

TEST SYSTEM ARCHITECTURES USING ADVANCED STANDARDIZED TEST LANGUAGES

Axel Rennoch

InSTA 2016, Chicago, 10th April 2016

CONTENTS

- **Introduction**
- **Advanced standardized test languages**
 - TTCN-3
 - UML testing profile
 - TDL
- **Test Automation Architecture**
- **Conclusions**

AXEL RENNOCH

- **Researcher at Fraunhofer FOKUS in Berlin, Germany**
- **User of standardized Modelling and Test languages**
 - LOTOS, SDL/UML
 - Tree and Tabular Combined notation (TTCN-1&2)
- **TTCN-3 user from the beginning**
 - Developer for ETSI, 3GPP and industry
 - Trainer and consultant
 - Research projects and evolution team

MOTIVATION

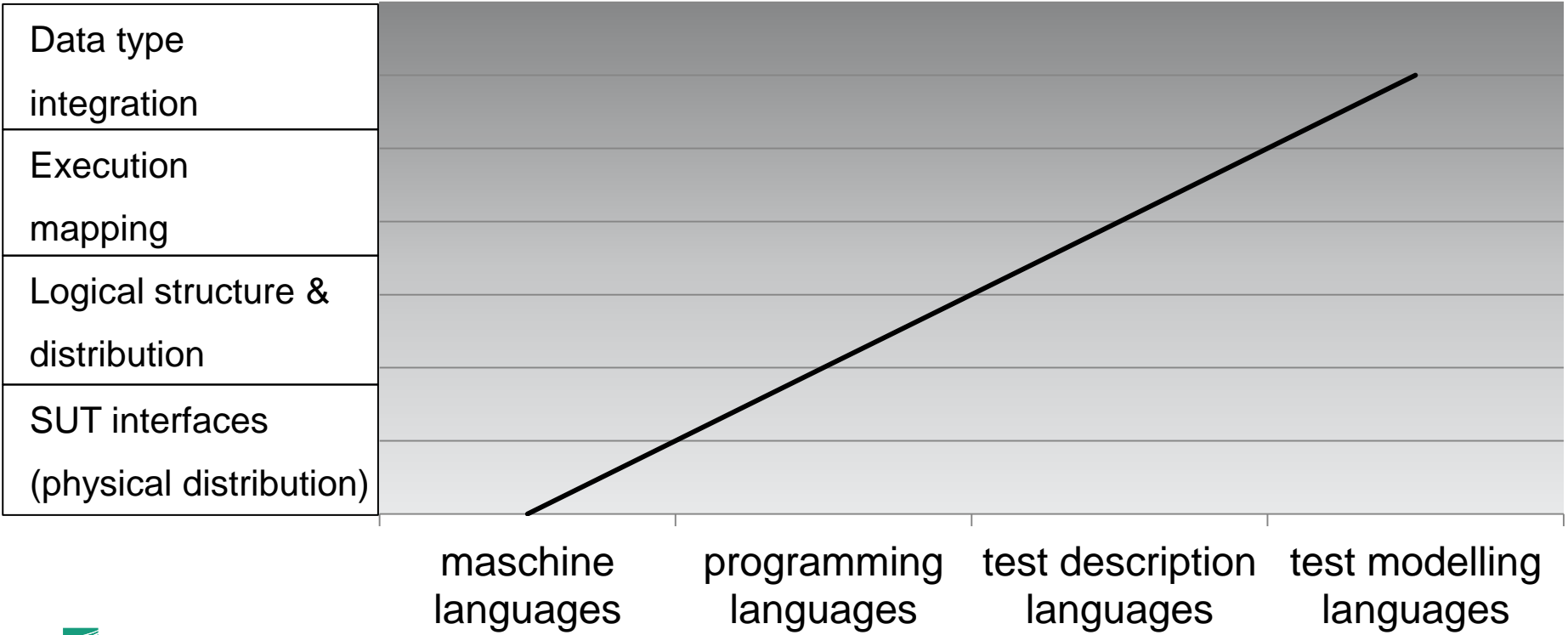
InSTA 2015:

- Keynote by **Sigrid Eldh** on
 - **Software Test Architecture**
 - definition & aspects

InSTA 2016:

- Continuation on **Advanced Standardized Test Languages** for
 - **Test System Architecture**
 - means & examples, user perspective

CHALLENGES FOR TEST SYSTEMS



SOME INITIAL QUESTIONS FOR TEST DEVELOPERS

- Identify SUT access interface points
- Test objectives: functional, load, security?

- Target :
 - standardization body (more abstraction, top down) or
 - in-house test solution (address concrete tools, bottom up)

- Parallel test components (scalability?)
- Coordination & synchronization (existing frameworks?)

CONTENTS

- Introduction

- Advanced standardized test languages
 - **TTCN-3**
 - UML testing profile
 - TDL

- Test Automation Architecture

- Conclusions

WHAT IS TTCN-3?

- Testing and Test Control Notation
- Internationally standardized testing language for formally defining test scenarios. Designed purely for testing

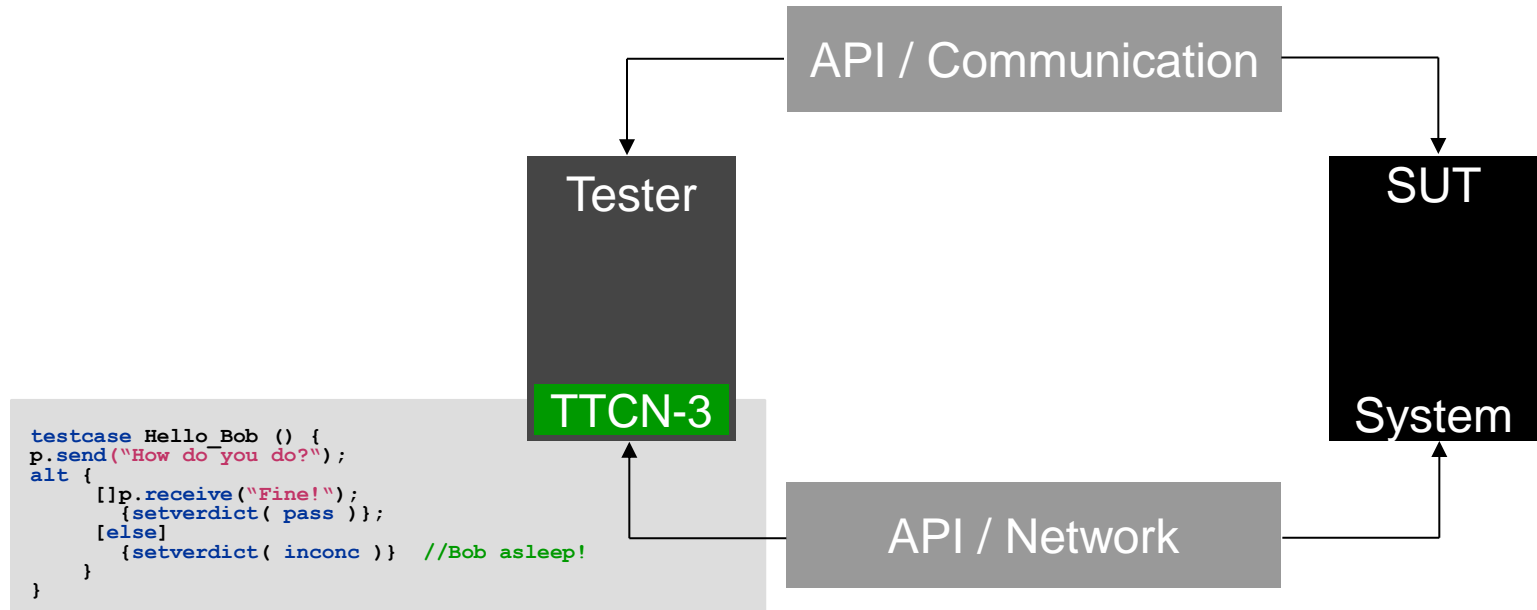
```
testcase Hello_Bob () {
p.send("How do you do?");
alt {
  []p.receive("Fine!");
      {setverdict( pass )};
  [else]
      {setverdict( inconc )} //Bob asleep!
}
}
```


IDEA & DESIGN PRINCIPLES OF TTCN-3

- **One test technology for different tests**
 - Distributed, platform-independent testing
 - Integrated graphical test development, -documentation and –analysis
 - Adaptable, open test environment

