UML TESTING PROFILE 2
OMG‘S NEW STANDARD FOR MODEL-BASED TESTING

Marc-Florian Wendland, Ina Schieferdecker

ICST 2017, Tokyo, March 16, 2017

Fraunhofer FOKUS

UML TESTING PROFILE 2

Goal of this tutorial

- Understand the scope of UTP 2
- Find its place in the ocean of testing standards
- Become aware of its capabilities in terms of building software test architectures
- Get inspiration how UTP 2 can be applied for automated test design

Fraunhofer FOKUS

AGENDA

1. The UML Testing Profile @ a Glance
2. What can I do with UTP 2?
3. What has not yet been said

AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

- **Test Language**
- **Arbitration Specifications**
- **Test Design Facility**

3. What has not yet been said

AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

3. What has not yet been said

THE UML TESTING PROFILE @ A GLANCE

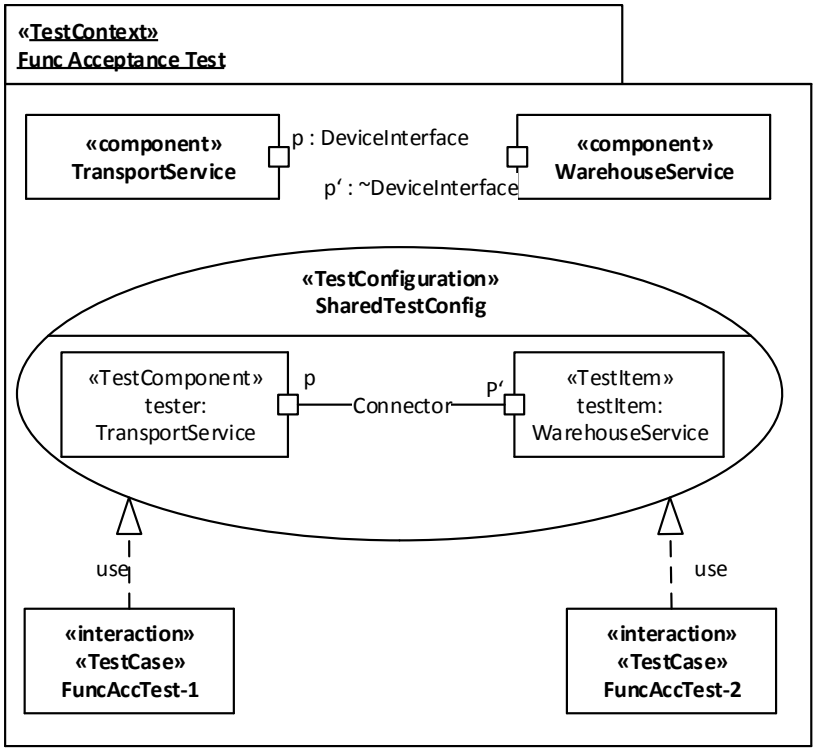
Understanding UTP

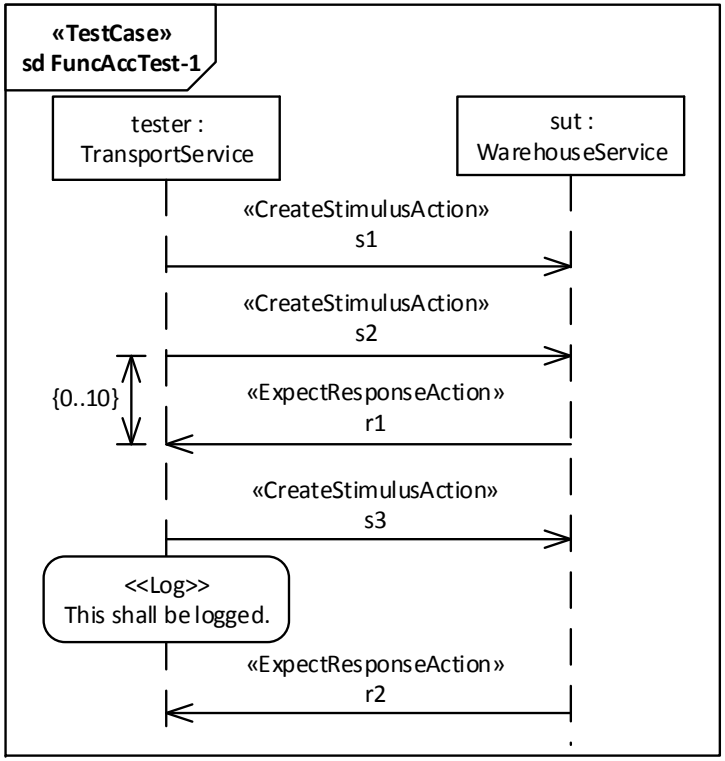
- A modeling **language** based on the UML [1]
- Supports **(test) engineers** in carrying out (manual or automated) **test design activities**
- Specification of **test models and test logs**
- Facilitates (manual or automated) **test execution** and **evaluation**
- Simplifies **communication and understanding** among stakeholders
- Vendor- and methodology-independent (i.e., open) standard

UTP abides by the idea of model-driven engineering but for testing (test automation) purposes

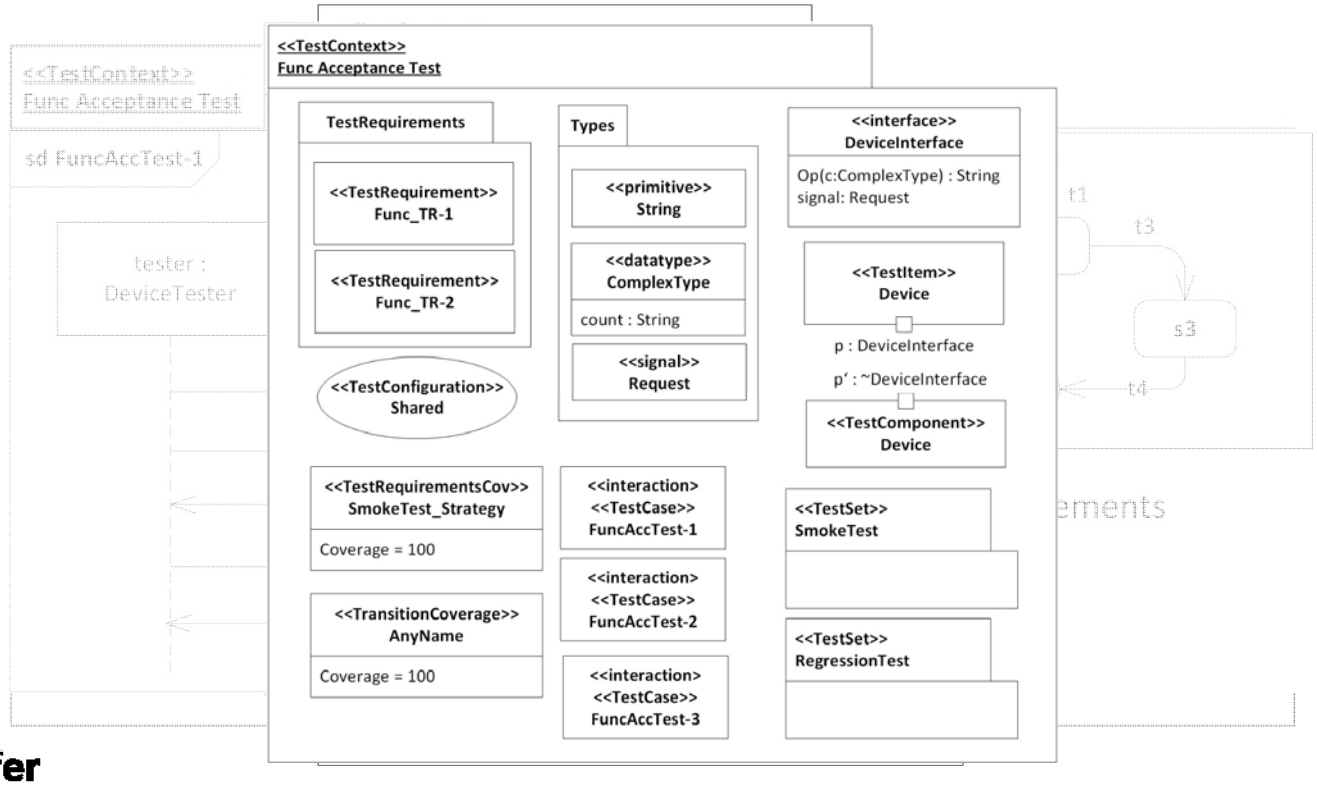
Fraunhofer FOKUS

THE UML TESTING PROFILE @ A GLANCE





WHAT CAN I DO WITH UTP 2?



