APPLYING TDL IN PRACTICE

Philip Makedonski, Gusztav Adamis, Martti Käärik, Finn Kristoffersen, Xavier Zeitoun
Overview

What is TDL?

- Test Description Language
  - Design, documentation, and representation of formal test descriptions
- Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)

UP4TDL, TDL or UML?

Where does TDL fit in?

Keyword-Driven Testing
  - MBT
  - Representation
    - Generation
      - Standards
    - Rail
      - Visualisation
    - Documentation
      - ITS
      - Interoperability
      - Conformance
What is TDL?

- Test Description Language
  - Design, documentation, and representation of formal test descriptions
- Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)
What is TDL?

- Design, documentation, representation?
- ease development and review
- improve productivity and quality
- both industry and standardisation
- reduce implementation details
What is TDL?

- Scenario-based?
  - describe interactions with a system
  - attach test objectives to scenarios
  - derive and automate tests
- Reactive, distributed, real-time
  - common black-box testing concepts
  - domain adaptation, agile development
What is TDL?

• Standardised?
  • clear semantics
  • interoperability of tools and test specifications
  • updated with user needs
  • maintenance commitment
What is TDL?

- **Contributions from:**
  - Siemens AG, Ericsson Hungary
  - Fraunhofer FOKUS, ETSI CTI
  - CEA, University of Göttingen
  - OU Elvior, Cinderella ApS

- **Guidance:**
  - Steering Group, TC MTS
What is TDL?

Part 1: MM Meta-Model and Semantics

Part 2: GR Graphical Syntax

Part 3: XF Exchange Format

Part 4: TO Structured Test Objective Specification
What is TDL?

TDL P1: MM

TDL P2: GR

TDL P3: XF

TDL P4: TO

User-defined syntax
What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time
What is TDL? Part 1: MM

- TDL main ingredients
- Test data
- Test configuration
- Test behaviour
- Test objectives
- Time

Annex B (informative):
Examples of a TDL Concrete Syntax

B.1 Introduction

The applicability of the TDL meta-model that is described in the main part of the present document depends on the availability of TDL concrete syntaxes that implement the meta-model (abstract syntax). Such a TDL concrete syntax can then be used by and users to write TDL specifications. Though a concrete syntax will be based on the TDL meta-model, it can implement only parts of the meta-model if certain TDL features are not necessary to handle a user’s needs.

This annex illustrates an example of a possible TDL concrete syntax in a textual format that supports all features of the TDL meta-model, called “TDLan”. Three examples are outlined below - two examples translated from source annotations and descriptions taken from [3] and [2], as well as an example illustrating some of the TDL document mapping concepts. The examples are accompanied by a complete reference description of the textual syntax of TDLan in EBSD.

B.2 A 3GPP Conformance Example in Textual Syntax

This example describes one possible way to translate clause 7.3.1 from TS 141 323-1 [2] into the proposed formal syntax, by mapping the concepts from the representation of the source document in the textual format in the TDL meta-model by means of the proposed formal syntax. The example has been enriched with some further information, such as explicit data definitions and test configuration details for completeness when applying.

The Test Description Language (TDL);
What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time
What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time
What is TDL? Part 1: MM

- Test data
  - data definition and data use
  - abstract types and instances
  - composed by using parameters
  - functions and actions
  - mappable to concrete data
  - variables and special values
What is TDL? Part 1: MM

Type Login;
Login correct;
Login incorrect;

Use "data.ttcn3" as DATA ;
Map correct to "johnny_correct" in DATA as correct_ttcn3;
Map incorrect to "johnny_incorrect" in DATA as incorrect_ttcn3;

template Login johnny_correct := {
    user := "johnny",
    password := "apple",
    hint := "seed",
    id := 1000
}
template Login johnny_incorrect := {
    user := "johnny",
    password := "orange",
    hint := "second favourite fruit",
    id := 2000
}
type record Login {
    charstring user,
    charstring password,
    charstring hint,
    integer id
} with {
    encode "xpath=./div[@id='login']";
    encode (user) "relative=./div/dd[3]";
    encode (password) "relative=./div/dd[4]";
};
What is TDL? Part 1: MM

Type Login;
Login correct;
Login incorrect;

Use "data.ttcn3" as DATA ;
Map correct to "johnny_correct" in DATA as correct_ttcn3;
Map incorrect to "johnny_incorrect" in DATA as incorrect_ttcn3;
What is TDL?

