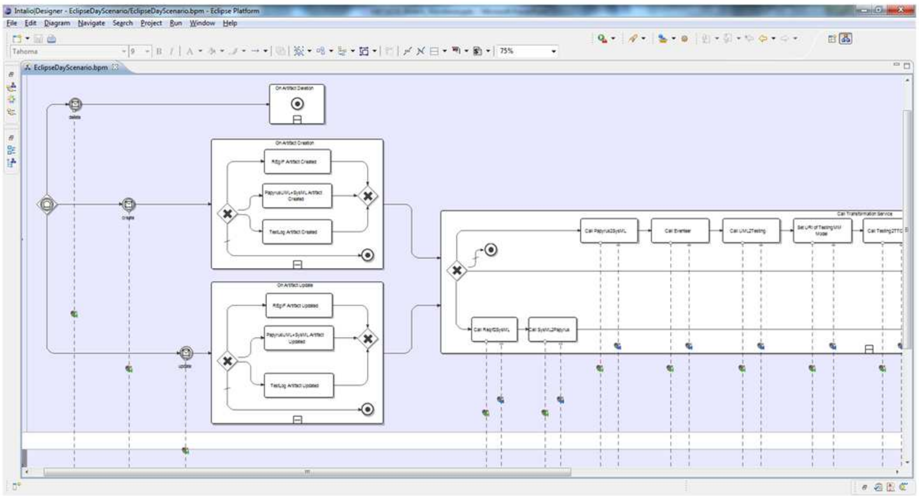
- Scenario use case: Modeling and testing of a microwave oven