Fraunhofer FOKUS

THE UML TESTING PROFILE @ A GLANCE

Understanding UTP – Out of Scope

- Methodologies
- Modeling of test processes and/or higher-level test management concepts (such as test strategies, role concepts etc.)
- Static testing such as audits/reviews, static code analysis, etc.

THE UML TESTING PROFILE @ A GLANCE

UTP in the ocean of testing and domain-specific standards

ISO 29119-1

ISTQB CMBT

IEEE:ATML

ETSI TDL

ISO 26262

ETSI MBT

ETSI TTCN-3

ISO 29119-4

SoaML

Do-178C

UML TESTING PROFILE

TestIF

IEEE:829

ISO 29119-3

ISTQB TAE

UML

EN 50129

ISO 29119-5

ETSI TPLan

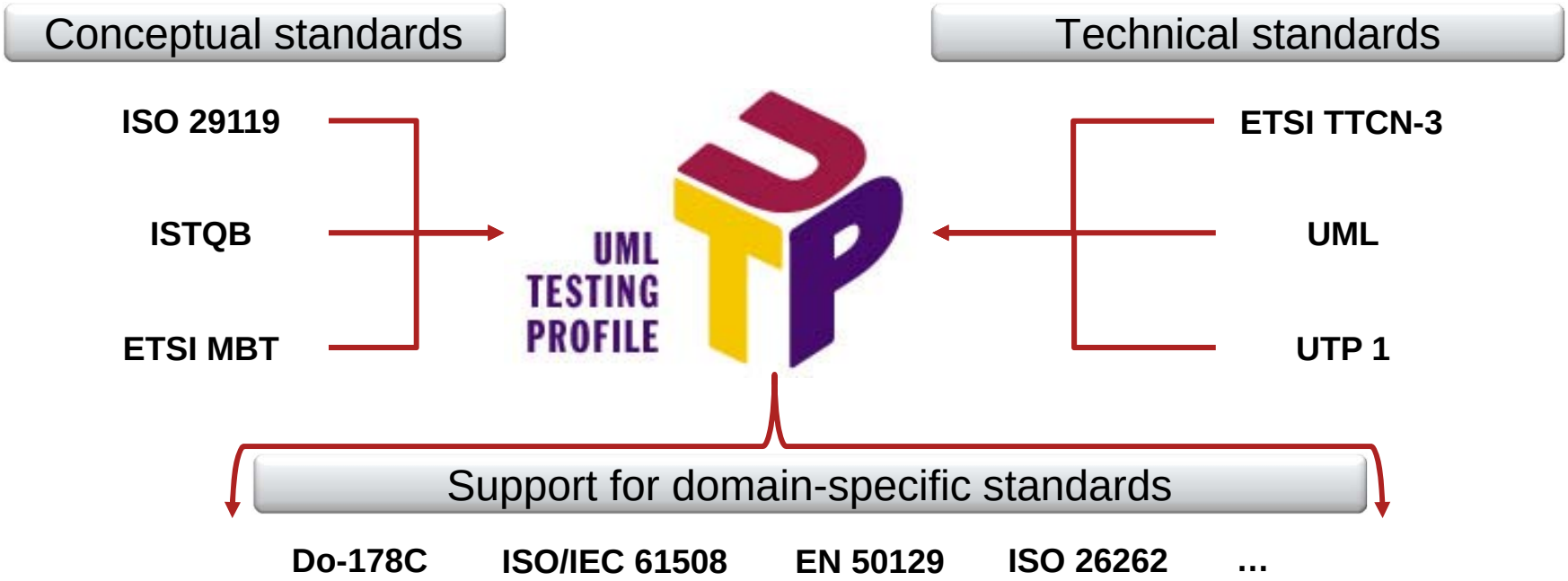
SysML

ISO/IEC 61508

UTP 1

THE UML TESTING PROFILE @ A GLANCE

Influencing standards



THE UML TESTING PROFILE @ A GLANCE

History Lessons

- RFP for a testing profile for UML 2.0 was initiated in 2003
- UML Testing Profile 1.0 was adopted by OMG in 2005
  - It was the first standardized UML 2.0 profile
  - It was released before UML 2.0 was finalized
- UML Testing Profile 1.1 was released in 2011
- UML Testing Profile 1.2 was released in 2012

  → **Why do we need a major revision?**

THE UML TESTING PROFILE @ A GLANCE

Towards UTP 2 – Reasons for a major revision

- UTP 1.0 was ahead of its time – in the meantime, things have changed
  - Incorporate experiences with model-based development and testing
  - Incorporate experiences of using UML and profiles
  - Incorporate new standards like ISO 29119 or ETSI ES 202 951 (MBT)
- Lack of/insufficiently elaborated concepts
  - Test design facility, test data values, test logging facility
- OMG policies to introduce new concepts in a minor revision are restrictive

UTP 2 is rather a technical modernization of the language instead of a reinvention of the wheel

Fraunhofer FOKUS

AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

- Test Language
- Arbitration Specifications
- Test Design Facility

3. What has not yet been said

WHAT CAN I DO WITH UTP 2?

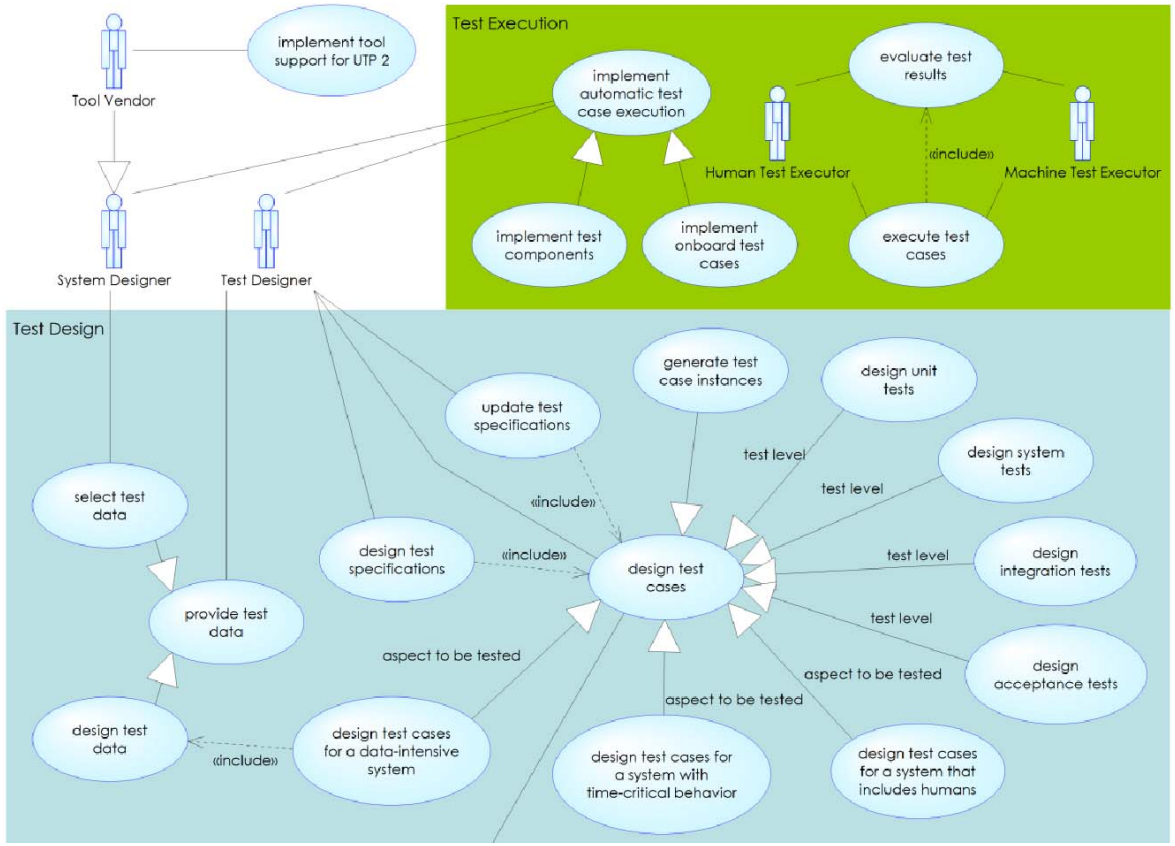
With UTP 2, I as a test engineer would want to