Part 1: MM

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

Type Login

Login correct
Login incorrect

Use "data.ttcn3" as DATA;

Map correct to "johnny_correct" in DATA as correct_ttcn3;

Map incorrect to "johnny_incorrect" in DATA as incorrect_ttcn3;
What is TDL? Part 1: MM

- Test configuration
  - typed components and gates
  - timers and variables
  - connections among gates
  - component roles
What is TDL? Part 1: MM

Gate Type \texttt{gt} accepts \texttt{Login}, \texttt{Response};

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

Test Configuration \texttt{tc} {
  create \texttt{Tester} \texttt{tester} of type \texttt{ct};
  create \texttt{SUT} \texttt{sut} of type \texttt{ct};
  connect \texttt{tester.g} to \texttt{sut.g};
}
What is TDL? Part 1: MM

- Test behaviour
  - defines expected behaviour
  - failure upon deviations by default
  - actions and interactions
  - alternative, parallel, iterative, conditional
  - defaulting, interrupting, breaking
What is TDL? Part 1: MM

Test Description td (p of type Login)
uses configuration tc {
  tester.g sends incorrect to sut.g;
  alternatively {
    sut.g sends failure to tester.g with {
      test objectives : tp;
    };
    set verdict to pass;
  } or {
    sut.g sends success to tester.g;
    set verdict to fail;
  }
}

or simply (relying on the default semantics):

Test Description td_default (p of type Login)
uses configuration tc {
  tester.g sends incorrect to sut.g;
  sut.g sends failure to tester.g with {
    test objectives : tp;
  };
}
What is TDL? Part 1: MM

- Test objectives
  - may be attached to
    - behaviour (atomic or compound)
    - whole test description
  - contain description and reference
What is TDL? Part 1: MM

Test Objective tp {
    description : "ensure that
    when incorrect login is provided
    a failure response is sent";
}

Test Description td (p of type Login) 
uses configuration tc {
    tester.g sends incorrect to sut.g;
    alternatively {
        sut.g sends failure to tester.g with {
            test objectives : tp;
        };
        set verdict to pass;
    } or {
        sut.g sends success to tester.g;
        set verdict to fail;
    }
}
What is TDL? Part 2: GR

- Graphical languages
  - common in (test) modelling
  - ease communication
- TDL Graphical Syntax
  - hybrid graphical language
  - simple shapes, compartments
  - textual visualisation of contents
What is TDL? Part 2: GR

- Aligned with UML
  - distinct where semantics differ
- One diagram to rule them all!
- BNF-like label specification
- Considers both ease of use and implementation
- Prototyped with Sirius
What is TDL? Part 2: GR

**ComponentTypeLabel**

**Timer: TimerListLabel**

**GateLabel**

context: ComponentType

**ComponentTypeLabel** ::= self.name

**TimerListLabel** ::= self.timer.name

…

Component Type

- ct
- Timer
- Variable
What is TDL? Part 3: XF

- Based on OMG XMI
  - XML: Metadata Interchange
  - Serialisation of MOF models
  - Exchange among MOF tools
- XMI concerns
  - complex, many options
What is TDL? Part 3: XF

- TDL specific XMI structure
  - exchange of TDL models
  - canonical TDL XMI structure
    - meta-class representations
    - multiplicity, associations, inheritance
  - restrict flexibility of XMI
  - syntactical validity only!
What is TDL? Part 3: XF

• Syntactical validity only?
  • two-step validation
  • syntax: XMI Schema
  • semantics: MOF model validation
What is TDL? Part 3: XF

```xml
<xsd:complexType name="ComponentInstance">
  <xsd:complexContent>
    <xsd:extension base="tdl:Element">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element name="gateInstance" type="tdl:GateInstance"/>
        <xsd:element name="variable" type="tdl:Variable"/>
      </xsd:choice>
      <xsd:attribute name="componentType" type="xsd:anyURI"/>
      <xsd:attribute name="role" type="tdl:ComponentInstanceRole"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```
What is TDL? Part 3: XF