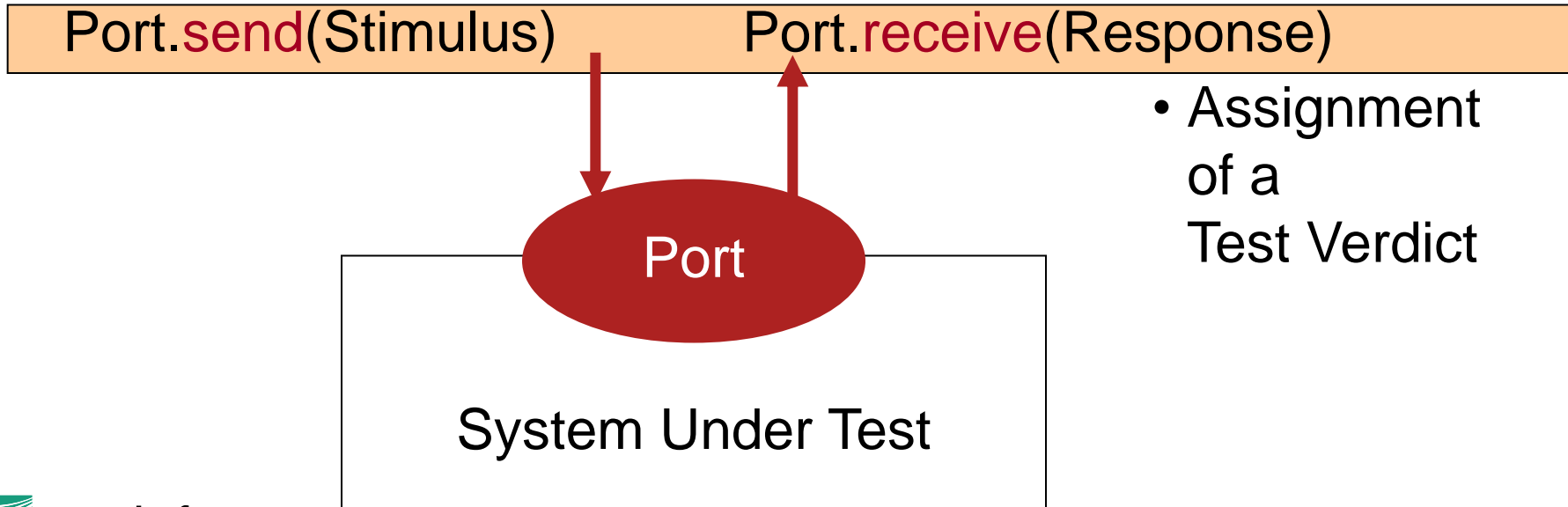
- **Areas of Testing**
 - Regression Testing
 - Conformance and Functionality Testing
 - Interoperability and Integration Testing
 - Load/ Stress Testing

TTCN-3 EXECUTION



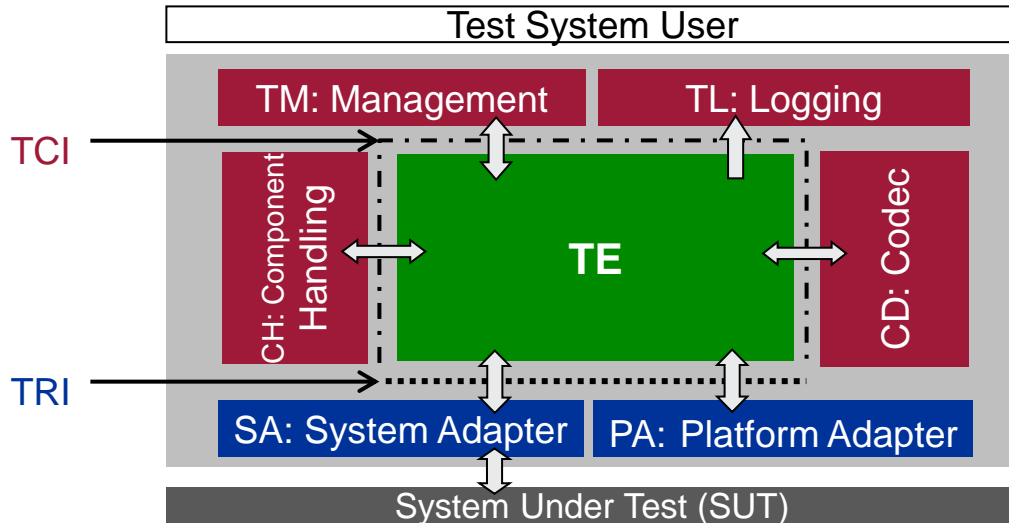
TTCN-3 IS DESIGNED FOR BLACK-BOX TESTING

TTCN-3 Test Case



- Assignment of a Test Verdict

A TTCN-3 TEST SYSTEM

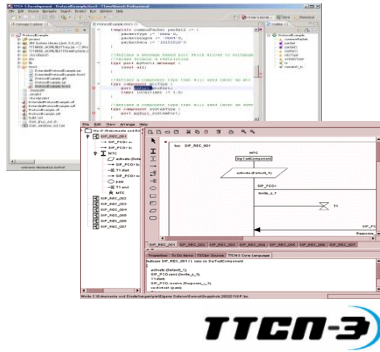


- TE – TTCN-3 Executable
- TM – Test Management
- TL – Test Logging
- CD – Codec
- CH – Component Handling
- SA – System Adapter
- PA – Platform Adapter
- SUT – System Under Test

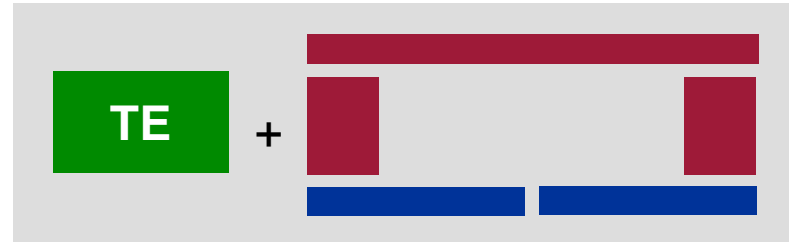
- ETSI ES 201 873-1 TTCN-3 Core Language (CL)
- ETSI ES 201 873-5 TTCN-3 Runtime Interface (TRI)
- ETSI ES 201 873-6 TTCN-3 Control Interfaces (TCI)

IMPLEMENTATION

ATS

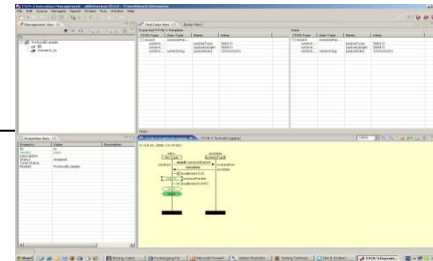


Test System



SUT

Communication



MAJOR LANGUAGE ELEMENTS OF TTCN-3 NOTATION

module definitions	
Imports	Importing definitions from other modules defined in TTCN-3 or other languages
Data Types	User defined data types (messages, PDUs, information elements, ...)
Test Data	Test data transmitted/expected during test execution (templates, values)
Test Configuration	Definition of the test components and communication ports
Test Behavior	Specification of the dynamic test behavior

TTCN-3 DOMAINS: TELECOM

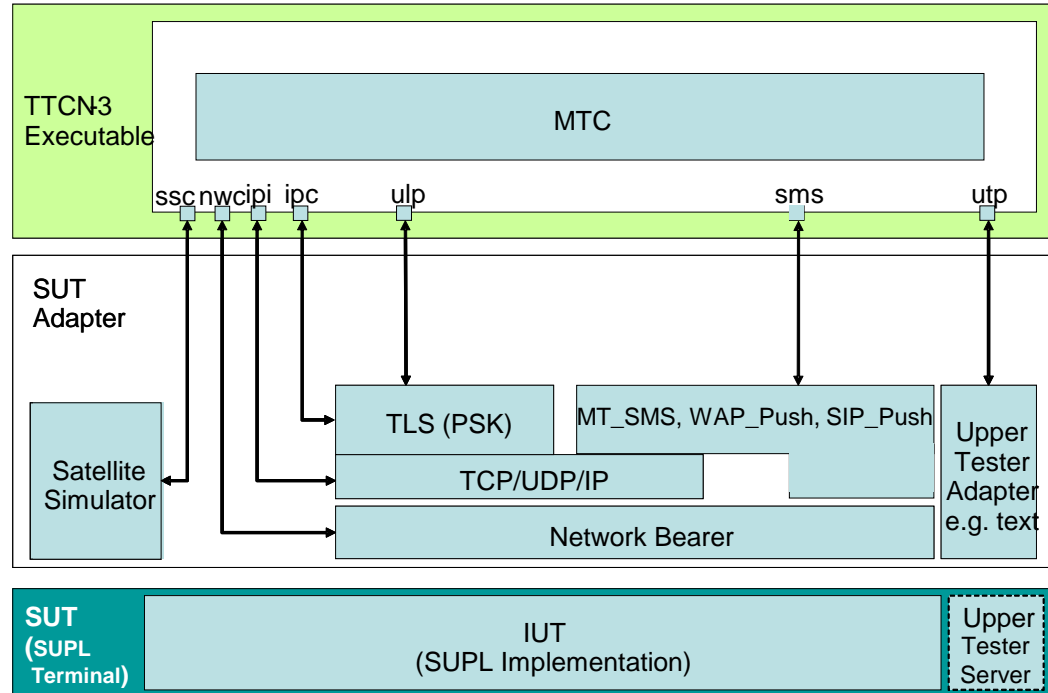
- Industrial use
 - Big companies with hundreds of TTCN-3 engineers: Ericsson, Nokia, Siemens, Motorola
 - large distribution among SME
- Standardization bodies
 - standardized test suites: ETSI / 3GPP (**LTE!**)/ OMA / TETRA and its members
 - IMS performance benchmark project: Intel, HP, BT, FOKUS and others
- Test tool manufacturer:
 - Commercial Tektronix, Catapult, Nexus, R&S, Spirent, ...
- Certification program based on TTCN-3: e.g. WiMax forum