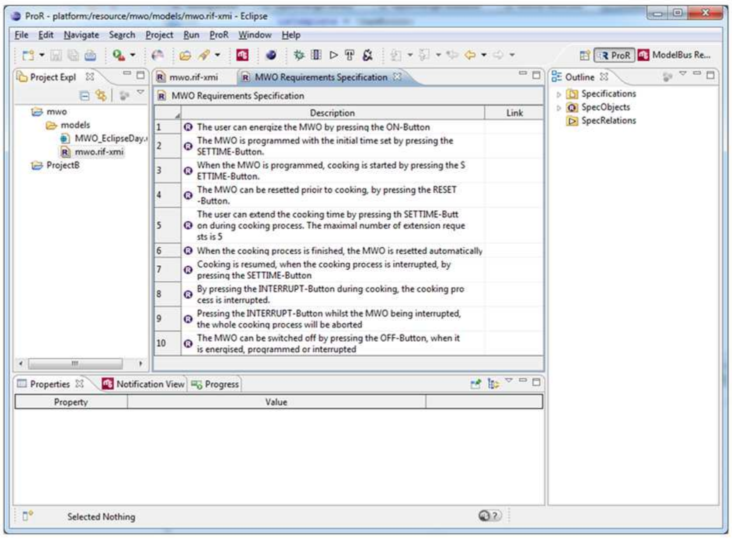
Demo Scenario - ModelBus and FOKUS!MBT

This slide will be shown in parallel to the tool screenshots

Process Orchestration on ModelBus



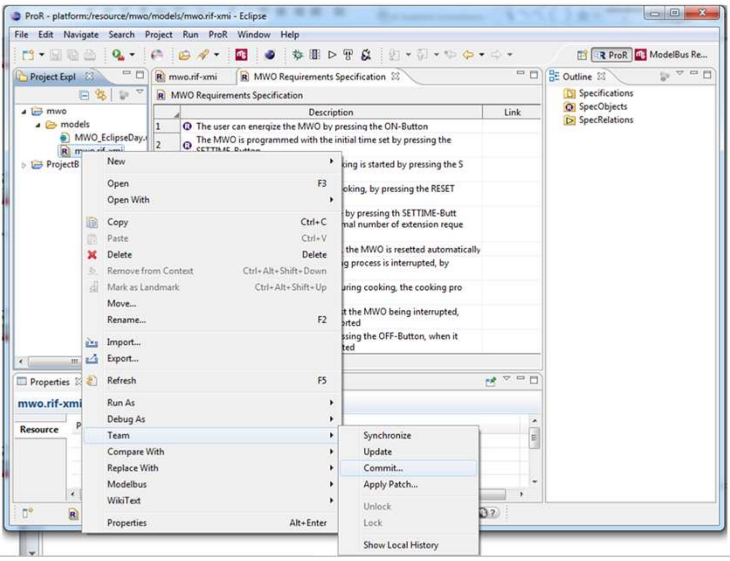
Definition of Requirements in ProR



MWO Requirements Specification

	Description	Link
1	The user can energize the MWO by pressing the ON-Button	
2	The MWO is programmed with the initial time set by pressing the SETTIME-Button.	
3	When the MWO is programmed, cooking is started by pressing the SETTIME-Button.	
4	The MWO can be resetted prioir to cooking, by pressing the RESET -Button.	
5	The user can extend the cooking time by pressing th SETTIME-Button during cooking process. The maximal number of extension requests is 5	
6	When the cooking process is finished, the MWO is resetted automatically	
7	Cooking is resumed, when the cooking process is interrupted, by pressing the SETTIME-Button	
8	By pressing the INTERRUPT-Button during cooking, the cooking process is interrupted.	
9	Pressing the INTERRUPT-Button whilst the MWO being interrupted, the whole cooking process will be aborted	
10	The MWO can be switched off by pressing the OFF-Button, when it is energised, programmed or interrupted	

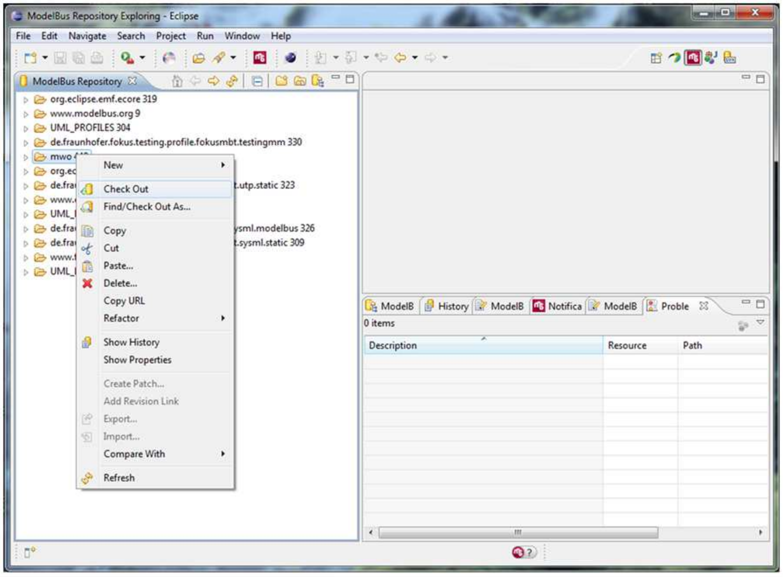
Checking in Requirements with ProR



MWO Requirements Specification

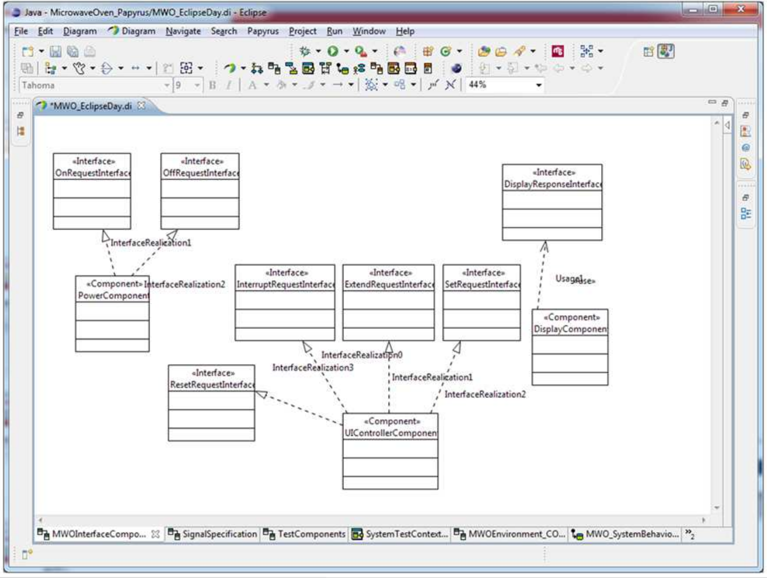
	Description	Link
1	The user can energize the MWO by pressing the ON-Button	
2	The MWO is programmed with the initial time set by pressing the SETTIME-Button	
	king is started by pressing the S	
	oking, by pressing the RESET	
	by pressing th SETTIME-Butt mal number of extension reque	
	the MWO is resetted automatically	
	g process is interrupted, by	
	uring cooking, the cooking pro	
	t the MWO being interrupted, rted	
	ssing the OFF-Button, when it ted	