```xml
<packagedElement xsi:type="tdl:ComponentType" xmi:id="_qKtZ33asEeWrfP0MdfQNpg" name="ct">
  <gateInstance xmi:id="_qKtZ4HasEeWrfP0MdfQNpg" name="g" type="_qKtZ3nasEeWrfP0MdfQNpg"/>
</packagedElement>
```
What is TDL? Part 4: TO

- Based on TPLan
  - refine test objectives
  - formalise specification
  - integrate and unify test description and test purpose specification
What is TDL? Part 4: TO

- Base Standard Specification
- Identification of Requirements
- Creation of ICS/IFS
- Definition of TSS
- Specification of Test Purposes
- Specification of Test Descriptions
- Specification of Test Cases
- Validation
What is TDL? Part 4: TO

- Base Standard Specification
- Identification of Requirements
- Creation of ICS/IFS
- Definition of TSS
- Specification of Test Purposes
- Specification of Test Descriptions
- Specification of Test Cases
- Validation
## What is TDL? Part 4: TO

<table>
<thead>
<tr>
<th>Base Standard Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of Requirements</td>
</tr>
<tr>
<td>Creation of ICS/IFS</td>
</tr>
<tr>
<td>Definition of TSS</td>
</tr>
<tr>
<td>Specification of Test Purposes</td>
</tr>
<tr>
<td>Specification of Test Descriptions</td>
</tr>
<tr>
<td>Specification of Test Cases</td>
</tr>
<tr>
<td>Validation</td>
</tr>
</tbody>
</table>
What is TDL? Part 4: TO

Test Purpose {
   TP Id "TP/CAM/INA/DOP/BV/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_PUBTRANSVEH
   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;
      }
   }
What is TDL?

- Ultimately standards need...
  - maturity
  - transparent change management
  - proof by implementation
  - validation by tests
### ETSI's Bug Tracker

Logged in as: makedonski (Philip Makedonski - manager)  13-09-2014 22:00 IST

**Viewing Issues (1 - 10 / 10)**

<table>
<thead>
<tr>
<th>P</th>
<th>ID</th>
<th>Project</th>
<th>Severity</th>
<th>Status</th>
<th>Updated</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0006768</td>
<td>TDL meta-model</td>
<td>major</td>
<td>assigned (Andreas Ulrich)</td>
<td>01-08-2014</td>
<td>New MM element as the starting point of the Behaviour Description of a Test Description</td>
</tr>
<tr>
<td></td>
<td>0006773</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Accessing DataProxy arguments</td>
</tr>
<tr>
<td></td>
<td>0006765</td>
<td>TDL meta-model</td>
<td>major</td>
<td>assigned (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Time Observation</td>
</tr>
<tr>
<td></td>
<td>0006764</td>
<td>TDL meta-model</td>
<td>minor</td>
<td>assigned (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Description of VerdictType shall be modified</td>
</tr>
<tr>
<td></td>
<td>0006763</td>
<td>TDL meta-model</td>
<td>minor</td>
<td>resolved (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Blocks of ParallelBehaviour should be able to declare Guards</td>
</tr>
<tr>
<td></td>
<td>0006767</td>
<td>TDL</td>
<td>minor</td>
<td>resolved (Andreas Ulrich)</td>
<td>11-07-2014</td>
<td>Allow to reference test descriptions that run on a different test (sub-) configuration</td>
</tr>
<tr>
<td></td>
<td>0006772</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Variable assignment from Interaction and ActionReference</td>
</tr>
<tr>
<td></td>
<td>0006771</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Component variables</td>
</tr>
<tr>
<td></td>
<td>0006770</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Named parameters</td>
</tr>
<tr>
<td></td>
<td>0006769</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Move parameters from DataInstance to DataSet</td>
</tr>
</tbody>
</table>

**New** | **Feedback** | **Acknowledged** | **Confirmed** | **Assigned** | **Resolved** | **Closed**

Proof by Implementation

TDL Website at tdl.etsi.org
What is TDL?

- Test Description Language
- Design, documentation, and representation of formal test descriptions
- Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)
What is TDL?

- Test Description Language
  - Design, documentation, and representation of formal test descriptions
- Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)
What is TDL?
UML Profile 4 TDL, but why?

“to enable its interoperability with and application in UML-based working environments and model-based testing approaches”

Examples of UML Base approaches:

- Use UML Based Tool such Papyrus to Edit «UP4TDL» models, Moka to animate a TDL TestDescription