TEST SYSTEM EXAMPLE 1: OMA SUPL

Secure UserPlane Location Protocol

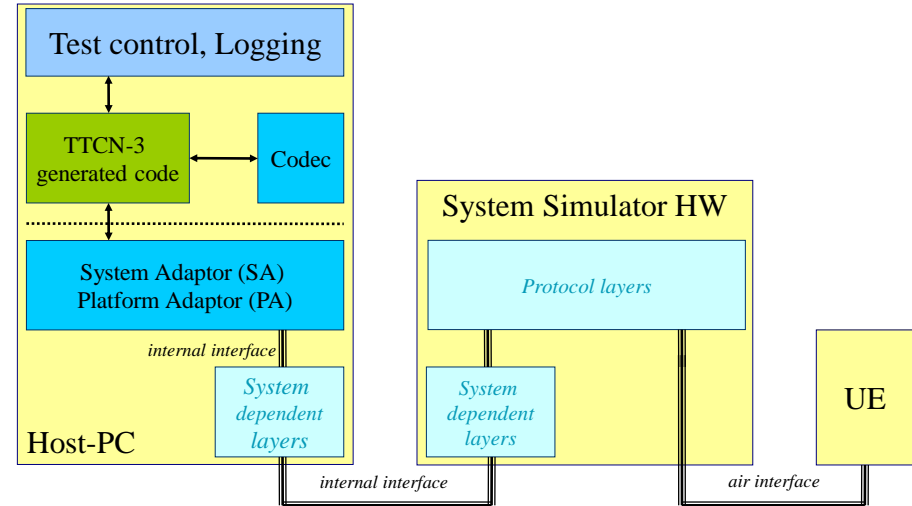
Single MTC controls e.g.:

- UlpPort (Lup interface)
- IpcPort (IP configuration)
- smsPort used for SMS
- UtpPort for upper tester commands
- IpiPort (IP information, e.g. release)
- NwcPort: network bearer control, e.g. handover trigger
- SscPort: satellite simulation control, e.g. scenario trigger



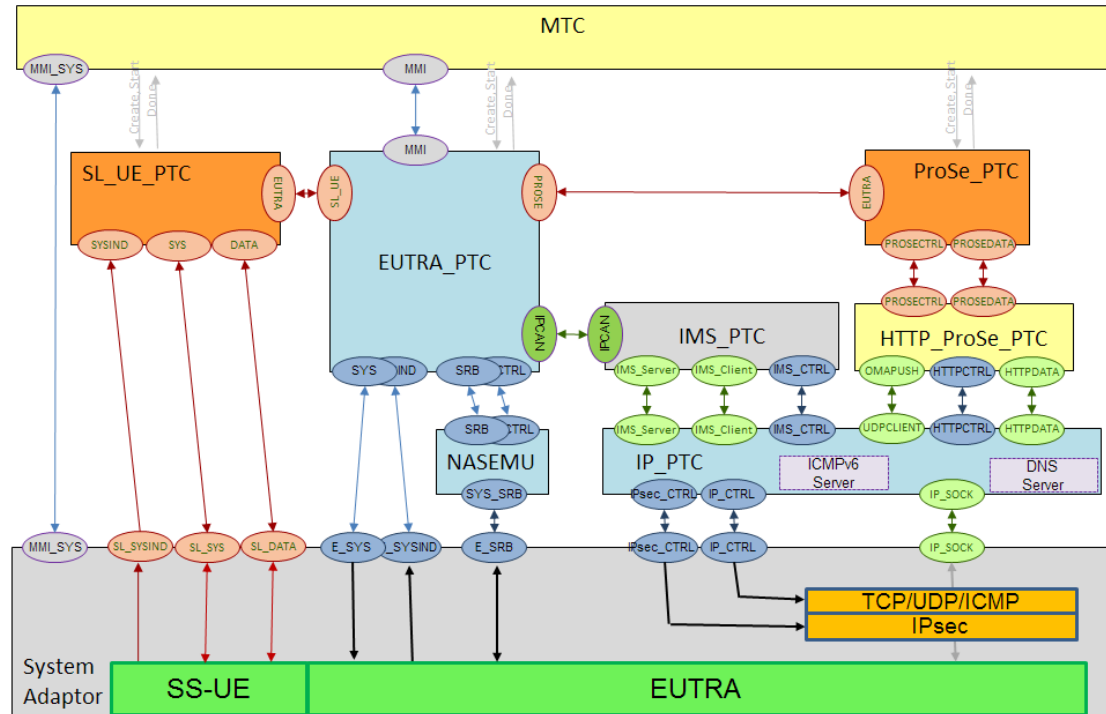
TEST SYSTEM EXAMPLE 2: 3GPP UE TESTING

- **E-UTRAN (LTE air interface):** 3GPP TS 36.523-3 V12.4.0 (2015-12)
- **Each radio access technology (RAT) is hosted by a separate TTCN-3 parallel component:** E-UTRAN, UTRAN, GERAN, others.
- **PTCs are controlled by the TTCN-3 main test component (MTC) which:**
 - is independent from the RAT;
 - may host the upper tester for MMI and AT commands;



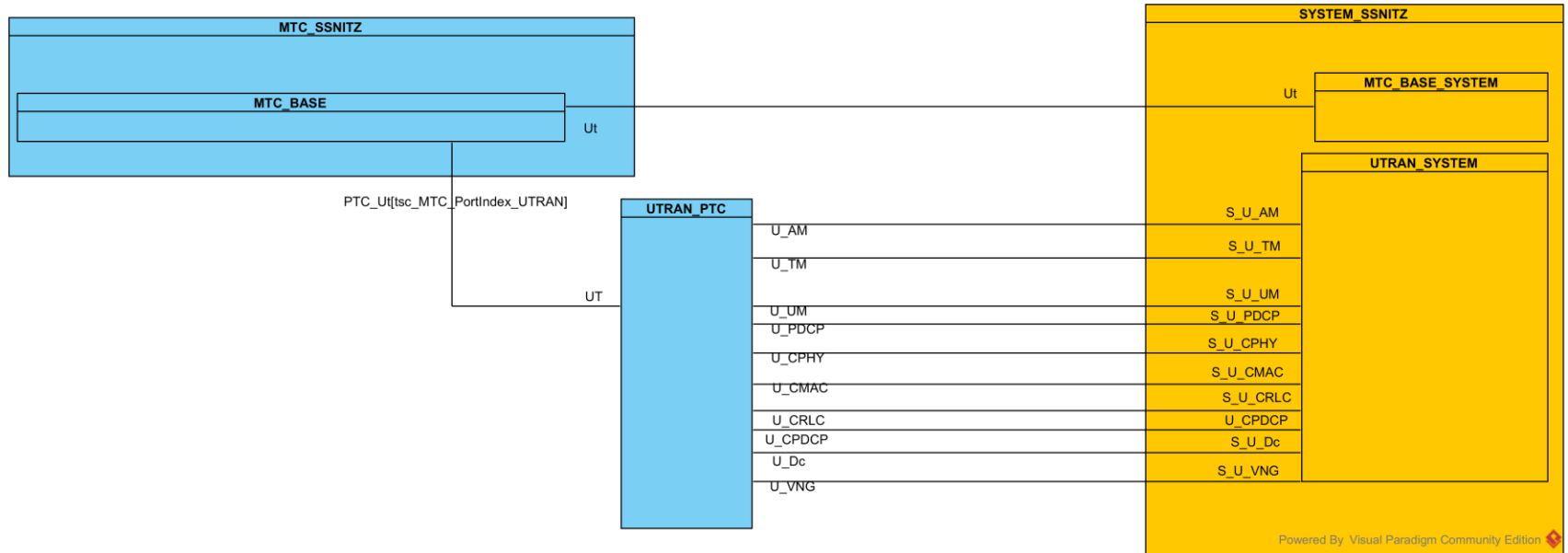
TEST SYSTEM EXAMPLE 2: 3GPP UE TESTING

- Multiple of configurations
- Several PTCs
 - Cooperating
 - Communicating



TEST SYSTEM EXAMPLE 2: 3GPP UE TESTING

- Simple scenario
- Reuse of nested component types for MTC, PTCs and TSI

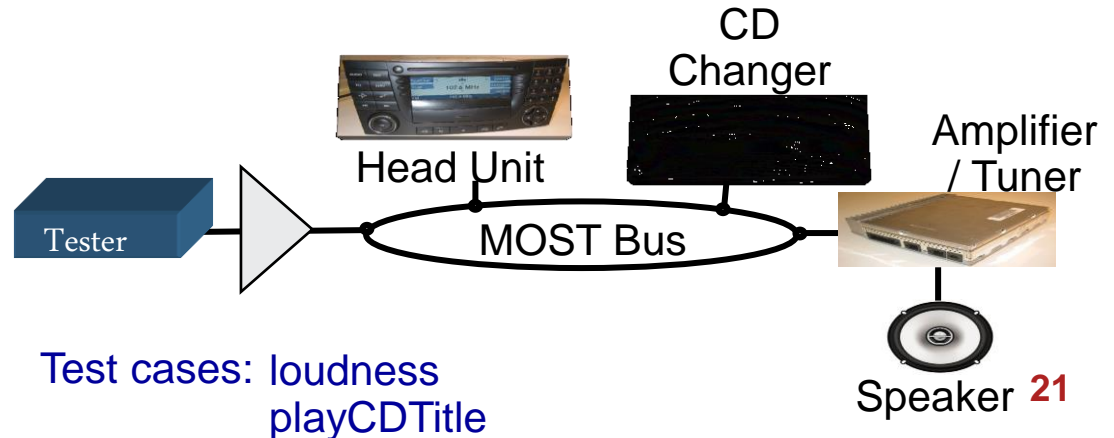


TTCN-3 DOMAINS: AUTOMOTIVE

- Car communication systems
 - Daimler, Volkswagen, SiemensVDO
 - edutainment bus system (test suite)
- Standardization groups:
 - AUTOSAR consortium
 - MOST cooperation
- Car-to-car communication

