THE ETSI TEST DESCRIPTION LANGUAGE

Introducing MBT to Standardization

Presented by Andreas Ulrich, Siemens AG for ICTSS 2017, St. Petersburg, Russia

© ETSI 2017. All rights reserved
European Telecommunication Standards Institute

- Develops standards and test specifications for ICT
  - Incl. fixed, mobile, radio, aeronautical, broadcast and internet technologies
  - Ensures and assesses interoperability

ETSI’s activities in testing

- Technical Committee on Methods for Testing and Specification (MTS)
  - Responsible for standardizing test and specification methods and languages, guidelines, frameworks
  - Testing and Test Control Notation version 3 (TTCN-3), Test Description Language (TDL)

- Centre for Testing and Interoperability (CTI)
  - Evaluates test specification technologies
  - Provides hands-on support and assistance to TC and projects

- Interoperability events (Plugtests Service)
TDL – SETTING THE CONTEXT
Challenges in Validating Complex Systems

Characterization
Complex design
→ system of systems
Complex behavior
→ real-time, concurrency
Complex data
→ big data

Adequate validation & testing methods needed
Proper modeling techniques
Proper test automation
Proper fault analysis
Support of agile approaches and scenario-based testing
Always shorter product release cycles lead to a merge of system development and system operation → **DevOps**

**Continuous Feedback on Quality**

**Shift Left**
- Design
- Develop
- Integrate

**Shift Right**
- Test
- Release
- Deploy

**Adaptive Test Architecture**

- **1 yr**
- **1 mo**
- **1 wk**
- **1 dy**
- **1 hr**
- **1 min**
- **1 s**

Release in production frequency

© ETSI 2017. All rights reserved
Cam Kaner on Scenario Testing, STQE Magazine, Sep./Oct. 2003

• The scenario is a story about someone trying to accomplish something with the product under test.

• Scenarios are useful to connect to documented software requirements, especially requirements modeled with use cases.

• A scenario test provides an end-to-end check on a benefit that the program is supposed to deliver.

→ Scenarios are used to systematically test for the correct implementation of requirements in the system.
Can MBT be used for the generation of conformance test suites?

Evaluated tools

- Microsoft Spec Explorer
- Conformiq Designer
- Sepp.med MBTsuite
- Fraunhofer FOKUS MDTester

YES, but...

- Model and test generation tool become parameters of the generated test suite
- Models and generated tests differ between tools and are not comparable → Which model and tool are best?
OMG UML Testing Profile

- Extension of UML to support (model-based) testing
- Wide scope modelling notations inherited from UML
- Might still not capture all needs, further profiles needed, e.g. MARTE
- Semantic fuzz of UML leads to different interpretations
The Beginning of TDL

Siemens AG, Munich, 14 Dec 2011
Business Value

Support to test design before rushing to test code

• Better quality of tests
• Higher productivity in testing

Common test concepts

• Enable development of specialized tools for
  • Test documentation and visualization
  • Generation of executable test code using different strategies
  • Consistency verification and further analysis of test descriptions

• Harmonize test descriptions coming from different tool sources

Support for customized syntax

• Graphical,
• Textual,
• Tabular representation of tests
Ingredients of a Software Language

Abstract Syntax

Concrete Syntax
BNF grammar

Semantics
static and dynamic semantics

Abstract Syntax
meta-model
Standardization in TDL

Abstract Syntax
Standardized, incl. extension capabilities

Concrete Syntax
Standardized graphical and exchange syntax

Semantics
Standardized (formal static, informal dyn. s.)
The TDL Standard Series ETSI ES 203 119

- **TDL-MM**, part 1: Abstract Syntax and Associated Semantics
- **TDL-GR**, part 2: Graphical Syntax
- **TDL-XF**, part 3: Exchange Format
- **TDL-TO**, part 4: Structured Test Objective Language
- **TDL-UML**, part 5: UML Profile for TDL (upcoming)

**Published documents**
**TDL is Adjustable by User**

- Concrete syntax may cover only parts of the meta-model.
- Meta-model can be extended by a user if need arises.
- User extensions of the meta-model can be subjected to further TDL standardization and maintenance.