- Use TDL on SysML model
- Use PhiSystem profile for CPS
- Use UTP to model tests
UML Profile 4 TDL, but why?

- Define a domain specific terminology, i.e. a domain specific notation instead of the plain UML2 notation.

- Complete/specialize the UML2 semantics for dealing with:
  - UML Semantics Variation Points,
  - For clarifying ambiguous definition,
  - For specializing an existing semantics aspect of UML2.

- Define usage constraints of the UML2 in order to drive/limit its usage e.g. for defining a domain specific methodology limiting the scope of UML.

- Define new meta-information for annotating a model for a given purpose e.g. for code generation purpose, for enabling model-based analysis such as quality performance analysis, etc.
UP4TDL, TDL or UML?

- MOF
- UML extends
- TDL
- TDL Models
- M2M
- UP4TDL
- UP4TDL Models
- XMI Import/Export
Extension Block: The stereotype

- **Extension**
  - Associations, Specializations, constraining, add properties (also called TaggedValues)

- **Example of property**
  - TestDescription holds the TestConfiguration it refers to.

- **Example of constraint**
  - Block: all MessageOccurrence related to the same message shall be consecutive
  - TestDescription shall have a TestConfiguration
  - ComponentInstanceBinding: same/compatible ComponentType
Foundation Package

Profile Definition

Concept Overview:
- Most concepts already in UML
- Main additions: Annotation, TestObjective

Profile Application

Functionality overview:
- Use UML “Packaging”
- Add Annotation & TestObjectives
Data Definition Package

Profile Definition

Concept Overview:

- Rely mostly on concepts present in the UML Class Diagram
- Main additions: DataMapping Concepts & Function

Profile Application

Functionality overview:

- Declare Types, Instance, Actions and Functions
- Map those Elements to a Resource
DataUse Package

Profile Definition

Concept Overview:
- Mostly extend UML::Expression
- aim: build a Data Use grammar

Profile Application

Functionality overview:
- Call functions, use variable, use Instance in:
  - Guards
  - Arguments of AtomicBehaviours (e.g. Interaction)
TestConfiguration Package

Profile Definition

Concept Overview:
- Rely on concepts involved in CompositeStructure Diagram
- Allows combination with SysML

Profile Application

Functionality overview:
- Edit TestConfiguration
  - Add ComponentInstance
  - Declare ComponentType
  - Add Gates & Connections
**CombinedBehaviour Package**

**Profile Definition**

**Concept Overview:**
- Stereotype extending CombinedFragment
- A Block Stereotype that extend InteractionOperand

**Profile Application**

**Functionality overview:**
- Create CombinedBehaviour & add blocks to these behaviours
Atomic Behaviour Package: Actions

Profile Definition
Concept Overview:

Profile Application
Functionality overview:
- Add/Edit Actions, Verdict Assignment, TestDescriptionReference to a TestDescription

Diagram: UML specification of the Atomic Behaviour Package with stereotypes for action references, inline actions, assignments, and test description references.
AtomicBehaviour Package : Interaction

Profile Definition

Concept Overview:
- Extension of OccurrenceSpecification

Profile Application

Functionality overview:
- Add Interaction, edit its argument
Editing UP4TDL Models

Papyrus-Based Editor:

- Planned:
  - Automatic model validation
  - Xtext editor For DataUse
  - Extended Wizards for TDL projects & TDL diagrams
  - Implementation of the TDL graphical syntax
- OpenSource Eclipse Editor for UML Extension of 3 UML Diagrams:
  - Composite Structure Diagram for TestConfiguration
  - Sequence Diagram for TestDescription
  - Class Diagram for Data Declaration
What is TDL?
Where does TDL fit in?

Keyword-Driven Testing

MBT

Generation

Standards

Documentation

Interoperability

Representation

Rail

Visualisation

ITS

Conformance
Where does TDL fit in?
Where does TDL fit in?
• Context
  • Conformance and interoperability test descriptions
  • Standardised test specifications for various ETSI technologies
  • Typically protocol oriented, used in certification schemes
  • End-to-end interoperability of systems involving different equipment
• Stakeholders
  • High-level discussions at large meetings (80-100 participants)
    • ETSI Technical Committees, 3GPP, other standards organisations, CTI Plugtests team and participants, industrial fora and equipment vendors