Telematics Applications in the Cockpit

- Audio (CD / Radio), Video
- Telephone, SMS
- Navigation
- Speech recognition
- User interface for body electronic



TTCN-3 DOMAINS: MEDICINE

Medicine

- SiemensMED (image processing)
- HL7 eHealth protocols (Interoperability)



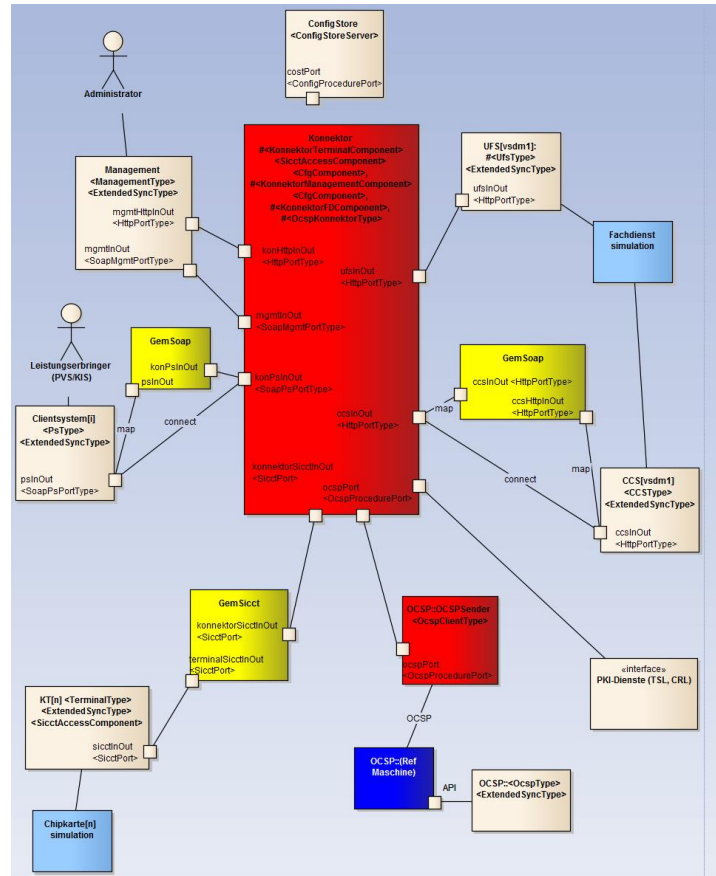
Upcoming E-Health infrastructure for Germany

- High security requirements (e.g. certificates, cryptography)
- Test development prior to SUT availability
- Multiple heterogenous interfaces:
 - cardterminals,, card simulations,
 - Webservices, OCSP server etc.



TEST SYSTEM EXAMPLE 3: E-HEALTH “KONNEKTOR”

- Complex configuration
- Illustration using IBM Rational Enterprise Architect:
 - PTC,
 - simulators,
 - SUT (mock),
 - real server,
 - adaptation code

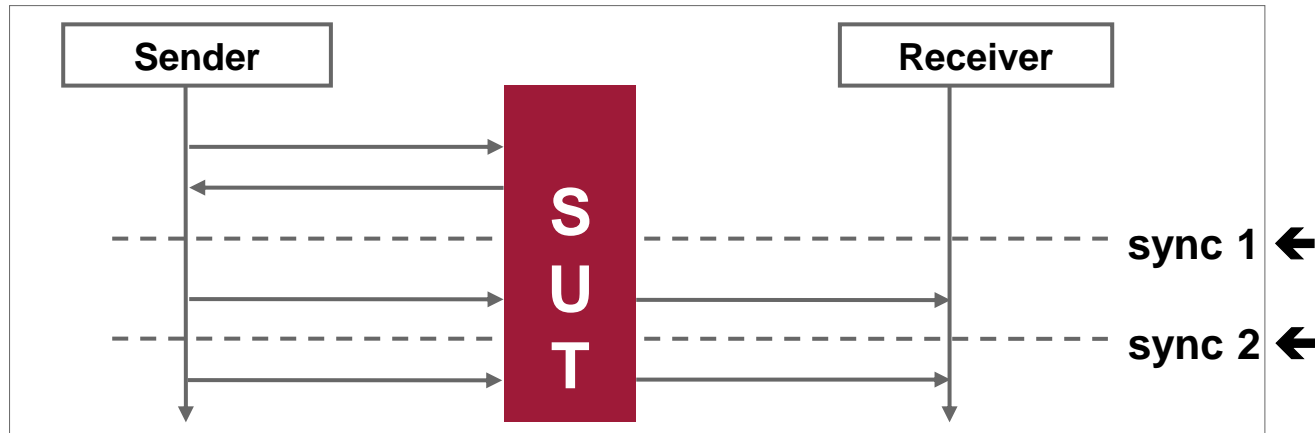


ETSI FRAMEWORK FOR SYNCHRONISATION

- **Source** STF 276 (IPv6 project)
- Set of **TTCN-3 functions** to e.g.
 - Start and control parallel components
 - Exchange synchronization signals between components
- Set of **charstring constants** for **synchronization points**
 - *preambleDone*, *sync1*, *sync2*
 - *testbodyDone*
- **Predefined timers** e.g. to avoid deadlocks at synchronization points
- <http://www.ttcn-3.org/index.php/development/devlibraries/devlib-libcommon>

SYNCHRONIZATION SAMPLES

- Synchronization occurs between parallel TTCN-3 components (using signals via MTC)



- Sync 1 ensures the completion of senders procedure
- Sync 2 confirms the arrival of a message at receivers side

MODULE IMPORT „LAYERING“

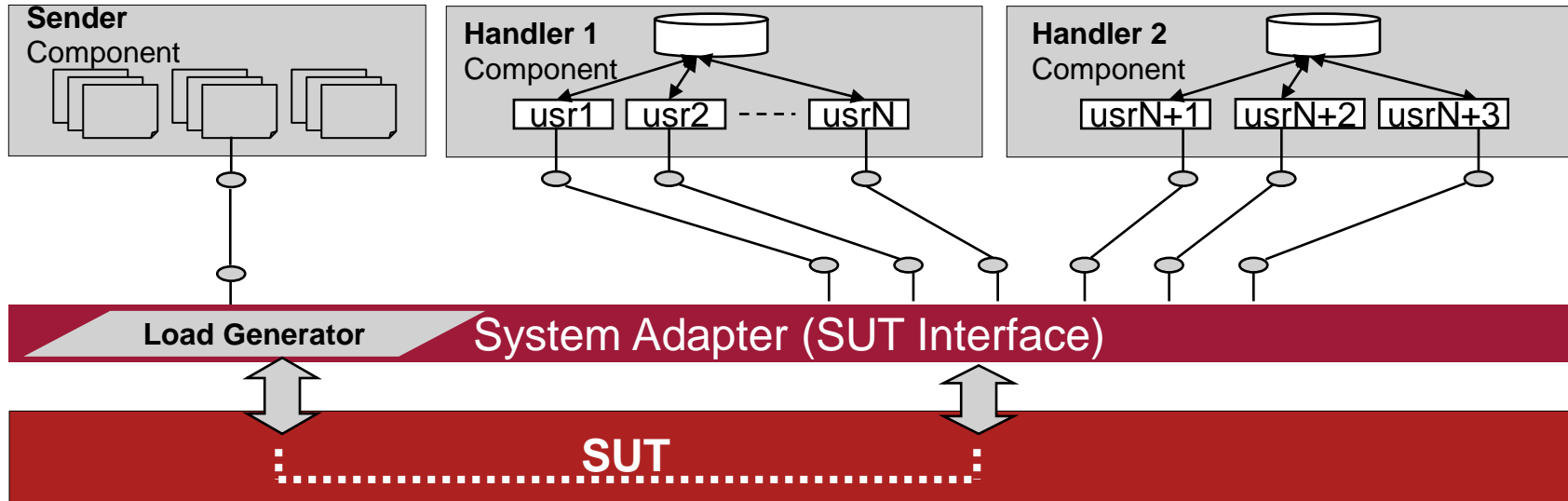
- Test suite specific code
 - testcase, test (component) functions
- Test suite specific library
 - Component types, test configuration, pre/postamble
- Interface/protocol specific library
 - SIP, DNS, IPv6 etc.
- Interface/protocol independent library
 - synchronizaton



high level:
(low
reusability)