Checking Out Model With Papyrus

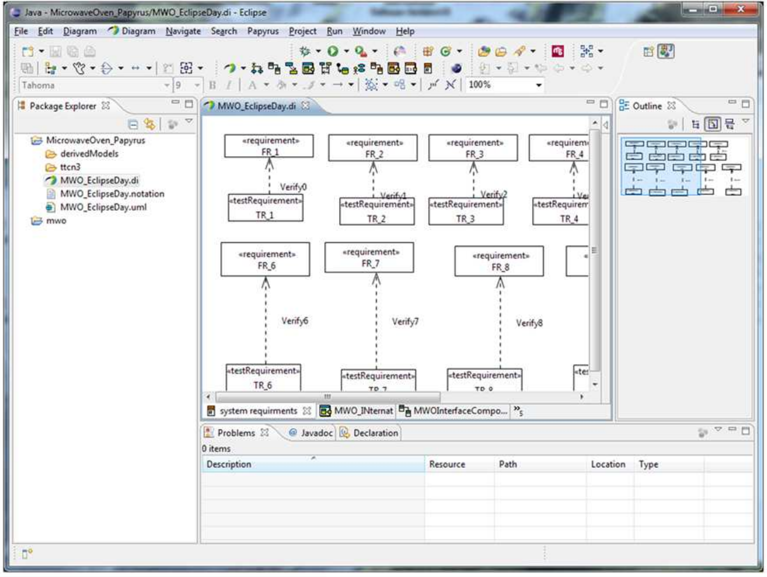


Fraunhofer
FOKUS

System Modeling in Papyrus


Java - MicrowaveOven_Papyrus/MWO_EclipseDay.di - Eclipse
File Edit Diagram Diagram Navigate Search Papyrus Project Run Window Help
Tahoma
9
44%
*MWO_EclipseDay.di
«Interface» OnRequestInterfac
«Interface» OffRequestInterfac
«Interface» DisplayResponseInterfac
InterfaceRealization1
«Component» PowerComponent
InterfaceRealization2
«Interface» InterruptRequestInterfac
«Interface» ExtendRequestInterfac
«Interface» SetRequestInterfac
Usage1
«use»
«Component» DisplayComponen
InterfaceRealization0
InterfaceRealization3
«Interface» ResetRequestInterfac
InterfaceRealization1
InterfaceRealization2
«Component» UIControllerComponen
MWOInterfaceCompo...
SignalSpecification
TestComponents
SystemTestContext...
MWOEnvironment_CO...
MWO_SystemBehavio...
»2

Test Modeling in Papyrus (Test Requirements)



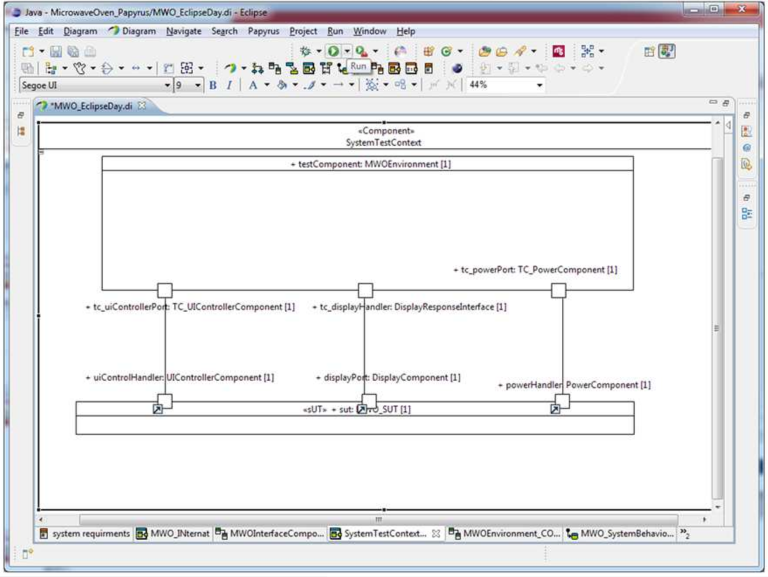
- MicrowaveOven_Papyrus
  - derivedModels
  - ttcn3
  - MWO_EclipseDay.di
  - MWO_EclipseDay.notation
  - MWO_EclipseDay.uml
- mwo