    • all need to be familiar with and fluent in the syntax being used.
    • different notions of “good” test
  • Better comprehension among developers with little or no testing expertise
    • bridge the gap between management, core specifications experts, testing experts
• Challenges

• Informal (Word, Excel) or semi-formal (TPLan) approaches
  • considered inadequate, no test descriptions as a consequence
  • no single consistent approach, varying level of quality, detail, difficult maintenance
  • certification requires completeness and accuracy, test descriptions are the design stage before developing TTCN-3 test cases

• Acceptance for more rigorous approaches among Technical Committees
  • applicable to a wide range of technologies (protocols, services, applications)
• TDL
• Standardised approach improves consistency
• Tools offer faster development, higher quality, easier maintenance
• Direct link to TTCN-3
• Initial run within ITS, expand to other Technical Committees
#### 7.2.2.3.1 Test Purpose (TP)

(1)

**with**: { UE in E-UTRA RRC_CONNECTED state }

**ensure that**: {
  **when**: { UE receives a 5 bit SN configured RLC PDU with Length Indicator value larger than RLC PDU size }
  **then**: { UE discards the RLC PDU }
}

#### 7.2.2.3.3.2 Test procedure sequence

**Table 7.2.2.3.3.2-1: Main behaviour**

<table>
<thead>
<tr>
<th>St</th>
<th>Procedure</th>
<th>Message Sequence</th>
<th>TP</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXCEPTION: the behaviour described in table 7.2.2.3.2-2 runs in parallel with steps 1 to 5 below.</td>
<td>- -</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>The SS transmits UMD PDU#1 containing first segment of RLC SDU#1.</td>
<td>&lt;-- UMD PDU#1 (SN=0)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>The SS transmits UMD PDU#2 containing last segment of RLC SDU#1 and first segment of RLC SDU#2.</td>
<td>&lt;-- UMD PDU#2 (SN=1)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>The SS transmits UMD PDU#3 containing last segment of RLC SDU#2, first segment of RLC SDU#3 and with Length Indicator that points beyond the end of the UMD PDU#3.</td>
<td>&lt;-- UMD PDU#3 (SN=2)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>The SS transmits UMD PDU#4 containing last segment of RLC SDU#3.</td>
<td>&lt;-- UMD PDU#4 (SN=3)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>The SS transmits UMD PDU#5 containing RLC SDU#4.</td>
<td>&lt;-- UMD PDU#5 (SN=4)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
7.2.2.3 UM RLC / Reassembly / 5-bit SN / LI value > PDU size

7.2.2.3.1 Test Purpose (TP)

(1)

with \{ UE in E-UTRA RRC_CONNECTED state \}

ensure that

\begin{align*}
&\text{when} \quad \{ \text{UE receives a 5 bit SN configured RLC PDU with Length Indicator value larger than RLC PDU size} \} \\
&\text{then} \quad \{ \text{UE discards the RLC PDU} \}
\end{align*}

7.2.2.3.2 Test procedure sequence

**Table 7.2.2.3.3.2-1: Main behaviour**

<table>
<thead>
<tr>
<th>St</th>
<th>Procedure</th>
<th>Message Sequence</th>
<th>TP</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.2.3.3.2-1</td>
<td>EXCEPTION: the behaviour described in table 7.2.2.3.3.2-2 runs in parallel with steps 1 to 5 below.</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>The SS transmits UMD PDU#1 containing first segment of RLC SDU#1.</td>
<td>&lt;-</td>
<td>UMD PDU#1 (SN=0)</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>The SS transmits UMD PDU#2 containing last segment of RLC SDU#1 and first segment of RLC SDU#2.</td>
<td>&lt;-</td>
<td>UMD PDU#2 (SN=1)</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>The SS transmits UMD PDU#3 containing last segment of RLC SDU#2, first segment of RLC SDU#3 and with Length Indicator that points beyond the end of the UMD PDU#3.</td>
<td>&lt;-</td>
<td>UMD PDU#3 (SN=2)</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>The SS transmits UMD PDU#4 containing last segment of RLC SDU#3.</td>
<td>&lt;-</td>
<td>UMD PDU#4 (SN=3)</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>The SS transmits UMD PDU#5 containing RLC SDU#4.</td>
<td>&lt;-</td>
<td>UMD PDU#5 (SN=4)</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 7.2.2.3.3.2-2: Parallel behaviour**

<table>
<thead>
<tr>
<th>St</th>
<th>Procedure</th>
<th>Message Sequence</th>