Low level:
(High
reusability)

EXAMPLE 4: IMS BENCHMARK TEST CONFIGURATION



- 400 subscribers per component **Handler**
- Component **Sender** initializes/provides data requests for load-generator
- 5.000 - 10.000 IMS subscribers (per server)
- Up to 250 requests per second (per server)

LESSONS LEARNED WITH TTCN-3

- **Study access interface points (Test System Interface)**
 - Available Test solution plugins?
- **Test objectives: functional, load, security?**
 - Consider synchronization overhead (e.g. ETSI's LibCommon)
 - Consider performance issues (e.g. encoded data preparation)
- **Concurrent test components with separated traces & verdicts (easier failure analysis)**
 - Consider test tool logging functions
- **Improve decomposition using libraries for**
 - handling single interface types
 - Nested component types (to be extended)
- **Use modelling tools for the illustration of the TTCN-3 architecture**

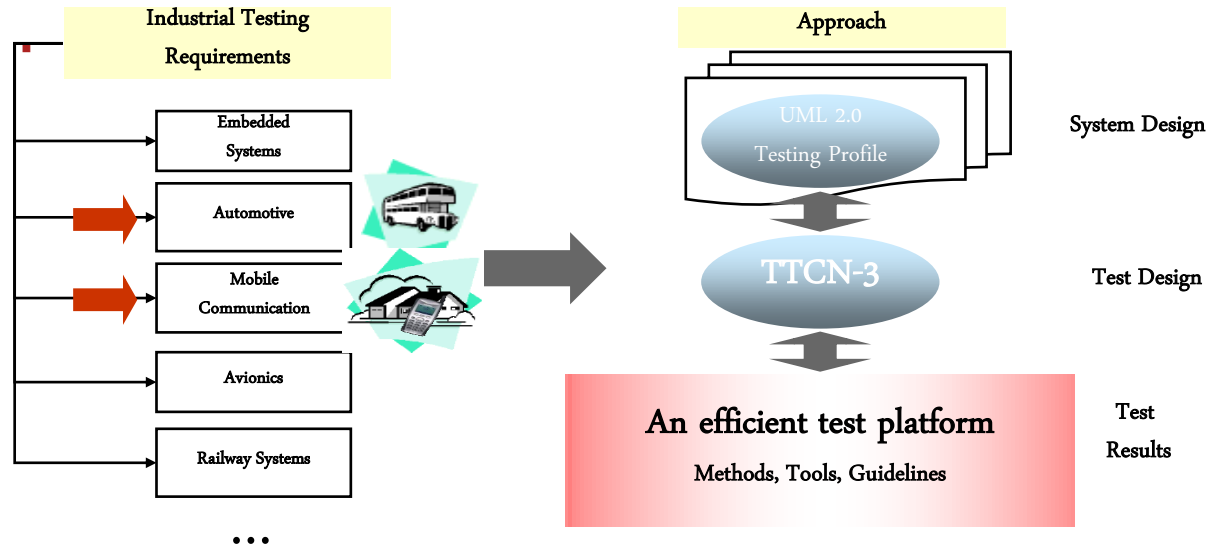
BEYOND TTCN-3: MBT

- TTCN-3 is used in several domains as binding link between modelling and execution
- Commercial tools do generate TTCN-3 code for test execution
 - lots of academic prototype tools
- Selection of industrial case studies:
e.g. European MIDAS project
 - pilots in SOA testing automation (*later*)

TTCN-3 LINK TO UTP

Objective:

- To develop an efficient **test platform** fulfilling **industrial testing requirements**
- To **execute high-level test models**, e.g. UML testing profile



CONTENTS

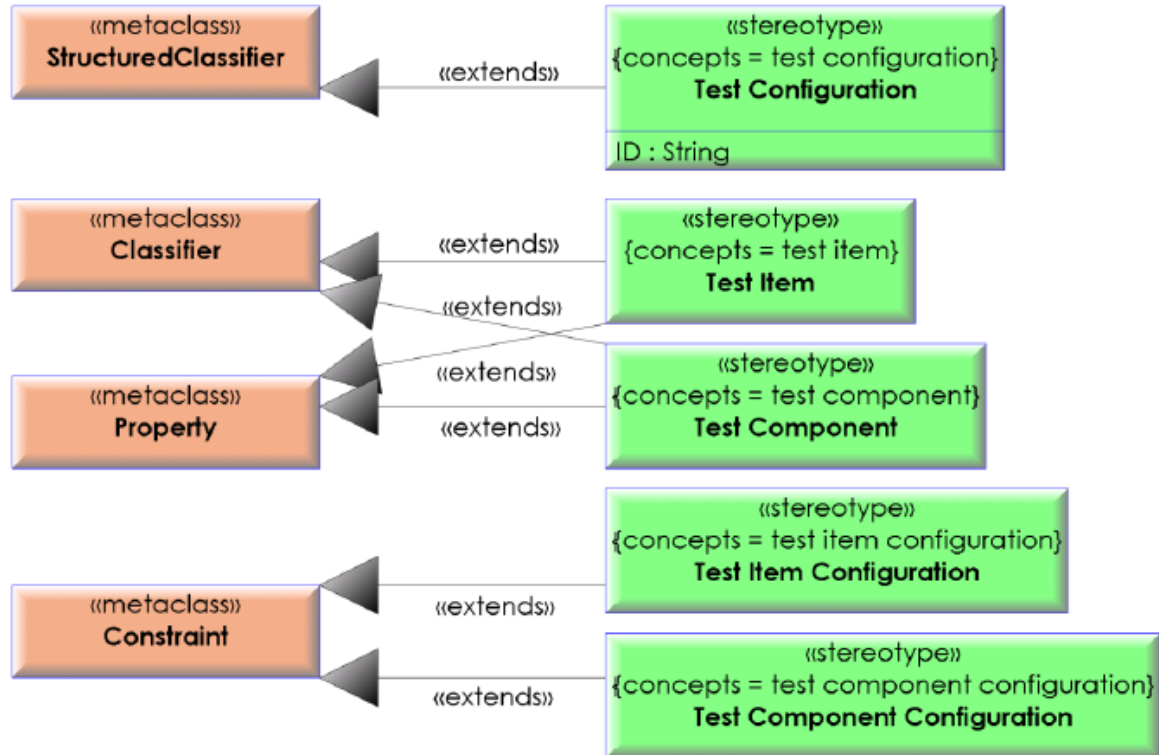
- Introduction
- Advanced standardized test languages
 - TTCN-3
 - **UML testing profile**
 - TDL
- Test Automation Architecture
- Conclusions

UML TESTING PROFILE (UTP)

- Standard by the Object Management Group:
Version 1.0 (2004), current version 1.2 (2013), revised draft version 2.0 (Nov 2015)
- Profile of UML version 2:
Industrial standard for (graphical) modeling of Test architectures, behavior and data.
- Conceptual Model:
Test context, cases, objectives, data, configuration, arbitration&verdicts, logs
- UTP library:
predefined types and values (e.g. ISO 25010 Quality model, ISTQB Test levels)
- Extras:
Mapping to TTCN-3 (**procedure-based communication only**)

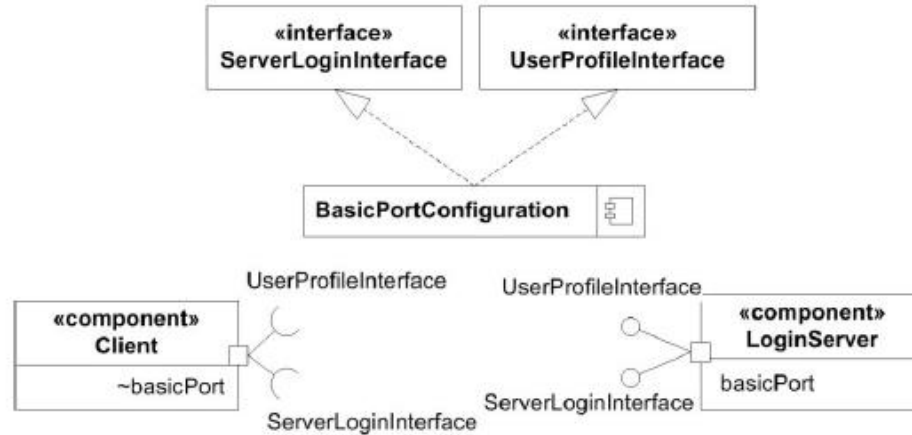
TEST CONFIGURATION OVERVIEW

- Standardized mapping of UTP stereotypes to UML metaclasses

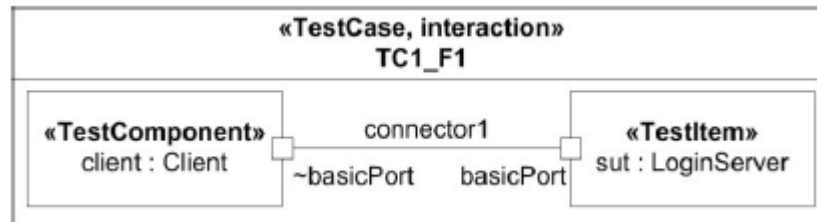


UTP TEST CONFIGURATION EXAMPLE (UTP 2.0 ONLY!)