- Generate test cases, test data and test schedules automatically…
- Visualize test cases, test data and test schedules …
- Specify and reuse test environments …
- Capture test execution results for further test evaluation…
- Specify matching mechanisms for actual and expected responses…
- Specify arbitration rules for verdict calculation…
- Generate executable test scripts and test results for a dedicated target platform…
- Produce test reports in a desired format…
- Specify test (automation) architectures in a technology-independent manner…

… so that comprehensibility and communication among stakeholders are improved, important knowkledge is preserved and the degree of automation is increased

Fraunhofer FOKUS

WHAT CAN I DO WITH UTP 2?

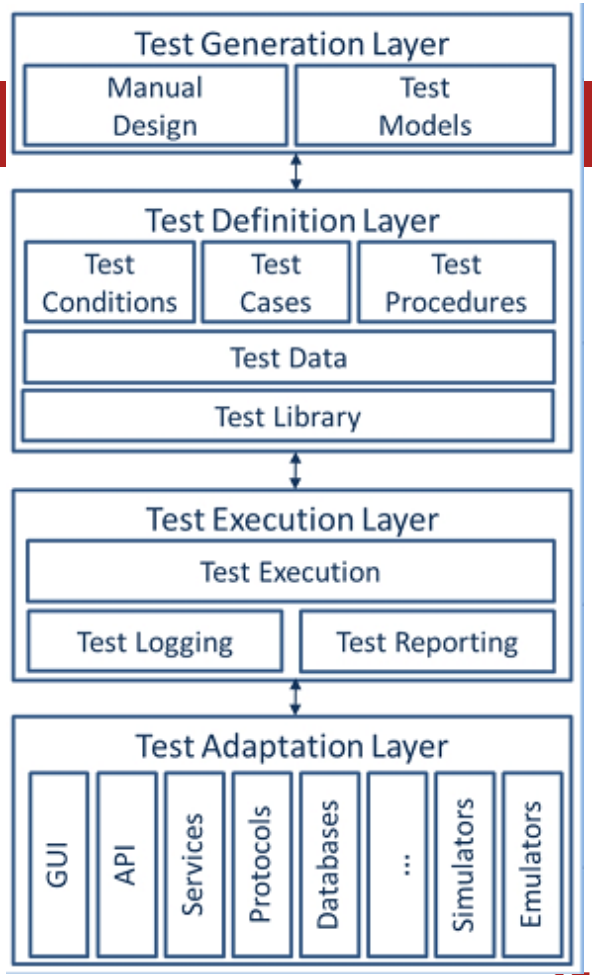


Fraunhofer FOKUS

WHAT CAN I DO WITH UTP 2?

Building test architectures with UTP 2

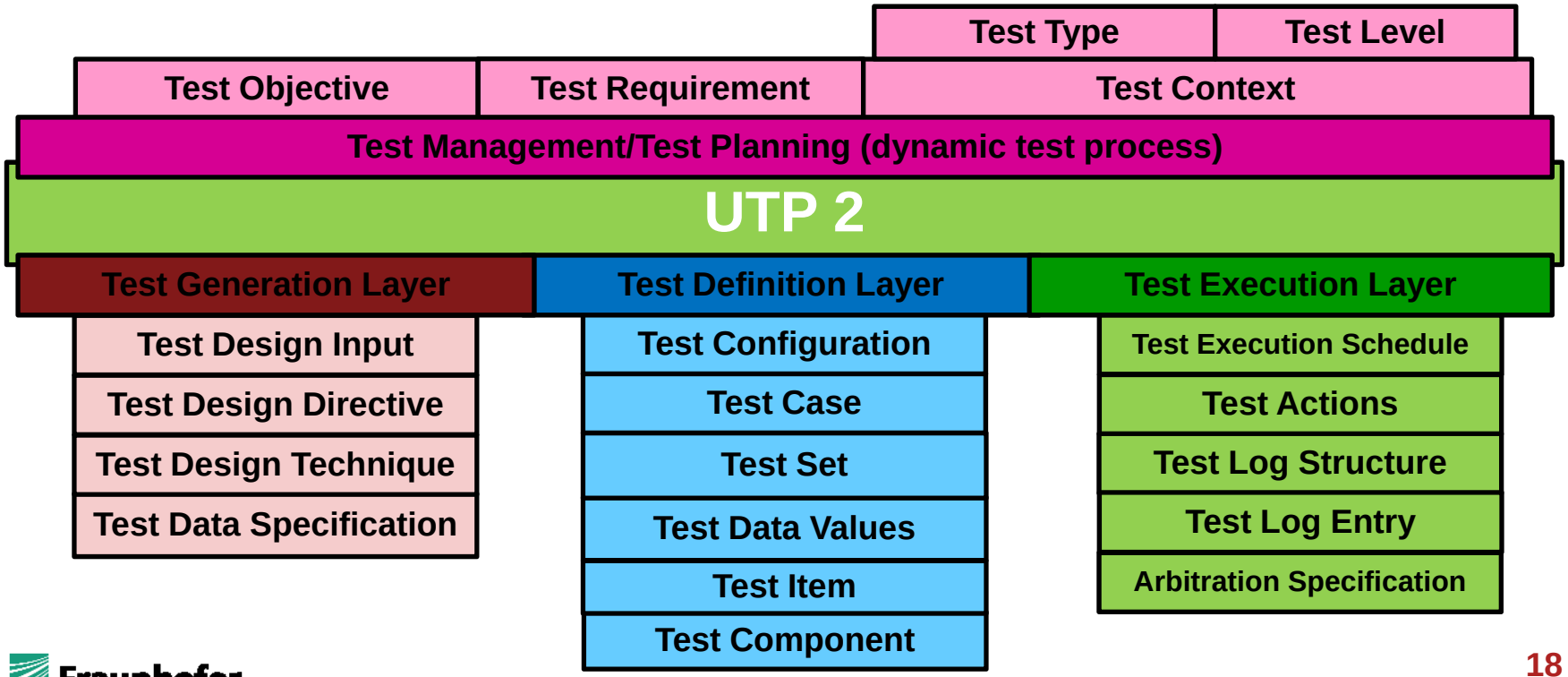
- Test Generation Layer
  Manual / automated design of test cases / test data
- Test Definition Layer
  Specification of test cases, test data, test procedures…
- Test Execution Layer
  Execution of test cases, logging of test execution, test evaluation & verdict arbitration
- Test Adaptation Layer
  Establishing communication with the system under test in order stimulate and observes it