0 items

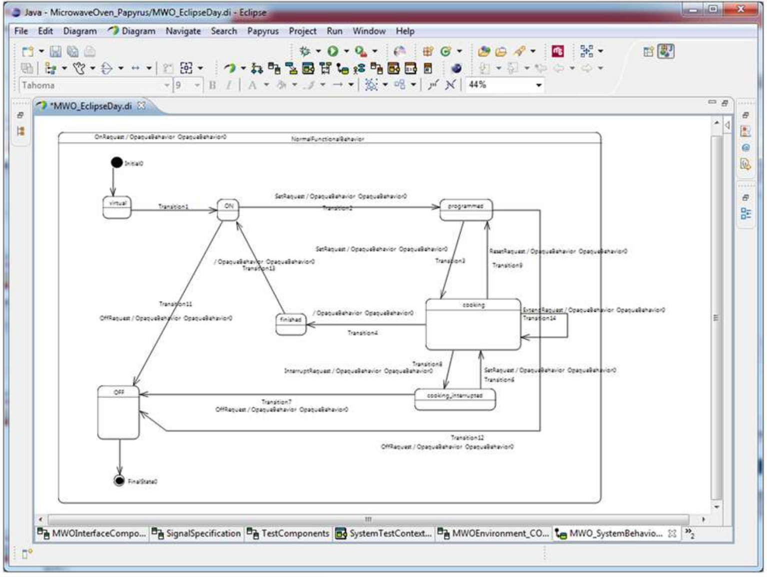
Description	Resource	Path	Location	Type

Fraunhofer
FOKUS

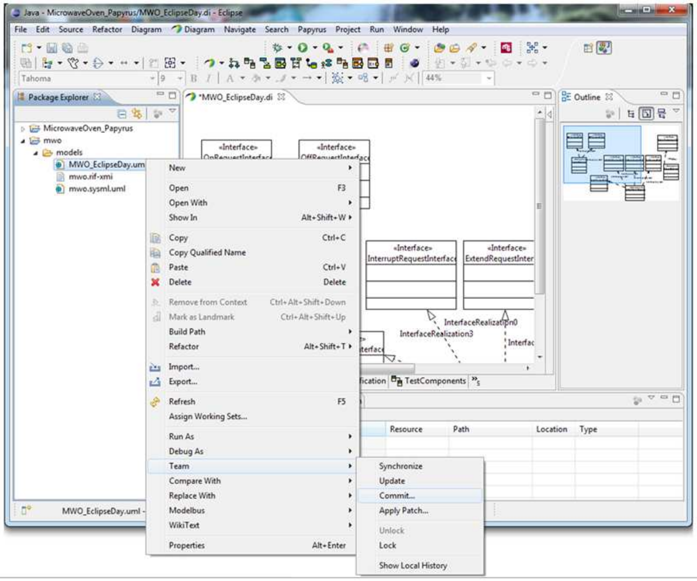
Test Modeling in Papyrus (Test Configuration)


Java - MicrowaveOven_Papyrus/MWO_EclipseDay.di - Eclipse
File Edit Diagram Diagram Navigate Search Papyrus Project Run Window Help
Run
Segoe UI
9
44%
*MWO_EclipseDay.di
«Component»
SystemTestContext
+ testComponent: MWOEnvironment [1]
+ tc_powerPort: TC_PowerComponent [1]
+ tc_uiControllerPort: TC_UIControllerComponent [1]
+ tc_displayHandler: DisplayResponseInterface [1]
+ uiControlHandler: UIControllerComponent [1]
+ displayPort: DisplayComponent [1]
+ powerHandler: PowerComponent [1]
«sUT» + sut: MWO_SUT [1]
system requirments
MWO_INternat
MWOInterfaceCompo...
SystemTestContext...
MWOEnvironment_CO...
MWO_SystemBehavio...
»2

Test Modeling in Papyrus (SUT State Chart, Tester‘s Perspective)