- LoginServer test components



- LoginServer test configuration



UTP MAJOR USE

- **Domain-independent test modeling for dynamic testing** approaches:
 - Test environments, test configurations, test case specifications (including test case derivation), test data specifications/values
- Test **evaluation**, i.e., managing and visualization **of test results**
- **Integration of best practices**
such as keyword-driven testing, equivalence class testing, etc.
- **Combination with other UML profiles** (e.g., SysML, MARTE, SoaML)
 - E.g. to achieve requirements traceability, ...

LESSONS LEARNED WITH UTP

- **Current version 1.2**
 - less industrial use since version 1.0 ten years ago
 - No big test suites
 - Only some tooling

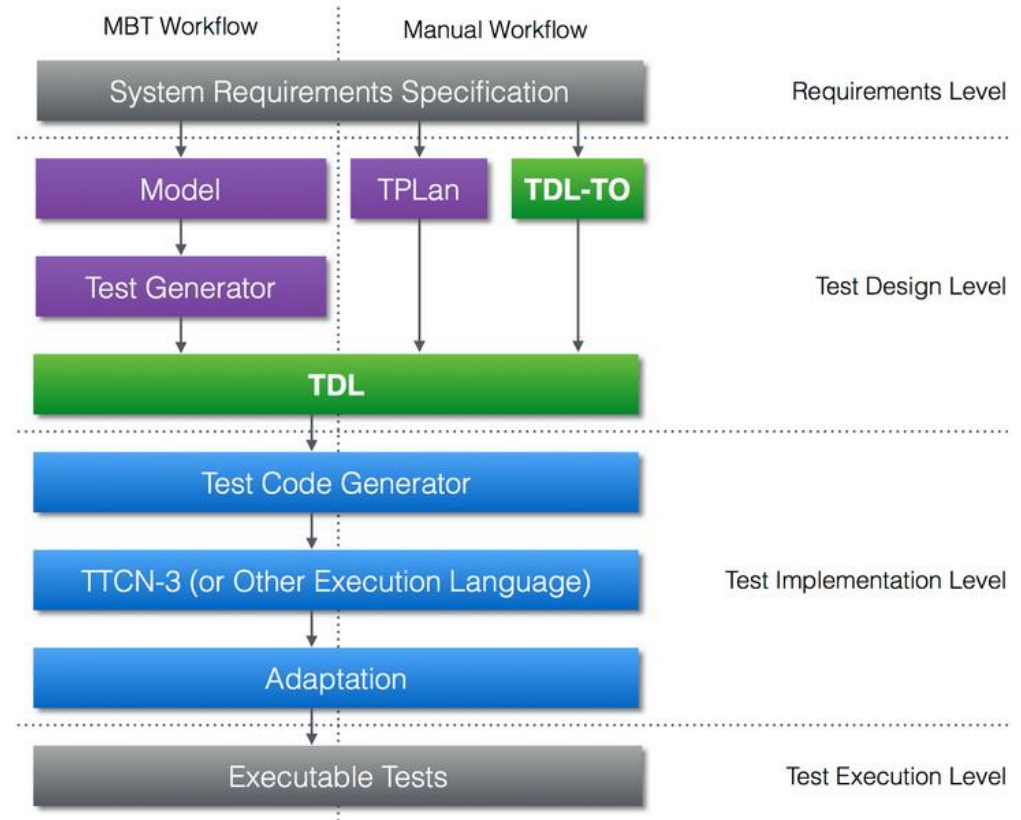
- **UTP Revision 2.0 is promising**
 - E.g. covers test configuration

CONTENTS

- Introduction
- Advanced standardized test languages
 - TTCN-3
 - UML testing profile
 - **TDL**
- Test Automation Architecture
- Conclusions

TEST DESCRIPTION LANGUAGE

- New ETSI Standard
ES 203 119-1 (V1.2.0, 2015-04)
- Fills gap between the high-level test purposes and TTCN-3
- Simple Text notation with graphical presentation



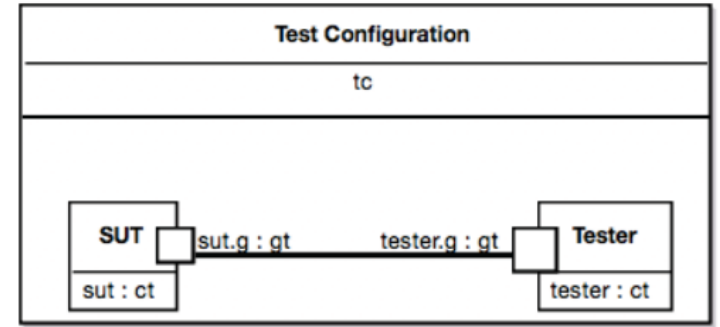
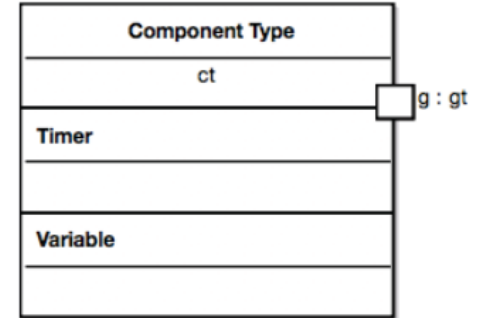
TDL TEST CONFIGURATION

- Typed components and gates
- Timers and variables
- connections among gates
- component roles

```
Gate Type gt accepts Login, Response;
```

```
Component Type ct having {  
  gate g of type gt;  
}
```

```
Test Configuration tc {  
  create Tester tester of type ct;  
  create SUT sut of type ct;  
  connect tester.g to sut.g;  
}
```



LESSONS LEARNED WITH TDL

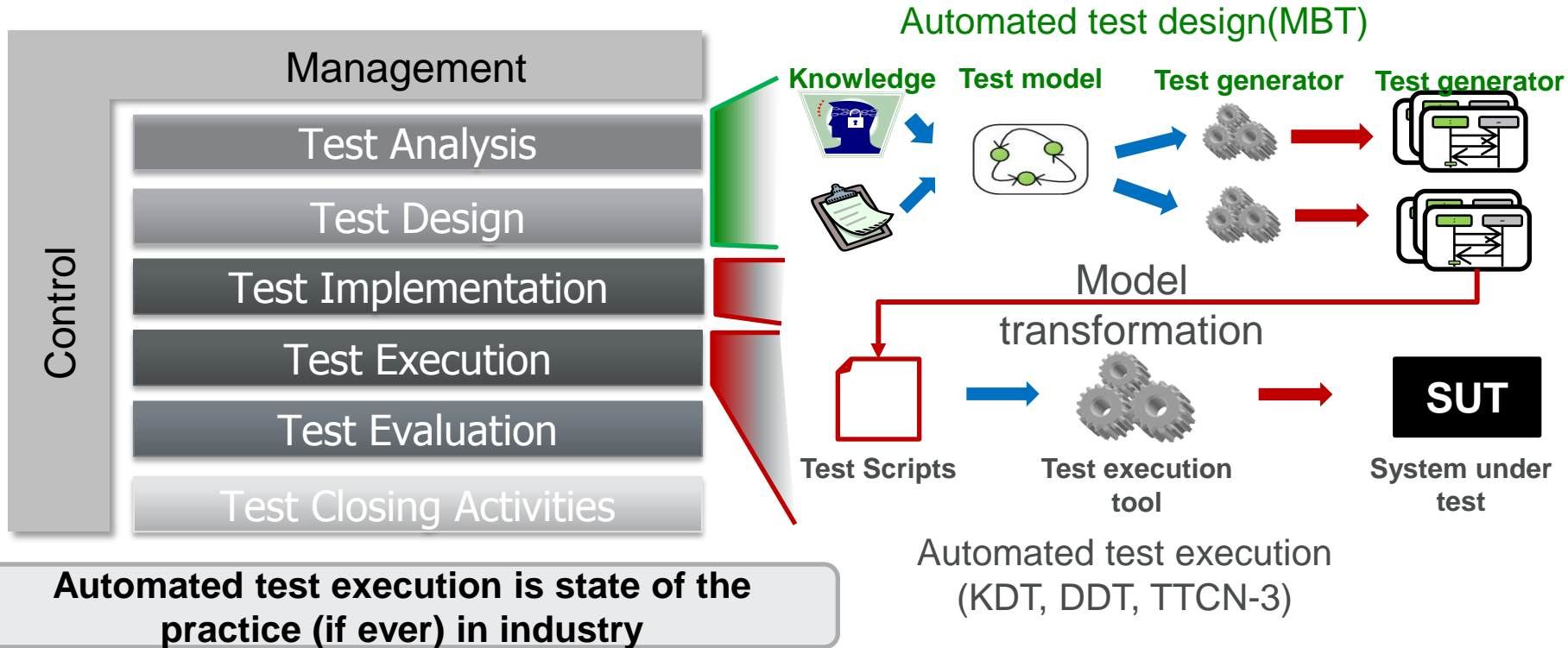
- **Pure testing view**
- **Compromise between UTP and TTCN-3**
 - Simple
 - Executable