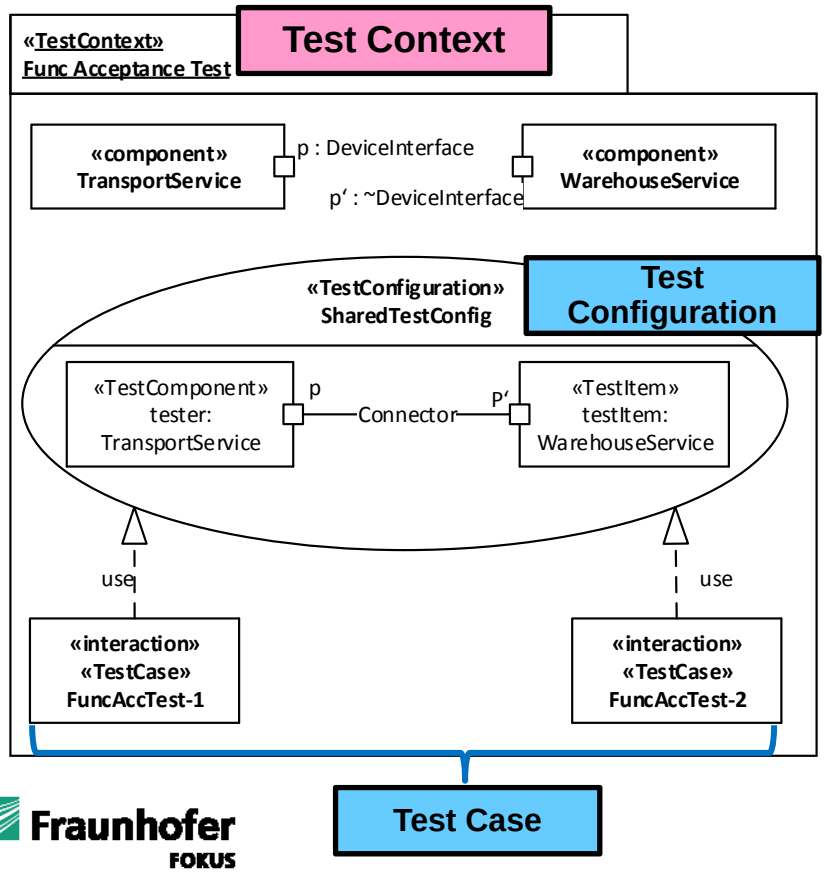
UTP 2 offers explicit concepts for the test generation, test definition and test execution layer

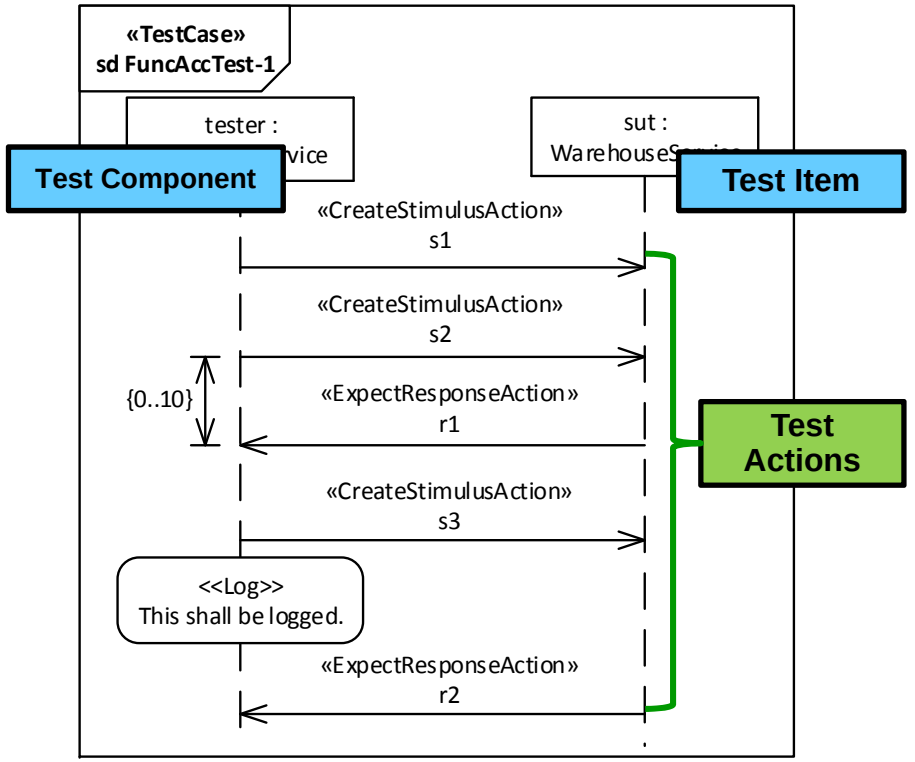
WHAT CAN I DO WITH UTP 2?

Conceptual overview



WHAT CAN I DO WITH UTP 2?





AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

- **Test Language**
- Arbitration Specifications
- Test Design Facility

3. What has not yet been said

WHAT CAN I DO WITH UTP 2?

Dedicated test actions in UTP 2

- Each UTP 2 **test case** consists of a **test procedure**
- A **test procedure** consists of **procedural elements**
- Procedural elements can be **atomic or non-atomic**
- A special kind of atomic procedural actions are **test actions**

  > *„An atomic procedural element that is an instruction to the tester that needs to be executed as part of a test procedure within some time frame.” [UTP2]*

- „Tester“ stands for both automated (i.e., test components, test drivers, test stubs) and manual testers

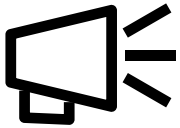
WHAT CAN I DO WITH UTP 2?

Overview: Test actions

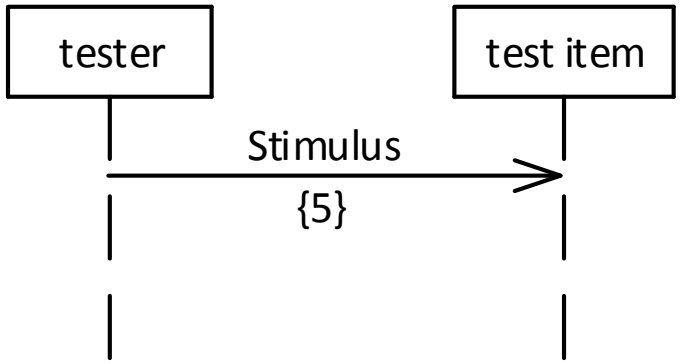
- A tester may want to
  - Stimuly the test item → **Create Stimulus Action**
  - Observe an expected response → **Expect Response Action**
  - Check some internal properties of the test item → **Check Property Action**
  - Submit a verdict to the arbitration specification → **Suggest Verdict Action**
  - Write something into the test log → **Create Log Entry Action**

WHAT CAN I DO WITH UTP 2?

Test action: Create Stimulus Action



„A test action that instructs the tester to submit a stimulus (potentially including test data) to the test item."



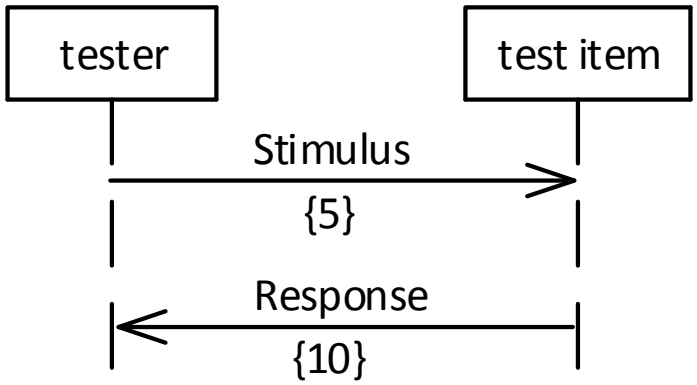
TTCN-3 equivalent

```
tester.send(Stimulus:{5});
```

WHAT CAN I DO WITH UTP 2?