User-defined TDL concrete syntax

Extends

Partially implements

TDL meta-model with user extensions
Key Concepts of a TDL Specification

**Test configuration**
- Set of interacting components in the roles Tester or SUT

**Test description**
- Represents the expected foreseeable (passing) behavior, i.e. any deviation is a fail
- Expresses a test in terms of interactions of data exchanged between tester and SUT components
- Interactions are totally ordered, i.e. they are implicitly synchronized among components

**Test data**
- Represented as abstract name tuples
TDL Meta-Model Elements

Base elements, Concept of time, Test configuration

Simple data, Structured data, Data use

Test description, Atomic behavior, Combined behavior

Test objective language extension → Part 4
Gate type – restricts communication to data of listed types;
Component types;
Test configuration – wiring of components of role Tester / SUT
Data type definitions;
Data instance definitions (values);
Function declarations
Example – Test Description, GR

Test Description
Configuration

1. interaction
2. interaction
3. interaction

time

test configuration being used
combined behavior
condition
time constraint

OK Resp

© ETSI 2017. All rights reserved
Example – Test Description, TDLan

```java
Test Description ConfigurationTest uses configuration GW_test {
    terminate;
}

Test Description SynchronizationTest uses configuration GW_test {
    execute ConfigurationTest;
    if (client -> moreInvoicesToSend) {
        client.gw_client sends Post_Invoice (invoice = instance returned from getNextInvoice()) to gw.client with {
            time label T_client;
        };
        gw.server sends ? : PostDocumentReq to server.gw_server with {
            time label T_server;
            time constraints : instance returned from lessThan {
                left = time label T_server,
                right = instance returned from plus (left = time label T_client, right = 3)
            };
        };
        server.gw_server sends OK_Resp to gw.server;
    } else {
        set verdict to inconclusive;
        terminate;
    }
}
```

Same test description expressed in textual format
An interaction is atomic and instantaneous

A common concept for three types of communication

- Message-based communication
  - The data represents a message being sent and received via gates
- Procedure-based communication
  - The data represents a (remote) function call being invoked or its return values
- Shared variable access
  - The data represents a shared variable being read or written
Example – TDL Test Objective Language

- Test objective specification (requirement to be tested)
- Support for Behavior-Driven Development
- Prose syntax
- Defined in TDL-TO, part 4

```plaintext
Test Purpose {
    TP Id "TP/CAM/INA/DOP/EV/02"
    Test objective "Checks that CAM message includes DoorOpen information 30s after closed"
    Reference “TS 102 637-2 [1], clauses 7.1 and 7.2"
    PICS Selection PICS_PUETRANSBH
    Initial conditions
    with {
        the IUT entity having reached an initial_state
        and
        the IUT entity having sent a valid CAM message
            containing DoorOpen TaggedValue;
    }
    Expected behaviour
    ensure that {
        when {
            the door entity is closed
        }
        then {
            the IUT entity sends a new CAM message
                containing DoorOpen TaggedValue;
        }
    }
}
```
“Typical” Efforts in Test Automation

- **Specification (20%)**
  - (textual, MSC-like)

- **Scripting (40%)**
  - (TTCN-3, Java)

- **Execution (40%)**

Figures are from an ICT project at Siemens with test team comprising about 24 members.

Execution consists of host tests, tests on target HW and regression tests.

Additional efforts for test env. / test case build and tool support exist.
Abstract vs Concrete Tests

- Test Model
- Test Derivation / Test Specification
- TDL Test Descriptions
- Abstract Test Cases
- Test Code Gen. / Test Scripting
- Concrete Test Cases

The hard part!

© ETSI 2017. All rights reserved
TDL Tool Example –
Generated Textual Editor (EMFText)

Interrupting interactions

Sequence of interactions

Sequence of blocks

SYNTAXDEF tdl
FOR <http://www.etsi.org/space/TDL/20130606>
START TDLSpecification
RULES {
  TDLSpecification ::= "TDLSpecification" "("
           "name" name[","',"]?
           comment* annotation* content* ")";

  TestDescription ::= "TestDescription" "("
                    "name" name[","',"]?
                    comment* annotation*
                    "{"owningPackage" owningPackage[]}?
                    "{"formalParameter" formalParameter[]}*
                    "testConfiguration" testConfiguration[]
                    "{"testObjective" testObjective[]}*
                    behaviour timeConstraint* ")";
Domain specific test description languages can be designed by editing the syntax rules to specific needs.

```
Interaction ::= "Interaction" "{" ("name" name['''',''''])? comment* annotation* ("testObjective" testObjective[])* ("timeConstraint" timeConstraint[])* argument "source" source[] "target" target[]("","" target[])* """";
```