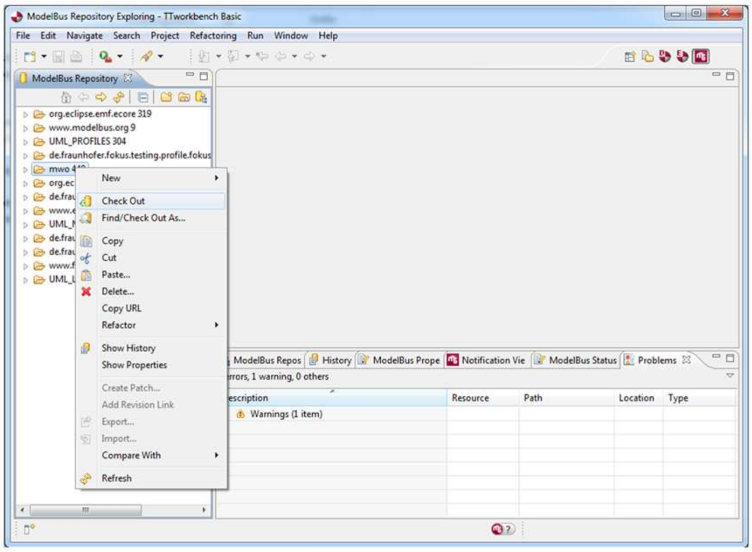

Java - MicrowaveOven_Papyrus/MWO_EclipseDay.di - Eclipse
File Edit Diagram Diagram Navigate Search Papyrus Project Run Window Help
Tahoma
9
44%
*MWO_EclipseDay.di
NormalFunctionalBehavior
OnRequest / OpaqueBehavior OpaqueBehavior0
Initial0
virtual
Transition1
ON
SetRequest / OpaqueBehavior OpaqueBehavior0
Transition2
programmed
SetRequest / OpaqueBehavior OpaqueBehavior0
Transition3
ResetRequest / OpaqueBehavior OpaqueBehavior0
Transition9
/ OpaqueBehavior OpaqueBehavior0
Transition13
Transition11
OffRequest / OpaqueBehavior OpaqueBehavior0
finished
/ OpaqueBehavior OpaqueBehavior0
Transition4
cooking
ExtendRequest / OpaqueBehavior OpaqueBehavior0
Transition14
Transition8
InterruptRequest / OpaqueBehavior OpaqueBehavior0
SetRequest / OpaqueBehavior OpaqueBehavior0
Transition6
cooking_interrupted
OFF
Transition7
OffRequest / OpaqueBehavior OpaqueBehavior0
Transition12
OffRequest / OpaqueBehavior OpaqueBehavior0
FinalState0
MWOInterfaceCompo...
SignalSpecification
TestComponents
SystemTestContext...
MWOEnvironment_CO...
MWO_SystemBehavio...
»2

Checking in System & Test Model in Papyrus



Fraunhofer
FOKUS

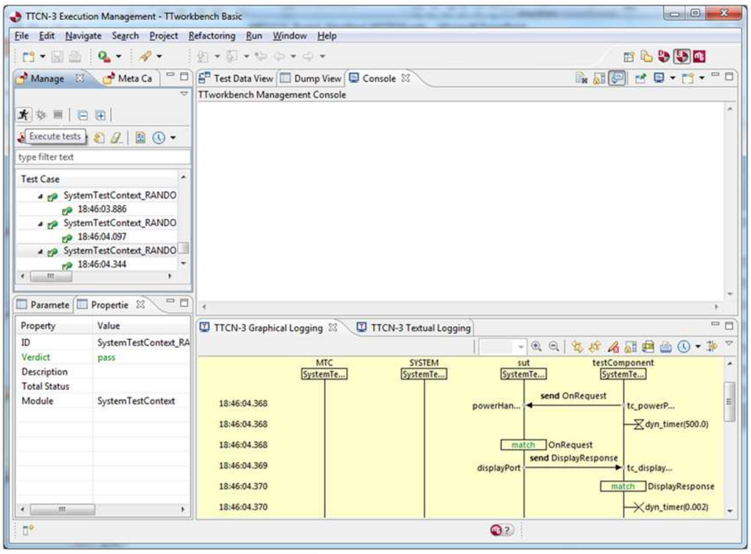
Check Out Generated Testscripts in TTWorkbench



rrors, 1 warning, 0 others

escription	Resource	Path	Location	Type
Warnings (1 item)				

Execution of Generated Test Cases With TTWorkbench

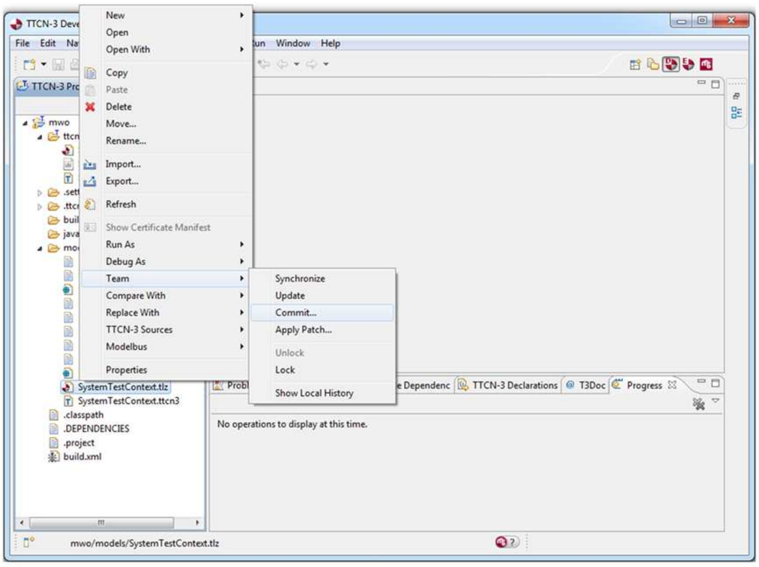


Test Case

- SystemTestContext_RANDO
  - 18:46:03.886
- SystemTestContext_RANDO
  - 18:46:04.097
- SystemTestContext_RANDO
  - 18:46:04.344