Test action: Expect Response Action

"A test action that instructs the tester to check the occurrence of one or more particular responses from the test item within a given time window."



TTCN-3 equivalent

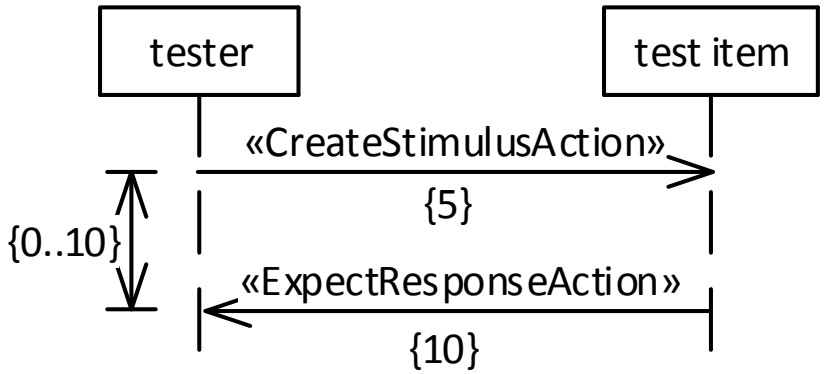
```
tester.send(Stimulus:{5});

tester.receive(Response:{10})
```

WHAT CAN I DO WITH UTP 2?

Test action: Expect Response Action (with timing)

"A test action that instructs the tester to check the occurrence of one or more particular responses from the test item within a given time window."



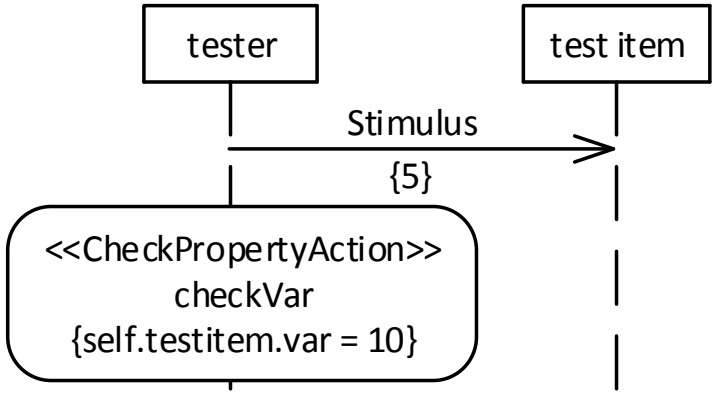
TTCN-3 equivalent

```
tester.send(Stimulus:{5});
timer.start(10);
alt{
[]tester.receive(Response:{10})
        {…}
[]timer.timeout
        {…}
}
```

WHAT CAN I DO WITH UTP 2?

Test action: Check Property Action

„A test action that instructs the tester to check the conformance of a property of the test item.“



TTCN-3 equivalent

Function or external function

```
function boolean f_checkVar(integer
expectedVar){…}
```

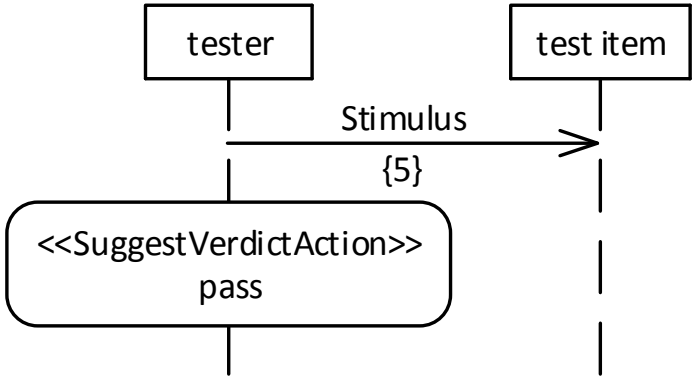
Invocation in the test procedure

```
tester.send(Stimulus:{5});
var b := f_checkVar(10));
```

WHAT CAN I DO WITH UTP 2?

Test action: Suggest Verdict Action

„A test action that instructs the tester to suggest a particular test action verdict to the arbitration specification of the test case for being taken into account in the final test case verdict.“



TTCN-3 equivalent

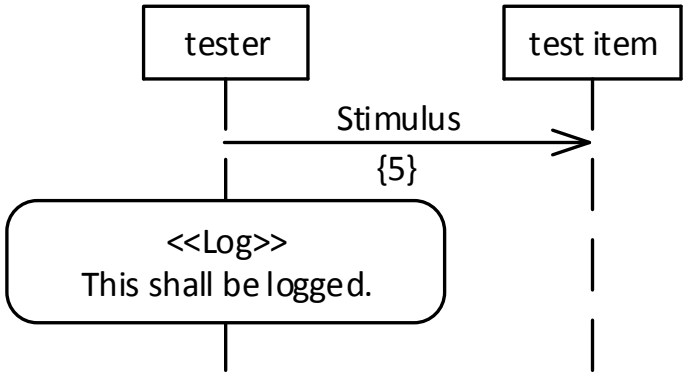
```
tester.send(Stimulus:{5});

setverdict(pass);
```

WHAT CAN I DO WITH UTP 2?

Test action: Create Log Entry Action

„A test action that instructs the tester to record the execution of a test action, potentially including the outcome of that test action in the test case log.“



TTCN-3 equivalent

```
tester.send(Stimulus:{5});

log(„This shall be logged“);
```

WHAT CAN I DO WITH UTP 2?

Test actions summary

- UTP 2 provides a basic set of dedicated test actions for the creation of test cases
- Support for **API-based observation and control**
  - Create stimulus action, expect response action
- Support for **unspecified observation**
  - Check property action
- Support for **test tool communication**
  - Suggest verdict action, create log entry action

AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

- Test Language
- **Arbitration Specifications**
- Test Design Facility

3. What has not yet been said

WHAT CAN I DO WITH UTP 2?

Introduction: Arbitration Specifications

- An arbitration specification is a **specification** of the rules reasoning about verdicts
- Arbitration specifications can be defined for **test sets**, **test cases** and **test actions**
  - If no arbitration specification is set, a **default** one is set
  - Arbitration specifications can be **replaced** for certain test actions, test cases and test sets
- Arbitration specifications help keeping the test cases and test procedures **agnostic** of any verdict-related information

WHAT CAN I DO WITH UTP 2?

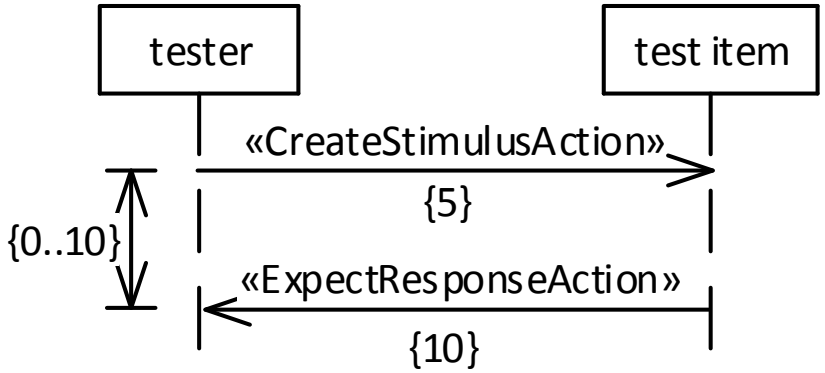
Interplay of arbitration specifications on different levels

- Test action, test case and test set represent ascending composition levels
- Each arbitration specification provides a verdict
  - Test action AS → test action verdict
  - Test case AS → test case verdict
  - Test set AS → test set verdict
- Test action verdicts result from the evaluation of atomic test actions
- Test action verdicts are conveyed to the test case AS that is responsible to calculate the test case verdict
- Test case verdicts are conveyed to the test set AS (if set) that is responsible to calculate the test set verdict

WHAT CAN I DO WITH UTP 2?