**Interaction** {ATPStatus(516) source ts_gate target op_gate}

```
Interaction ::= "Message: " source[] "->" target[]("","" target[])* argument name['''','''']? testObjective[]* """";
```

**Message**: ts_gate -> op_gate ATPStatus(516) ;
A Scalable TDL-Based Tool Architecture

- Textual Editor (incl. ES 203119-4)
- UML-based Graph. Editor
- TDL Model Analyzer
- TDL Test Generator
- TDL Exchange Format (ES 203119-3)
- Test Plan
- Test Code Generator
- C-code, TTCN-3

Front-end tool
Back-end tool
Artefact (output)

© ETSI 2017. All rights reserved
Lower the barrier of entry for both users and tool vendors in getting started with using TDL

Graphical and textual editors, validation facilities

Visit https://top.etsi.org/ or https://tdl.etsi.org/top
CONCLUSIONS
What Was Achieved?

- A new, standardized approach for the design of test descriptions and test purposes in TDL
  - Graphical, textual and user-defined syntaxes supported
  - Common exchange format

- TDL harmonizes and eases development of tools for scenario-based testing
  - Editors generated from TDL meta-model
  - Re-useable back-end tools, e.g. model analyzers – code & documentation generators
  - Eclipse ecosystem enables quick and low-cost tool development
Timeline

2011
• Very first discussions on TDL

2013
• Work on TDL design and standardization started

2015
• Launch of TDL standards

2017
• Launch of TDL Open Source Project
  • First applications

© ETSI 2017. All rights reserved
## ETSI User Conference on Advanced Automated Testing (UCAAT)

- [https://ucaat.etsi.org/](https://ucaat.etsi.org/)
- TDL tutorials and applications

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Speakers</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:00 - 16:20</td>
<td>Using TDL in the development of standardized test specifications for IoT and other technologies</td>
<td>Michele Carignani and Anthony Wiles (ETSI)</td>
<td>[Abstract &amp; Bio](Abstract &amp; Bio)</td>
</tr>
<tr>
<td>16:20 - 16:40</td>
<td>Mapping TDL to TTCN-3</td>
<td>Philip Makedonski, Gusztav Adamis, Martti Käärik, Finn Kristoffersen and Gyorgy Rethy (ETSI STF)</td>
<td>[Abstract &amp; Bio](Abstract &amp; Bio)</td>
</tr>
<tr>
<td>16:40 - 17:00</td>
<td>Suitability of UTP and TDL for model-based-testing — Checking for compliance with ES 202 951</td>
<td>Marc-Florian Wendland and Ina Schieferdecker (Fraunhofer)</td>
<td>[Abstract &amp; Bio](Abstract &amp; Bio)</td>
</tr>
<tr>
<td>11:40 - 12:00</td>
<td>TDL for testing collaboration IT services: the NetResults experience</td>
<td>Francesco Oppedisano, Sergio Borghese, Francesco Lamonica and Enrico La Vela (NetResults)</td>
<td>[Abstract &amp; Bio](Abstract &amp; Bio)</td>
</tr>
</tbody>
</table>
What Comes Next?

More applications!

Making TDL executable

• Work on (partial) language mapping from TDL to TTCN-3 has started at ETSI MTS
• No general solution, but fault-model dependent
  → Can the different test generation strategies be classified?
• Better understanding of abstraction, underspecification, fault detection, i.e. what features can be omitted from a TDL spec such that the outcome is still predictable?
• Test models in TDL
The author grateful for contributions from
• Philip Makedonski (U Göttingen)
• Martti Käärik (Elvior)

Thanks to the TDL Team @ ETSI MTS
• Michele Carignani, Anthony Wiles (ETSI)
• Gusztav Adamis, György Rethy (Ericsson)
• Finn Kristoffersen (Cinderella ApS)
• Marc-Florian Wendland (FhG FOKUS)
• Xavier Zeitoun (CEA List)
Contact Details:

Andreas Ulrich
Siemens AG, Corporate Technology
Munich, Germany
andreas.ulrich@siemens.com
TDL website, [https://tdl.etsi.org/](https://tdl.etsi.org/)
- Latest standards, documents
- Bug reporting
- Mailing list, [tdl@list.etsi.org](mailto:tdl@list.etsi.org)
- TDL Open Source Project (TOP)

TOP is officially launched at UCAAT 2017
- [https://top.etsi.org/](https://top.etsi.org/)