<th>TP</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The UE transmits RLC SDU#1.</td>
<td>--&gt;</td>
<td>(RLC SDU#1)</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Check: Does the UE transmit RLC SDU#2?</td>
<td>--&gt;</td>
<td>(RLC SDU#2)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Check: Does the UE transmit RLC SDU#3?</td>
<td>--&gt;</td>
<td>(RLC SDU#3)</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>The UE transmits RLC SDU#4.</td>
<td>--&gt;</td>
<td>(RLC SDU#4)</td>
<td>-</td>
</tr>
</tbody>
</table>
### 4.5.1 General Capabilities

#### 4.5.1.1 SIP messages longer than 1 500 bytes

<table>
<thead>
<tr>
<th>Interoperability Test Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier:</td>
</tr>
<tr>
<td>Summary:</td>
</tr>
<tr>
<td>Configuration:</td>
</tr>
<tr>
<td>SUT</td>
</tr>
<tr>
<td>References</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Use Case ref.:</td>
</tr>
</tbody>
</table>

**Pre-test conditions:**
- HSS of IMS_A and of IMS_B is configured according to table 1
- UE_A and UE_B have IP bearers established to their respective IMS networks as per clause 4.2.1
- UE_A and IMS_A configured to use TCP for transport
- UE_A is registered in IMS_A using any user identity
- UE_B is registered user of IMS_B using any user identity
- MESSAGE request and response has to be supported at II-NNI (TS 129 165 [16] see tables 6.1 and 6.3)

<table>
<thead>
<tr>
<th>Test Sequence:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conformance Criteria:</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>TP.IMS_4002_01 in CFW step 4 (MESSAGE) ensure that {</td>
</tr>
<tr>
<td></td>
<td>when { UE_A sends a MESSAGE to UE_B</td>
</tr>
<tr>
<td></td>
<td>containing a Message_Body greater than 1 300 bytes }</td>
</tr>
<tr>
<td></td>
<td>then { IMS_B receives the MESSAGE</td>
</tr>
<tr>
<td></td>
<td>containing the Message_Body greater than 1 300 bytes }</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Direction</th>
<th>Message</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User A</td>
<td>User B</td>
<td>User A sends an instant message to user B</td>
</tr>
<tr>
<td>2</td>
<td>User B</td>
<td>User A</td>
<td>MESSAGE</td>
</tr>
<tr>
<td>3</td>
<td>IMS_A</td>
<td>User B</td>
<td>MESSAGE</td>
</tr>
</tbody>
</table>
4.5.1 General Capabilities

4.5.1.1 SIP messages longer than 1 500 bytes

<table>
<thead>
<tr>
<th>Identifier:</th>
<th>TD_IMS_MESS_0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>IMS network shall support SIP messages greater than 1 500 bytes</td>
</tr>
<tr>
<td>Configuration:</td>
<td>CF_INT_CALL</td>
</tr>
<tr>
<td>SUT</td>
<td>IMS_B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>References</th>
<th>Test Purpose</th>
<th>Specification Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP_IMS_4002_1</td>
<td></td>
<td>TS 124 229 [1], clause 4.2A ¶1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>User A</th>
<th>User B</th>
<th>IMS A</th>
<th>IMS B</th>
<th>IBCF A</th>
<th>IBCF B</th>
<th>IMS B</th>
<th>User A</th>
<th>User B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>User A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>User A</td>
<td>IMS A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>User A</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS A</td>
<td>IMS B</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>8</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>9</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>10</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>11</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>12</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
<tr>
<td>13</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
<td>IMS B</td>
</tr>
</tbody>
</table>

User A sends an instant message to user B

User B is informed about the instant message

Optional: User A is presented a delivery report
From ETSI TS 102 868-2 V1.1.1 (2011-03):

<table>
<thead>
<tr>
<th>TP Id</th>
<th>TP/CAM/INA/DOP/BV/02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test objective</td>
<td>Checks that CAM message includes DoorOpen information 30s after closed</td>
</tr>
<tr>
<td>Reference</td>
<td>TS 102 637-2 [1], clauses 7.1 and 7.2</td>
</tr>
<tr>
<td>PICS Selection</td>
<td>PICS_PUBTRANSVEH</td>
</tr>
</tbody>
</table>

**Initial conditions**

with {
  the IUT being in the "initial state" and