Arbitration Specifications

– What is the semantics of the following test case with respect to its verdict?



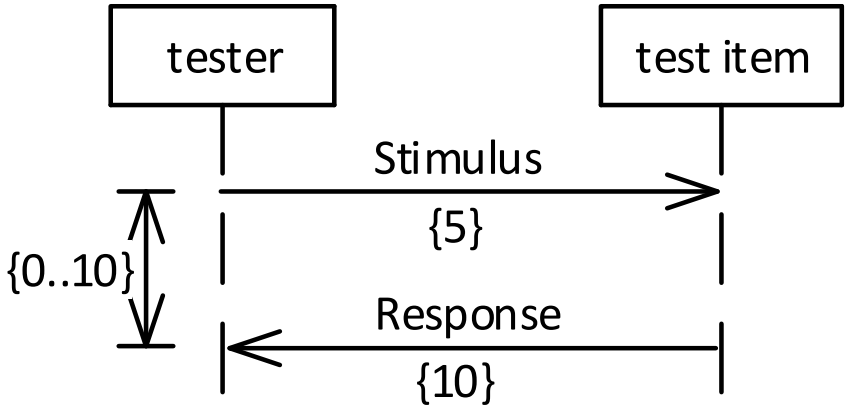
Semantics is given by the applied **arbitration specification**.

Questions

- What verdict shall be set, if the expected response is received?
- What verdict shall be set, if another response is received before?
- What is the initial verdict of the test case?
- Is there a precedence rule of verdicts similar to TTCN-3 (i.e., none < pass < inconclusive < fail < error)?

WHAT CAN I DO WITH UTP 2?

A Simple Arbitration Specification



TTCN-3 equivalent

```
tester.send(Stimulus:{5});
timer.start(10);
alt{
     []tester.receive(Response:{10})
        {setverdict(pass);}
     []timer.timeout
        {setverdict(fail);}
}
```

WHAT CAN I DO WITH UTP 2?

A Revised Arbitration Specification

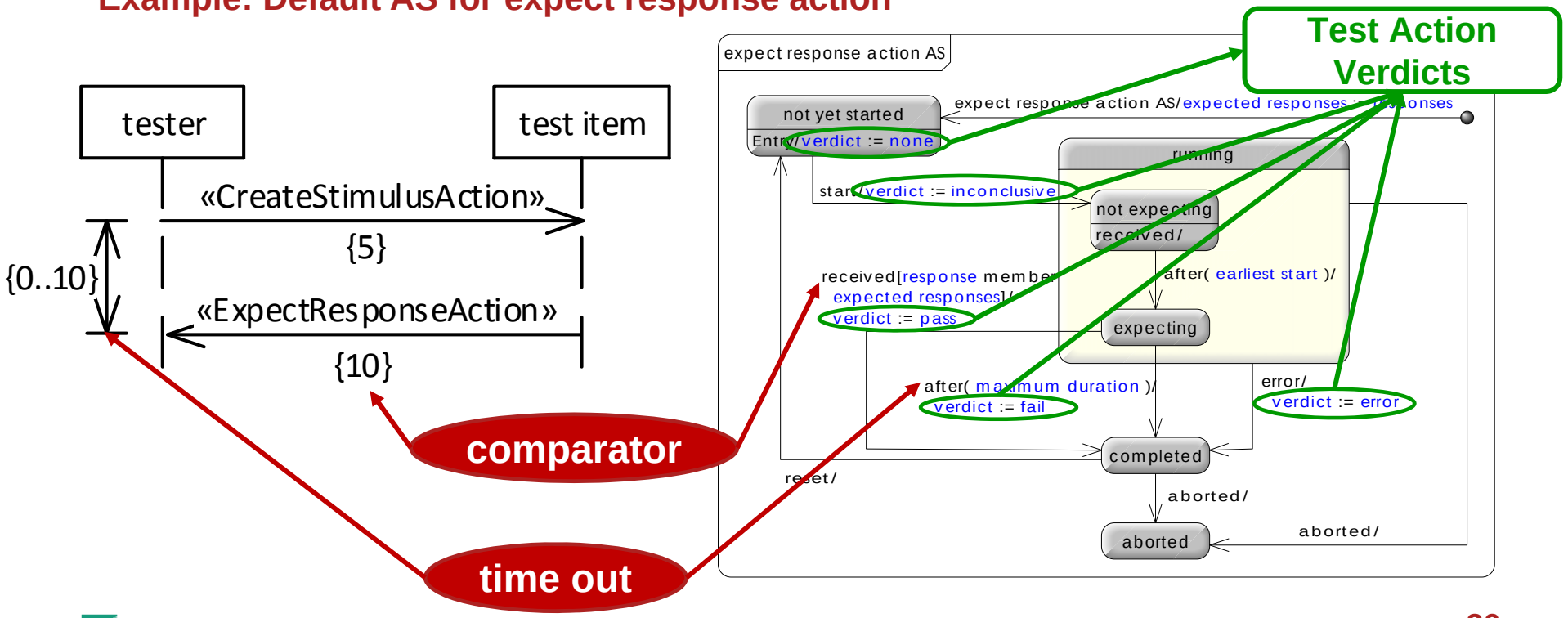
- Let us assume we exchange the AS of the expect response action with the following algorithm:
  - If the actual response matches the expected response, the verdict is *pass*
  - If the actual response does not match the expected response, the verdict is *fail*
  - It the duration constraint is violated, the verdict is *inconclusive*

TTCN-3 equivalent

```
tester.send(Stimulus:{5});
timer.start(10);
alt{
[]tester.receive(Response:{10})
        {setverdict(pass);}
    []tester.receive
        {setvderdict(fail);}
    []timer.timeout
        {setverdict(inconc);}
}
```

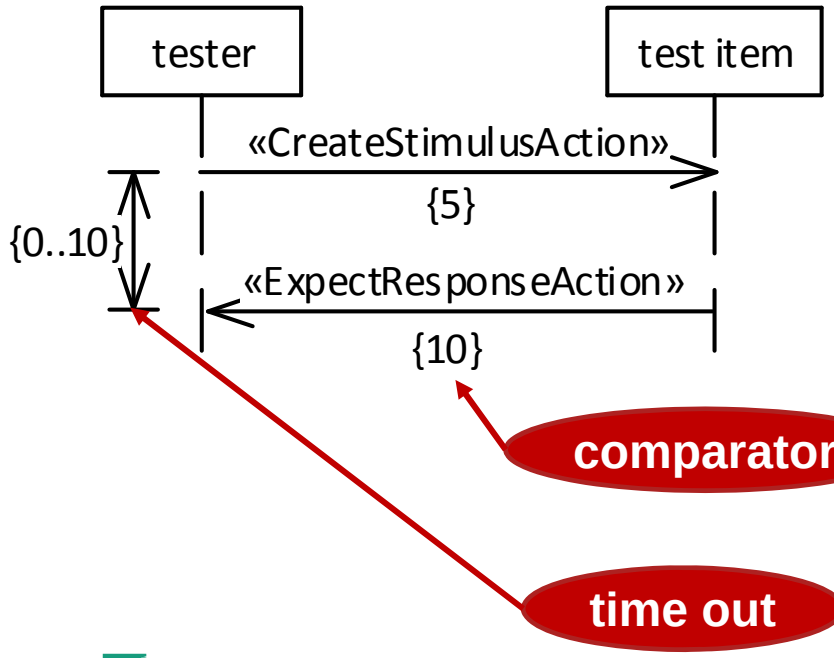
WHAT CAN I DO WITH UTP 2?

Example: Default AS for expect response action



WHAT CAN I DO WITH UTP 2?

Example: Default AS for expect response action



TTCN-3 equivalent

```
tester.send(Stimulus:{5});
timer.start(10);
alt{
     []tester.receive(Response:{10})
        {setverdict(pass);}
     []tester.receive
        {repeat;}
     []timer.timeout
        {setverdict(fail);}
}
```

WHAT CAN I DO WITH UTP 2?