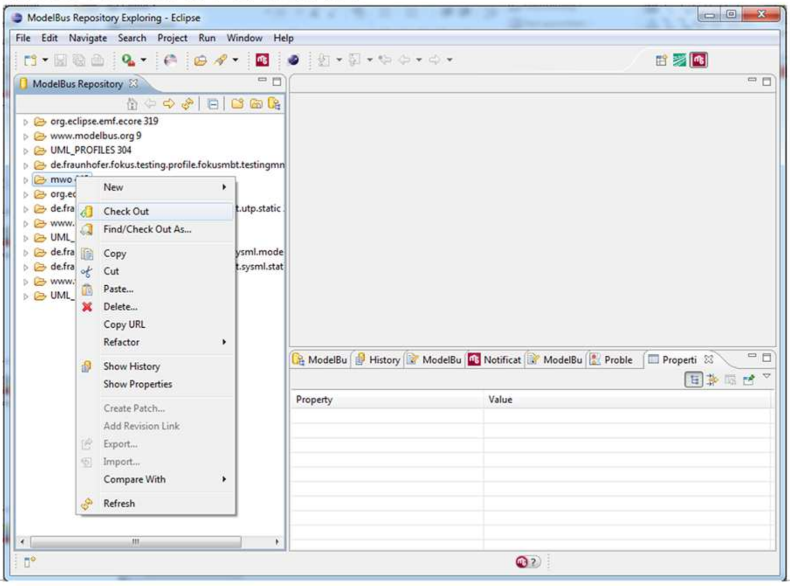
Property	Value
ID	SystemTestContext_RA
Verdict	pass
Description	
Total Status	
Module	SystemTestContext

Check in Test Results in TTWorkbench



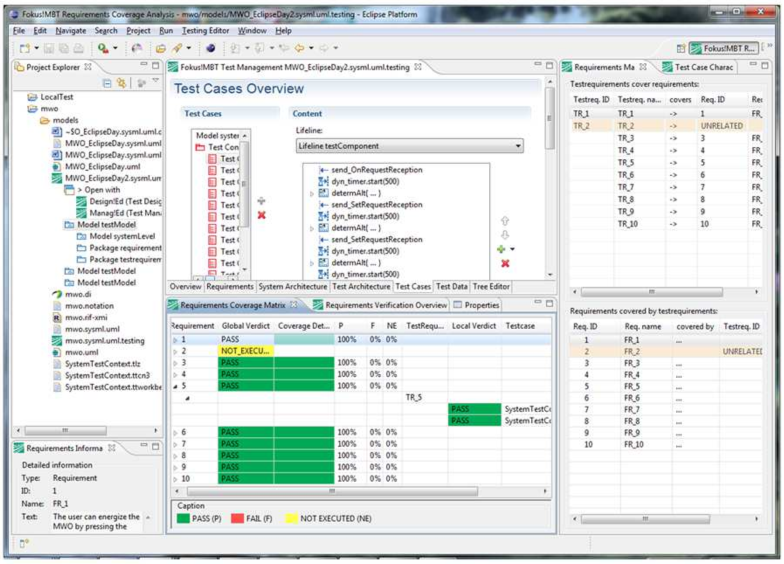
No operations to display at this time.

Check Out Results in FOKUS!MBT



Fraunhofer FOKUS

Analyze Results in FOKUS!MBT



send_OnRequestReception
dyn_timer.start(500)
determAlt(...)
send_SetRequestReception
dyn_timer.start(500)
determAlt(...)
send_SetRequestReception
dyn_timer.start(500)
determAlt(...)
dyn_timer.start(500)

Requirement	Global Verdict	Coverage Det...	P	F	NE	TestRequ...	Local Verdict	Testcase
1	PASS		100%	0%	0%			
2	NOT_EXECU...							
3	PASS		100%	0%	0%			
4	PASS		100%	0%	0%			
5	PASS		100%	0%	0%			
						TR_5		
							PASS	SystemTestC
							PASS	SystemTestC
6	PASS		100%	0%	0%			
7	PASS		100%	0%	0%			
8	PASS		100%	0%	0%			
9	PASS		100%	0%	0%			
10	PASS		100%	0%	0%			