  the IUT having sent a valid CAM message
  containing DoorOpen TaggedValue
}

**Expected behaviour**

ensure that {
  when {
    the door is closed
  }
  then {
    the IUT sends CAM messages
    containing DoorOpen TaggedValue during the 30s following the door closing event
  }
}
• Context
  • TDL in MBT: Keyword driven UI testing
  • Create behavioural model of the SUT using symbolic action descriptions
    • define keywords once
    • map abstract keyword definitions to keyword implementations in execution language
  • Generate abstract test sequences by means of MBT
  • Convert abstract test sequences to a test execution language
• Challenges
  • Generated test sequences
    • proprietary format - not accessible, tool-specific integrations to requirements management, test planning
    • straight to executable code - loss of meta-data, difficult parameterisation
  • Mapping between abstract (symbolic) and real test system interface
    • implicit - error-prone
    • implemented in test execution language - additional overhead, language limitations
• TDL
  • Interoperability with requirements management by explicit test objectives
  • Parameterisation of test descriptions and symbolic data representations
  • Explicit data mapping to underlying data system of execution language
• Advantages over alternatives
  • Less ambiguity, testing specific (e.g. break, stop, default concepts)
• Context
  • Test automation tools for performance and load tests

• Challenges
  • Textual test specifications with sequence diagram-like examples (or using a different graphical notation)
  • Manual derivation of TTCN-3 code and configuration settings
  • Too wide a gap between input and output!
• **TDL**
  - Raises the abstraction level of the test description
    - multiple levels of test specification (from system to implementation), iterative and agile development
  - Concentrate on the problems themselves rather than programming details

• **Application**
  - Visualisation of test case behaviour
  - Automatic generation of TTCN-3 code from TDL test descriptions
Where does TDL fit in?

- Requirements
  - User stories
- System models
- Manual test derivation in Syntax #1
- Automatic test generation
- Executable tests
  - NUnit
- Test documentation in Syntax #2
- Information broker (communication channel)
- Automatic Train Protection, Interlocking, HMI, Track Database

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 2: Graphical Syntax

Early Draft
ES 203 119-2
V0.0.1
(2014-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Exchange Format

Early Draft
ES 203 119-3
V0.0.1
(2014-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Extensions: Advanced Test Objective Specification

Stable Draft
ES 203 119-1
V1.2.1
(2014-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

Test documentation in Syntax #2

Executable tests

NUnit
• **Context**
  - Testing communication between independent rail sub-systems

• **Challenges**
  - High-level concurrency and non-determinism
  - Multiple aspects over the whole system - safety, real-time, functionality
  - Different development techniques for different components
Requirements
User stories
System models

Executable tests
NUnit

Test documentation
in Syntax #2
Where does TDL fit in?

- User Requirements Notation (URN)
  - Elicitation, analysis, specification, and validation of requirements
  - Complementary views - goals (GRL) and scenarios (UCM)
  - ITU-T Recommendation Z.151 (10/12)
• Context
  • Test systems for cockpit systems and avionics solutions
  • Alternative means for
    • standards-based and model-based test generation and test automation
    • replace proprietary solutions
  • Transformation from high-level requirements and scenarios in UCM to TDL
  • Transformation from TDL to TTCN-3
• **Goals**
  
  • URN/UCM suitable starting point for modelling requirements?
  
  • TDL appropriate intermediate representation or even starting point?
  
  • TTCN-3 viable technology in the avionics industry?

• **Stakeholders**
  
  • Research, industry, agencies
  
  • Test engineers, test developers, test managers, analysts and modellers
• **Motivation**

• Tree-like structure of tests
  - TDL/TTCN-3 reflect this, existing transformations from UCM to e.g. MSC/UML do not

• UCMs do not include much data information
  - appropriate stage to add data for executable test cases (UCM/TDL/TTCN-3/other)?

