THE ETSI TEST DESCRIPTION LANGUAGE (TDL)

Results from the ETSI project STF 454

A. Ulrich, G. Adamis, F. Kristoffersen, Ph. Makedonski, M.-F. Wendland, A. Wiles
Outline

- Motivation and introduction
- Design principles
- An example
- Conclusions, next steps
MOTIVATION AND INTRODUCTION
Validating Complex Systems

Engineered systems become more and more complex
- Complex design (system of systems)
- Complex behavior (real-time)
- Complex data (big data)

Validation and testing need to cope with complexity
- Proper modeling techniques
- Proper test automation
- Proper fault analysis techniques
An agile process follows different approaches
- Story/feature driven modeling
- Test driven development, etc.

Leads to scenario-based approach in testing
- Describe a scenario of interacting with the system
- Define test objectives from requirements and connect them to scenarios
- Derive tests from scenarios and automate them
TDL Addresses Needs from Practice

TDL for testing reactive distributed real-time systems

- Provides common black-box testing concepts
- Adjustable to domain-specific needs
- Supporting agile testing process

TDL is standardized

- Clear semantics
- Interoperability of tools and test specifications
- Maintained and kept updated with user needs

TDL use cases

- Manual specification of tests for functional/conformance/interoperability testing
- Representing tests from other sources, e.g. output from MBT test generators
- Documentation of tests
ETSI Standards on TDL Design

ETSI ES 203 119-1 (2014)


Abstract Syntax
(UML based meta-model)

Standardized

Concrete Syntax

Defined by user & standardized
(graphical + exchange)

Semantics
(currently informal)

Standardized
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
Key elements of a TDL specification

Test configuration

• Set of interacting components in the roles Tester or SUT

Test description

• Represents the expected foreseeable (passing) behaviour, 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 Overview

- Structure of TDL specification
- Tester and SUT components, test configuration
- Types and data for interactions
- Concepts for time and time constraints

---

- Structure of test description
- Behavioral elements, e.g. seq, alt, par, loop
- Tracing of test objectives to elements of a test descr.
AN EXAMPLE
Example: Scenario on a Rail Interlocking System (Siemens, MBAT)

Test Configuration

testConfiguration: TrainInterlocking

Test Description

testDescription: StopAndProceed

«testObjective»
reference RQ-1.2.3
description "Verify that the train stops at a signal showing 'stop' and proceeds after signal aspect changes to 'proceed'."

«SUT»
TrainSystem : CompType

Operator : CompType

«Tester»

step
RequestTrainPower(1.0)
RequestSwitchPosition(85, Reverse)
ATPStatus(516, 0)

[interrupt]
ATPStatus(Not 516)

step
RequestSignalAspect(516, Proceed)
ATPStatus(912)

[interrupt]
ATPStatus(Not 912)
Generated Editor for Textual TDL Specifications (EMFText)

Sequence of interactions

Interrupting interactions

Sequence of blocks

(Partial) TDL Concrete Syntax

```
SYNTAXDEF tdl
FOR <http://www.etsi.org/spec/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* 
  "" 
```

Conclusions, Next Steps

TDL meta-model is available currently as an ETSI draft standard

- Further validation of the meta-model necessary
- Final draft for publication planned for January 2014

Next steps

- Design of concrete syntaxes (graphical + exchange format)
- Getting tool support: editors, analyzers, test generators
- Further refinement of the TDL meta-model
- Extend TDL to support test automation
  - Extensions to ensure executability
  - Composition of test descriptions → User story models
Team

Andreas Ulrich, Siemens AG, andreas.ulrich@siemens.com (STF lead)
Gusztáv Adamis, Ericsson, Gusztav.Adamis@ericsson.com
Finn Kristoffersen, Cinderella, finn@cinderella.dk
Philip Makedonski, U Göttingen, makedonski@informatik.uni-goettingen.de
Marc-Florian Wendland, FhG FOKUS, marc-florian.wendland@fokus.fraunhofer.de
Anthony Wiles, ETSI CTI, Anthony.Wiles@ETSI.ORG

Acknowledgement

A. Ulrich, Siemens AG acknowledges partial funding of this activity from the ARTEMIS Joint Undertaking, grant agreement no. 269335 (MBAT) and the German BMBF.