Summary: Arbitration Specifications

- Arbitration specifications have been **newly introduced** (in contrast to Arbiter) by UTP 2
- Help keeping the test case **clean** of verdict/arbitration-related logic
- UTP 2 **provides default** arbitration specification for test actions, test case and test sets
  - If no explicit arbitration is given, the default one will be taken by definition
  - Easy replacement of arbitration specifications through tagged values
- Arbitration specifications **do not have** to be expressed in a formal, yet executable way → but UTP 2 provides a formal semantics for ist default AS

AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

- Test Language
- Arbitration Specifications
- **Test Design Facility**

3. What has not yet been said

WHAT CAN I DO WITH UTP 2?

Essentials of test design

- Test design activities derive an **output** (a set of test artifacts, e.g., test case, test data etc.) from an **input** (state machine, class diagram, etc.)
- The input is called **test model** (in ISO 29119 and ISTQB)
- Test design is **directed** by **test design techniques** (or coverage criteria or test goals…)
- Derivation depends on the **specific configuration** of a automated test generator or **instructions** for a human test generator
- **Capability/ability** of the test generator **limits** applicable test design techniques

UTP is (and was from the very beginning) intended to support **test design activities**

Fraunhofer FOKUS

WHAT CAN I DO WITH UTP 2?

CADP TGV

Spec Explorer

Smartesting
CertifyIT

MBTsuite
THE TESTING FRAMEWORK

ALL4TEC

Matelo

TestBench

tedeso

How to cope with the variety of test generators (including the human test generator)?

CEA Diversity

Graphwalker

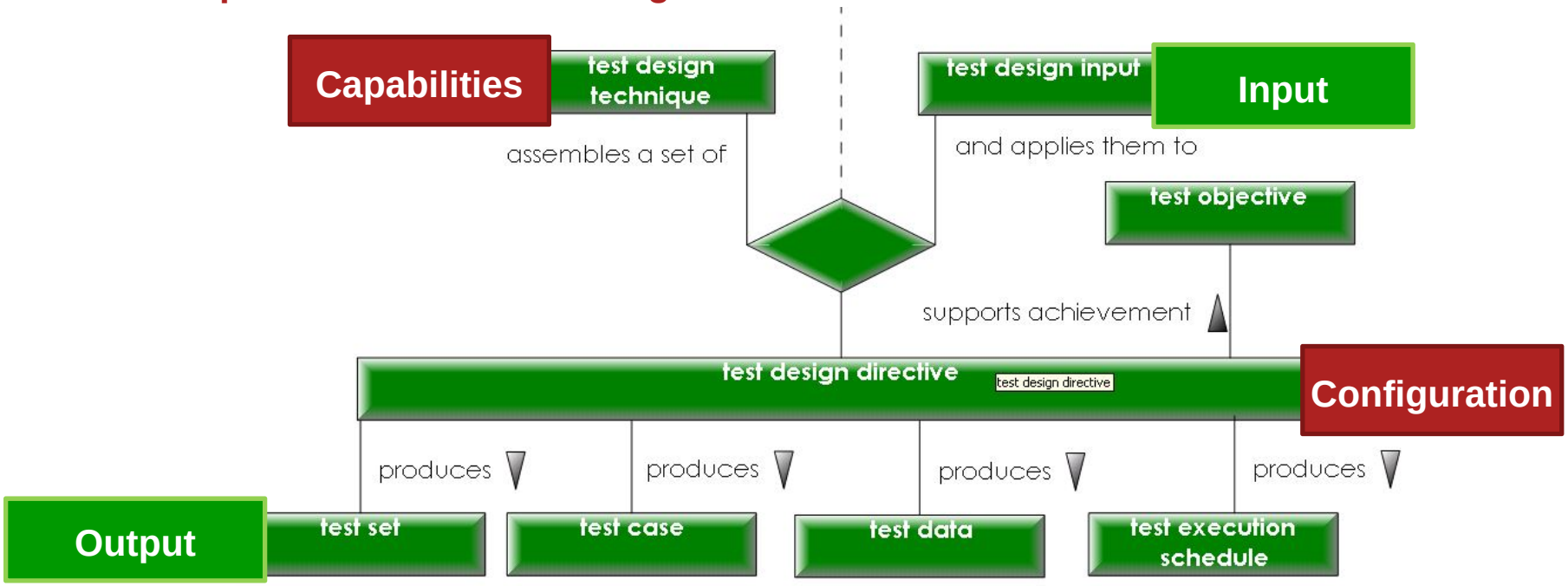
SRL TRUST

UGOE
AutoQuest

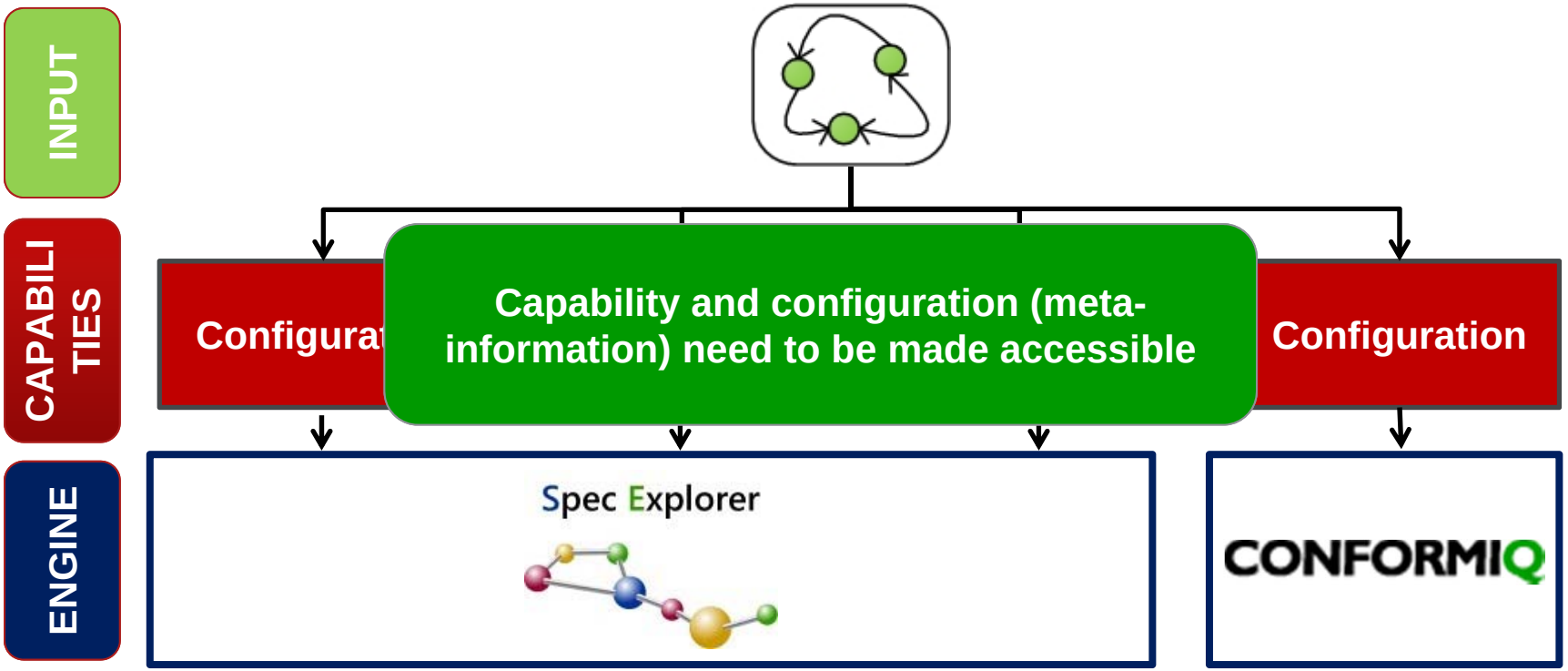
Fraunhofer
FOKUS

WHAT CAN I DO WITH UTP 2?