- **Not ready to use**

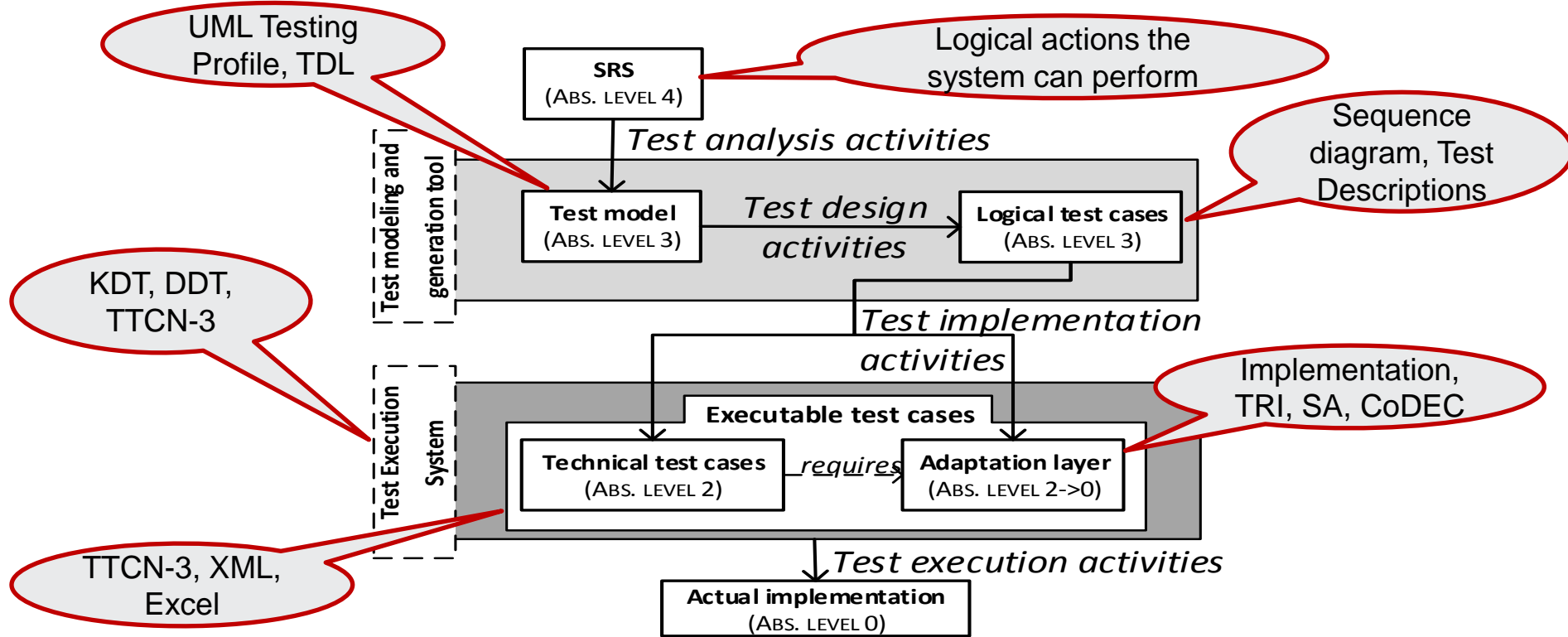
CONTENTS

- **Introduction**
- **Advanced standardized test languages**
 - TTCN-3
 - UML testing profile
 - TDL
- **Test Automation Architecture**
- **Conclusions**

THE ISTQB FUNDEMANENTAL TEST PROCESS



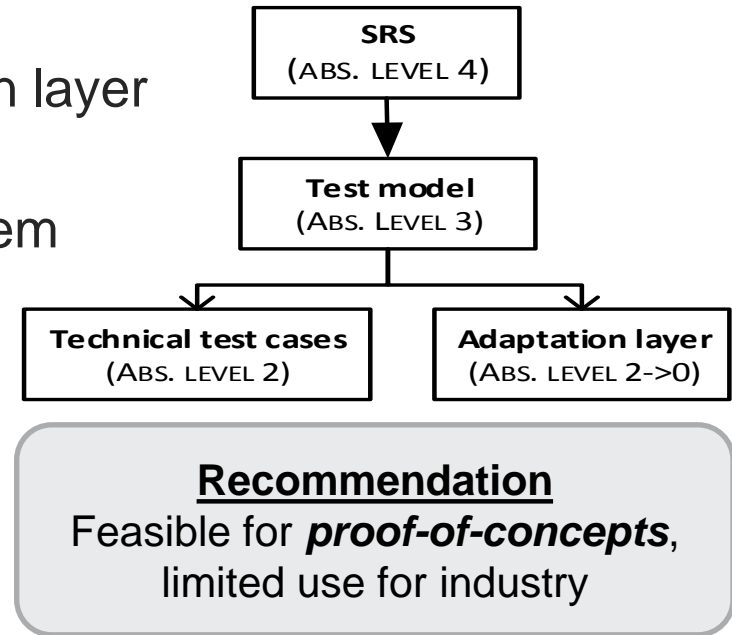
ABSTRACTION LEVELS IN TEST AUTOMATION



TOP-DOWN APPROACH

Model is master

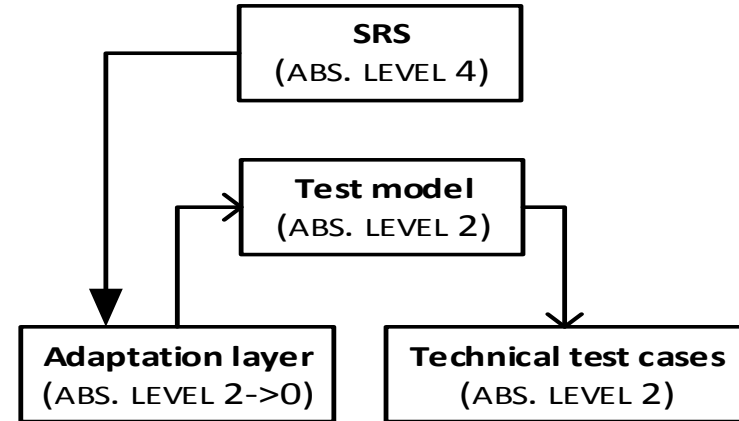
- Test design independent from adaptation layer or test execution system
- No constraint on the test execution system
- Often used in academic prototypes



BOTTOM-UP APPROACH

Adaptation layer is master

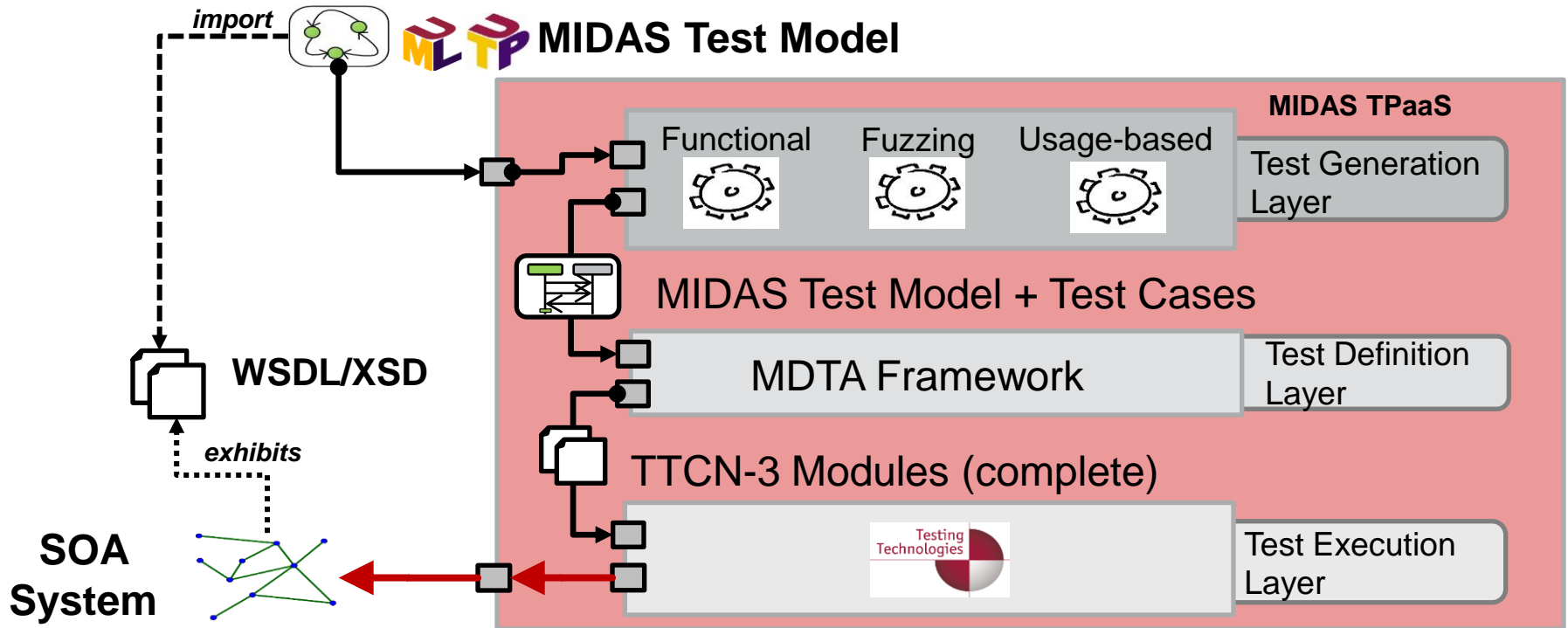
- Ensures immediate automated test execution
- **Requires** available adaption layer
- Test model derived from the adaption layer