Caption
🟩 PASS (P) 🟥 FAIL (F) 🟨 NOT EXECUTED (NE)

Detailed information
Type: Requirement
ID: 1
Name: FR_1
Text: The user can energize the MWO by pressing the

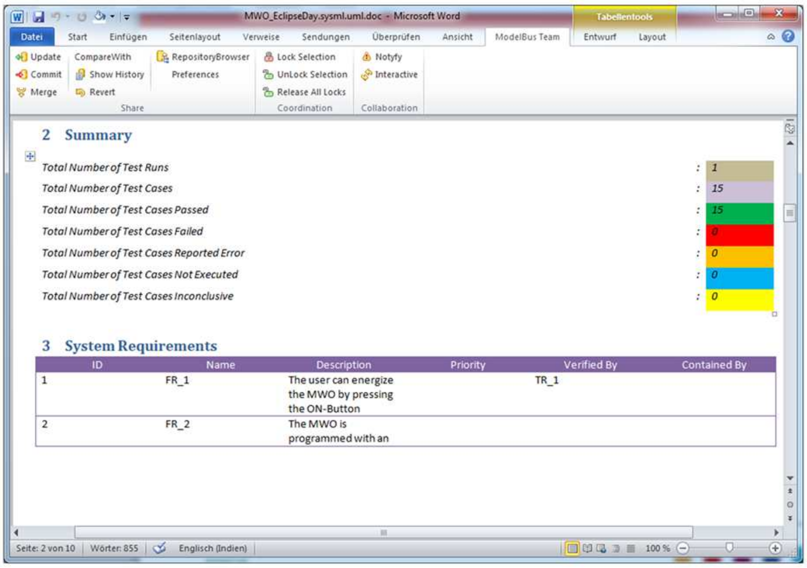
Testrequirements cover requirements:

Testreq. ID	Testreq. na...	covers	Req. ID	Re
TR_1	TR_1	->	1	FR_
TR_2	TR_2	->	UNRELATED	
	TR_3	->	3	FR_
	TR_4	->	4	FR_
	TR_5	->	5	FR_
	TR_6	->	6	FR_
	TR_7	->	7	FR_
	TR_8	->	8	FR_
	TR_9	->	9	FR_
	TR_10	->	10	FR_

Requirements covered by testrequirements:

Req. ID	Req. name	covered by	Testreq. ID
1	FR_1	...	
2	FR_2		UNRELATEI
3	FR_3	...	
4	FR_4	...	
5	FR_5	...	
6	FR_6	...	
7	FR_7	...	
8	FR_8	...	
9	FR_9	...	
10	FR_10	...	