A conceptual model for test design

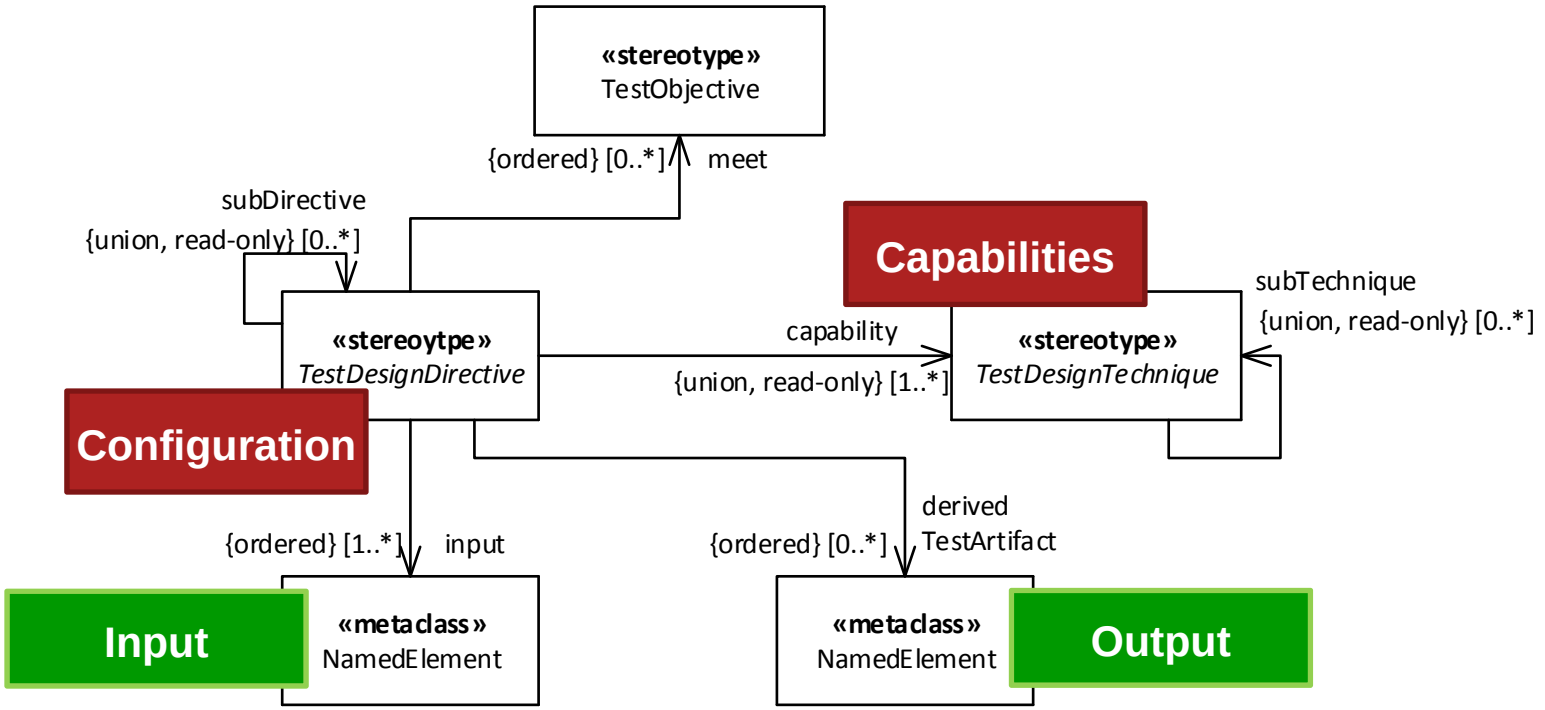


WHAT CAN I DO WITH UTP 2?



WHAT CAN I DO WITH UTP 2?

The UTP 2 test design facility implementation



WHAT CAN I DO WITH UTP 2?

Summary: Test design directive and test design technique

- Domain experts and test engineers **shall not be bothered** with complex **internals** of test generators
- Test design directives and techniques **decouple the semantics** of test design technqiues design **from dedicated implementations** of test generators
- Tool vendors can **publish and provide** the capabilities of their test generators in a standardized and integrated manner
- The test design facility is **easily extensible** to add further capabilities
- A set of **widely-used** test design techniques is offered out of the box (as library)

Fraunhofer FOKUS

AGENDA

1. The UML Testing Profile @ a Glance

2. What can I do with UTP 2?

3. What has not yet been said

WHAT HAS NOT BEEN SAID YET

Relation to ETSI‘s Test Description Language cont.

- ETSI Test Description Language is a MOF-based metamodel that supports the definitions of test purposes and test descriptions
  - A test purpose is semantically equivalent to a test requirement
  - A test description is semantically equivalent to a sequence diagram, but by definition not-executable, because it resides on a higher-level of abstraction
- TDL is/was mainly intended to be used as a language for test specifications as basis for test case implementations, but already on a (semi-)formal level
- Bridges the gap among natural language requirements and test purposes and executable TTCN-3 test case implementations

WHAT HAS NOT BEEN SAID YET

Relation to ETSI‘s Test Description Language cont.

- UTP and TDL share a common idea of test modeling
  - TTCN-3 influenced UTP and (years later) UTP/UML re-influenced TDL
- TDL has no support for
  - Test directives and test design techniques
  - Arbitration specifications
  - Test logs
  - Deployment specifications
- ETSI has worked on a standardized UML profile for TDL

→ If you are interested in TDL, visit the upcoming UCAAT conference at Fraunhofer FOKUS premises

Fraunhofer FOKUS

WHAT HAS NOT BEEN SAID YET

Timeline and Roadmap

- **June 2017**: Submission of revised submission; adoption by OMG as alpha standard expected; charter of finalization task force expected
- **June 2018**: Submission of FTF; release of UTP 2.0 by OMG expected; charter of UTP 2.1 revision task force
- **June 2019:** Release of UTP 2.1 expected

WHAT HAS NOT BEEN SAID YET

Summary

- UTP 2 is a graphical modelling language based on ÚML
- A graphical modelling language to support test design activities
- Terminology in particular influenced by ISO 29119 and ISTQB
- Concepts provided to describe (parts of) test automation architectures
- Just a specification technique! Transformations not part of UTP 2

WE ARE OVER AND DONE…

Thank you very much for your attention.

Questions?

CONTACT

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
www.fokus.fraunhofer.de

Marc-Florian Wendland
Senior Researcher, SQC
marc-florian.wendland@fokus.fraunhofer.de
Phone +49 (0)30 3463-7395

REFERENCES

[1] Object Management Group (OMG), Unified Modeling Language (UML), http://ww.omg.org/spec/UML

[2] International Software Testing Qualifications Board (ISTQB), GISTQB Glossary V3.01, http://www.istqb.org/downloads/finish/20/206.html

[3] Internation Standardisation Organisaton (ISO), ISO 29119, Software and systems engineering - Software testing

[4] Object Management Group, UML Testing Profile (UTP), http://ww.omg.org/spec/UTP

[5] European Telecommunications Standardisation Institute (ETSI): ES 202 951: Requirements for Modeling Notations. ETSI Standard, Methods for Testing and Specification (MTS); Model-Based Testing (MBT). V1.1.1 (2011-07)

[6] Wendland, Marc-Florian, Abstractions on Test Design Techniques, in Proc.: Federated Conference on Computer Science and Information Systems (FedCSIS), 2014

[7] Object Management Group (OMG), Request for Proposal for UML Testing Profile 2.0, Doc#: ad/2013-11-02

[8] Wendland, Marc-Florian, Towards UTP 2, presented at 2nd User Conference on Advanced Automated Testing (UCAAT), 2014