Recommendation

Only if adaptation layer is clear for test developers

TEST AUTOMATION ARCHITECTURE: MIDAS



LESSONS LEARNT FROM TEST AUTOMATION ARCHITECTURE

- Two approaches have been shown
- Bottom-up approach was realized in MIDAS
- **Integration of data types** (e.g. WSDL) is challenging
- (Initial) engineering effort can be quite high

CONTENTS

- **Introduction**
- **Advanced standardized test languages**
 - TTCN-3
 - UML testing profile
 - TDL
- **Test Automation Architecture**
- **Conclusions**

SUMMARY: OVERVIEW

	TTCN-3	UTP	TDL
Standardization body	ETSI, ITU-T	OMG	ETSI
History	Since 1992	Since 2004	Since 2014
Applicability	All domains and testing types	All domains	Conformance, interop
Execution tools and solutions	+	<i>proprietary via C-unit</i>	TTCN-3 mapping <i>in preparation</i>
Current user groups	Industry, Research	Academic, Research	n/a

SUMMARY: TEST ARCHITECTURE

	TTCN-3	UTP 2	TDL
Component extension	+	UML generalization	- (<i>reuse elements</i>)
Coordination/ synchronization	+ <i>via libraries</i>	using general ordering	explicitly
Import of WSDL, IDL, etc.	(+)	<i>proprietary via SoaML</i>	-
Graphical Test architecture	- needed!	+	+
Link to UML	-	+	in preparation

GOOD REASONS FOR STANDARDIZED TEST LANGUAGES

- They significantly increases your system quality.
- You can focus on what to test, not on how.
- They reduce costs and efforts in test system maintenance.
- They are independent of access technology, operating system and implementation domain

- They support communication between system development and test department.

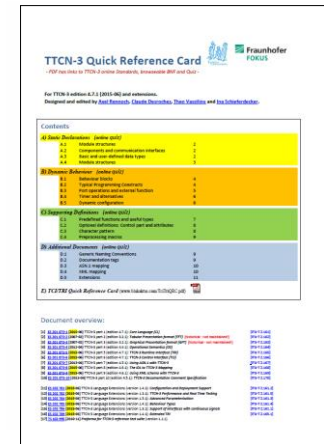
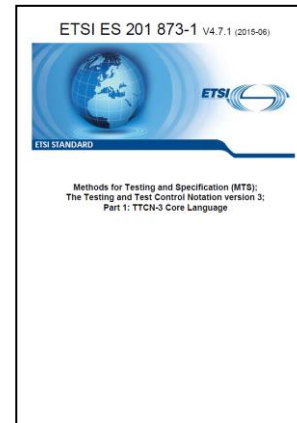
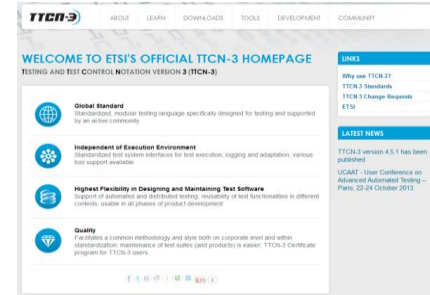
- You can count on available, trained and certified experts

CONCLUSION

- **TTCN-3, UTP and TDL**
 - international Standards for testing
 - allow abstract definitions for testing
 - (partly) accepted in research and industry
- **Tool support** (still weak)
 - UML -> UTP/TDL -> TTCN-3
- Test automation need further enhancements

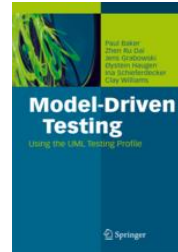
TTCN-3 SOURCES

- Online information
→ www.ttcn-3.org
- TTCN-3 User Conference
→ 2016 in in Budapest, Hungary
- TTCN-3 Standards, Papers, Book
→ <http://www.ttcn.de/>
- Quick Reference
→ <http://www.blukaktus.com/>
- Exercises and Tooling
→ research licenses




UTP SOURCES

- Online information
→ <http://utp.omg.org/>
- MBT User Conference
→ 2016 in Budapest, Hungary
- UTP Standards
→ <http://www.omg.org/spec/UTP/>
- MDT/UTP Book
→ <http://www.springer.com>
- Exercises and Tooling
→ research licenses



UML Testing Profile 2 (UTP2) revised issue only
Date: 10th May 2014



UML Testing Profile 2 (UTP2)

Initial Submission
In response to UML Testing Profile 2 (UTP2) RFP (OMG Document ad/2013-12-08)

OMG Document Number: ad/2014-05-01
Normative reference: ad/2014-05-02
Machine readable tags: <http://www.omg.org/spec/UTP2/40001/utp2.rtf>
Normative: <http://www.omg.org/spec/UTP2/40001/utp2.rtf>
<http://www.omg.org/spec/UTP2/40001/utp2.rtf>
Non-normative: -

The following OMG number specifications submitted this specification (in alphabetic order):
 • Fraunhofer FOKUS, Germany
 • SOFTTEAM, France

The following OMG and external contributor organizations supported this specification (in alphabetic order):
 • Amap, United Kingdom and USA
 • Blekinge University of Applied Sciences, Germany
 • Non-Governmental, Switzerland
 • General Software Testing, USA
 • RELEX SL, Italy
 • Smeets Research Lab, Norway

Mission:
As software becomes increasingly pervasive, the need for quality and reliability in software systems continues to increase. Established methods have been an important development in testing, organizations build software with higher quality. Prior to the development of the UML Testing Profile (UTP), UML, which is the most popular modeling language did not have support for verifying model behavior.

The UTP provides extensions to UML to support the design, verification, identification, analysis, construction, and documentation of artifacts involved in testing. It is independent of implementation languages and technologies, and can be applied in a variety of contexts of development.

In defining the UTP, the group of organizations who participated in its development had a core set of goals:

- **Modularity** - wherever possible, use constructs provided by UML, thereby minimizing the overhead necessary to learn and use the profile.
- **Clarity** - provide a clear separation of testing concepts that helps quality engineers perform their job and address complex concerns as easily as possible.
- **Extensibility** - create a generic profile that can be extended and applied in a variety of domains - (e.g., telecommunications, healthcare, aerospace and biotechnology (e.g. IJSE, NET)).

Contacts:
The Consortium is constituted by leading universities and research institutes in testing methods and technologies and in UML.

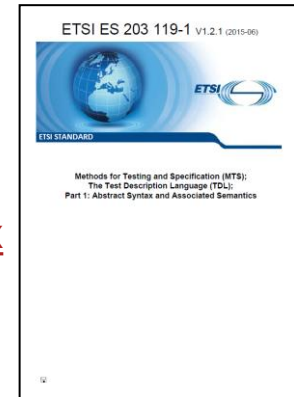
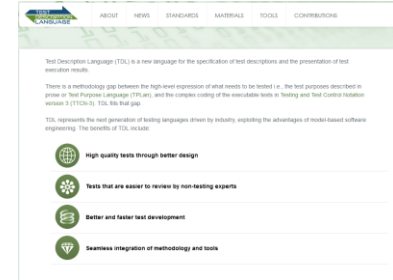
Consortium Partner **Contact Person**
 Fraunhofer FOKUS UML2, UML3, UML4, UML5
 Blekinge UML2, UML3, UML4, UML5
 Non-Governmental UML2, UML3, UML4, UML5
 General Software Testing, USA UML2, UML3, UML4, UML5
 RELEX SL, Italy UML2, UML3, UML4, UML5
 Smeets Research Lab, Norway UML2, UML3, UML4, UML5

The following mailing lists are available for requesting extensions and reporting issues:
 • utp2@lists.sourceforge.net to request general information
 • utp2@lists.sourceforge.net to report issues
 In the case of general questions regarding the UML Testing Profile, please contact:
 • utp2@lists.sourceforge.net
 or any other person listed on the [Consortium page](#)

How To Get Involved:
 If you are interested in getting involved with the group, visit our information or contact list to come as a guest to an upcoming meeting and obtain temporary access to the mailing list. Please contact one of our [document maintainers](#) or contact the Chair.

TDL SOURCES

- Online information
→ <http://tdl.etsi.org/>
- TDL User Conference
→ 2016 in Budapest, Hungary
- TDL Standards
→ <http://tdl.etsi.org/index.php/downloads>
- Exercises and Tooling
(in preparation)
→ <https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx>





**Thank you
for your attention!**

www.fokus.fraunhofer.de
(System Quality Center)

CONTACTS

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

Axel Rennoch
Project Manager
axel.rennoch@fokus.fraunhofer.de
Phone +49 30 3463-7344

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