View Test Results in Word File

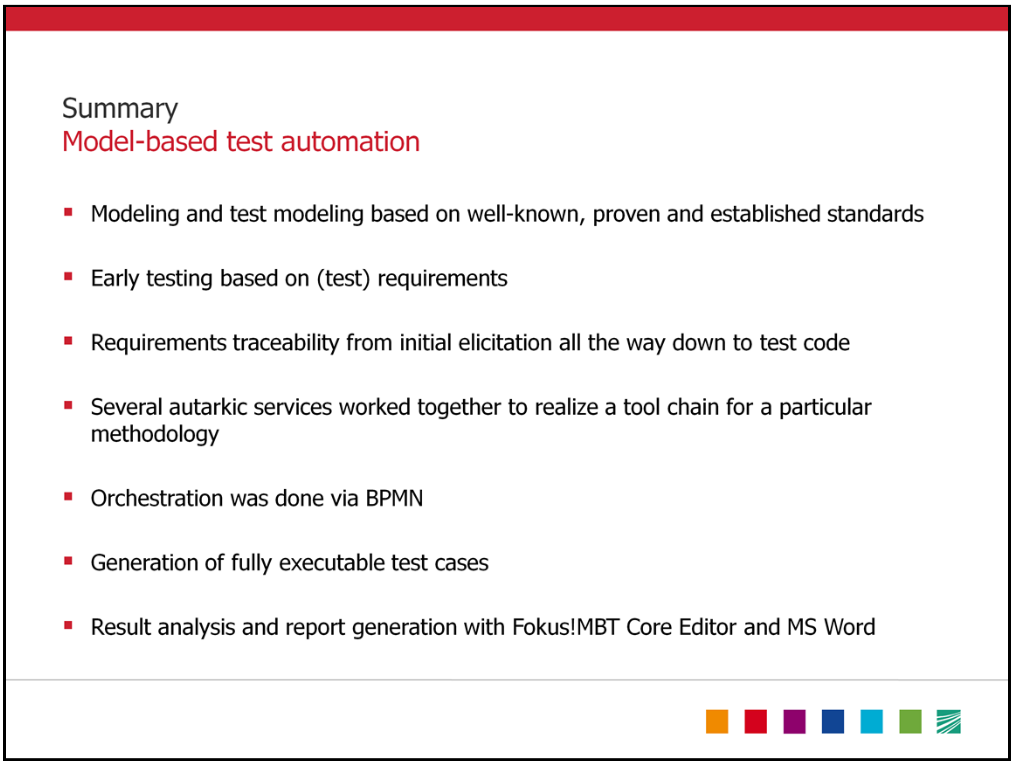


2 Summary

Total Number of Test Runs	:	*1*
Total Number of Test Cases	:	*15*
Total Number of Test Cases Passed	:	*15*
Total Number of Test Cases Failed	:	*0*
Total Number of Test Cases Reported Error	:	*0*
Total Number of Test Cases Not Executed	:	*0*
Total Number of Test Cases Inconclusive	:	*0*

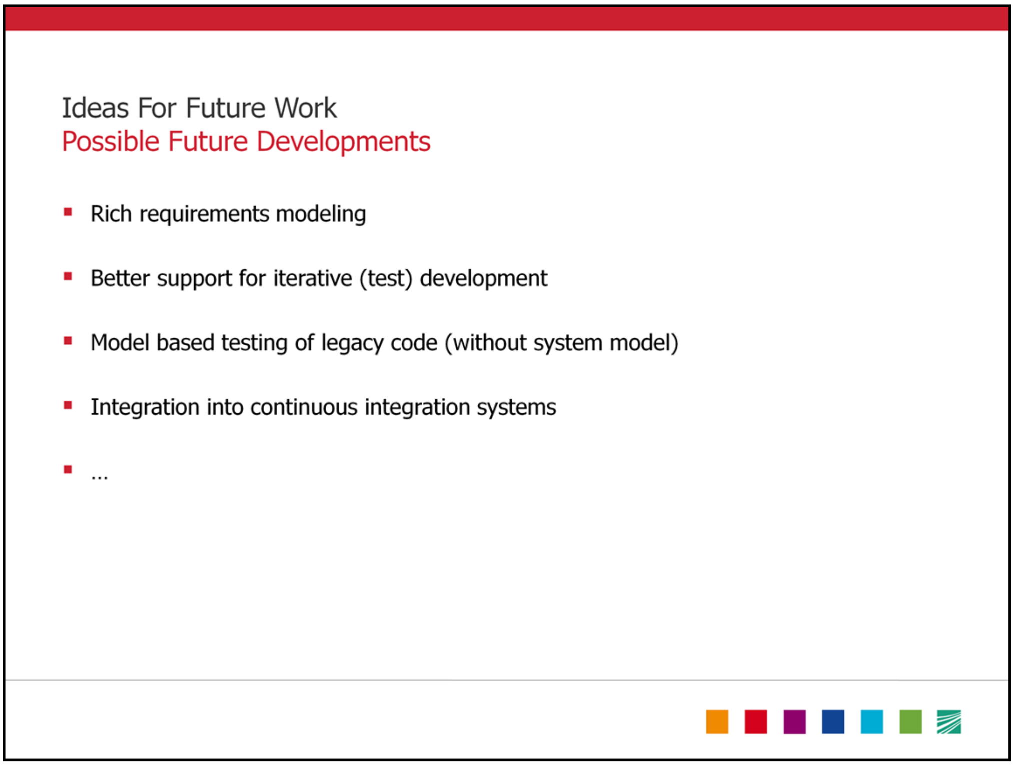
3 System Requirements

ID	Name	Description	Priority	Verified By	Contained By
1	FR_1	The user can energize the MWO by pressing the ON-Button		TR_1	
2	FR_2	The MWO is programmed with an			



Summary
Model-based test automation

- Modeling and test modeling based on well-known, proven and established standards
- Early testing based on (test) requirements
- Requirements traceability from initial elicitation all the way down to test code
- Several autarkic services worked together to realize a tool chain for a particular methodology
- Orchestration was done via BPMN
- Generation of fully executable test cases
- Result analysis and report generation with Fokus!MBT Core Editor and MS Word



Ideas For Future Work

Possible Future Developments

- Rich requirements modeling
- Better support for iterative (test) development
- Model based testing of legacy code (without system model)
- Integration into continuous integration systems
- …

Open architecture allows addition of services to the tool chain

Thanks for your attention

QUESTIONS?