• Peculiarities of the domain
  - support testing in an environment where an unknown number of sensors can send alarms (over unreliable channels) and messages in parallel
• TDL
  • Close enough to UCM for test generation
  • Close enough to TTCN-3 for generating executable test cases and test configurations

• Prototype
  • Part of jUCMNav (v6.0.0), developed at EECS (University of Ottawa)
  • Support for sequence and concurrent events (no alternatives yet)

http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/ExportTdlUserGuide
http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/ExportTdlUserGuide
Concluding remarks

- New technology, growing rapidly
- Open-source reference implementation under way
  - lower barrier to entry, accelerate adoption
  - commercial tool support not yet available
- Custom tools can be put together in a matter of hours
  - basic yet capable
  - make early adoption easier
- Advanced solutions still require additional effort
  - not immediately necessary to get started with using TDL
Summary

What is TDL?

- Test Description Language
- Design, documentation, and representation of formal test descriptions
- Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)

Where does TDL fit in?

Keyword-Driven Testing

MBT

Generation

Standards

Representation

Rail

Visualisation

Documentation

Interoperability

ITS

Conformance

Where would you consider using TDL?


TDL - tdl.etsi.org
What would you want to see in TDL?

[Image of ETSI's Bug Tracker]

Logged in as: makedonski (Philip Makedonski - manager) 13-09-2014 22:00 IST

Project: TDL

Logged in as: makedonski (Philip Makedonski - manager) 13-09-2014 22:00 IST

Project: TDL

Viewing Issues (1 - 10 / 10)

- [ Print Reports ]
- [ CSV Export ]
- [ Excel Export ]

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Severity</th>
<th>Status</th>
<th>Updated</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006768</td>
<td>TDL meta-model</td>
<td>major</td>
<td>assigned (Andreas Ulrich)</td>
<td>01-08-2014</td>
<td>New MM element as the starting point of the Behaviour Description of a Test Description</td>
</tr>
<tr>
<td>0006773</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Accessing DataProxy arguments</td>
</tr>
<tr>
<td>0006765</td>
<td>TDL meta-model</td>
<td>major</td>
<td>assigned (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Time Observation</td>
</tr>
<tr>
<td>0006764</td>
<td>TDL meta-model</td>
<td>minor</td>
<td>assigned (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Description of VerdictType shall be modified</td>
</tr>
<tr>
<td>0006763</td>
<td>TDL meta-model</td>
<td>minor</td>
<td>resolved (Andreas Ulrich)</td>
<td>31-07-2014</td>
<td>Blocks of ParallelBehaviour should be able to declare Guards</td>
</tr>
<tr>
<td>0006767</td>
<td>TDL</td>
<td>minor</td>
<td>resolved (Andreas Ulrich)</td>
<td>11-07-2014</td>
<td>Allow to reference test descriptions that run on a different test (sub-) configuration</td>
</tr>
<tr>
<td>0006772</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Variable assignment from Interaction and ActionReference</td>
</tr>
<tr>
<td>0006771</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Component variables</td>
</tr>
<tr>
<td>0006770</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Named parameters</td>
</tr>
<tr>
<td>0006769</td>
<td>TDL</td>
<td>feature</td>
<td>assigned (Andreas Ulrich)</td>
<td>10-06-2014</td>
<td>Move parameters from DataInstance to DataSet</td>
</tr>
</tbody>
</table>

MantisBT 1.2.14 [^]
Copyright © 2000 - 2014 MantisBT Team
denis.filatov@etsi.org

MantisBT 1.2.14 [^]
Copyright © 2000 - 2014 MantisBT Team
denis.filatov@etsi.org

STF 492 - [https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx](https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx)
TDL - [tdl.etsi.org](tdl.etsi.org)
Getting started with TDL?

- Cook up your own tooling?
- Demos
  - CEA - “Embedding TDL into the UML environment”
  - Elvior - “Visualising generated tests with TDL”
  - MetaCase - “Custom representations and editors for TDL”
  - UG - “TDL in education with custom tooling”
- Visit us at the TDL booth for further information!
Applying TDL in Practice:
A Hands-on Tutorial

Philip Makedonski, Gusztav Adamis, Martti Käärik,
Finn Kristoffersen, Xavier Zeitoun


© ETSI 2015. All rights reserved