TESTING AND MODELING WITH TDL

Philip Makedonski, Gusztav Adamis, Martti Käärik, Finn Kristoffersen, Andreas Ulrich, 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)

Why TDL?
- For users
  - separate test specification from test implementation
  - amenable to tool-supported verification
  - adjustable to stakeholders
  - focus on what to test vs how

- For tool vendors
  - universal standardised exchange format
  - support for customers from different domains
  - reuse of and integration with existing tools
  - focus on core expertise, add value through interoperability

Where does TDL fit in?
- Keyword-Driven Testing
- MBT Generation
- Representation Standards
- Documentation Visualisation
- ITS Interoperability
- Conformance
What is TDL?

- Test Description Language
  - Design, documentation, and representation of formalised 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?
  - canonical reference
  - stable documentation
  - clear semantics
  - interoperability and independence
  - 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
Semantics

A 'GateType' represents a type of communication points, called 'GateInstance's. A 'GateType' specifies the 'DataType's that can be exchanged via 'GateInstance's in both directions.

Generalization

- PackageableElement

Properties

- dataType: DataType [1..*] {unique}
  The 'DataType's that can be exchanged via 'GateInstance's shall adhere to the 'DataType's that are allowed to be exchanged.

Constraints

There are no constraints specified.
6.4.2 GateType

Concrete Graphical Notation

- **Data Type**: `DATATYPELISTLABEL`

Formal Description

```plaintext
context GateType
GATETYPENAMELABEL :: self.name
DATATYPELISTLABEL :: self.dataType.name->separator(’,,)
```

Comments

No comments.
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 syntax 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.” These examples are explained below - two examples translated from EBNF descriptions taken from [1.2] and [1.3], as well as an example illustrating some of the TDL data parameterization concepts. The examples are accompanied by a complete reference description of the textual format that supports all features of the TDL meta-model.

B.2 A 3GPP Conformance Example in Textual Syntax

This example describes one possible way to translate clause 7.1.3.1 from TS 14 1523 (2) into the proposed textual syntax, by mapping the concepts from the representation in the source document to the corresponding concepts in the TDL meta-model by means of the proposed textual syntax. The example has been enriched with additional information, such as explicit data definitions and test configuration details for completeness when applicable.
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
}
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 $gt$ 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.$g$;
}
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**

…
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

ETSI ES 203 119-3 V1.2.1 (2016-09)
Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 3: Exchange Format
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

```
<packagedElement xsi:type="tdl:ComponentType" xmi:id="_qKt23asEeWrfP0MdfQNpg" name="ct">
  <gateInstance xmi:id="_qKt24HasEeWrfP0MdfQNpg" name="g" type="_qKt23nasEeWrfP0MdfQNpg"/>
</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

- 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

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?
What is TDL?
Why TDL?

- Model-based Testing
  - modelling features to describe and generate abstract tests
- Test automation
  - support for common testing patterns
  - clear semantics for generating concrete tests
- Agile Development
  - test-driven and behaviour-driven development
  - scenario-based testing from user stories
  - multiple representation formats for different stakeholders
Why TDL?

- For users
  - separate test specification from test implementation
  - amenable to tool-supported verification
  - adjustable to stakeholders
  - focus on what to test vs how
- For tool vendors
  - universal standardised exchange format
  - support for customers from different domains
  - reuse of and integration with existing tools
  - focus on core expertise, add value through interoperability
Why TDL?

- Transparent change management

Why TDL?

- Proof by implementation
Why TDL?

- Why not UTP?
  - requires knowledge and understanding of UML
  - loose semantics inherited from UML
  - tool-specific implementations
  - poor transferability as a consequence
  - additional profiles for timing aspects
A more comprehensive example...

Structured Data Instance
Request:Message

sessionId:=
content:=Greeting001

Structured Data Type
Message

sessionId:Integer
content:String

Node

Variable
name:String

socket:Data

Test Configuration
ClientAndServer

TESTER
Server:Node

socket

SUT
Client:Node

socket
- **Atomic**

- **Combined**

- **Exceptional**

  ```
  • Atomic
  
  • Combined
  
  • Exceptional
  ```
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 2: Graphical Syntax

ETSI STANDARD

ETSI ES 203 119-3
V1.2.1 (2016-09)
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 3: Exchange Format

ETSI STANDARD

ETSI ES 203 119-4
V1.2.1 (2016-09)
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

ETSI STANDARD

ETSI ES 203 119-1
V1.3.1 (2016-09)
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

TESTER
Server:Node

SUT
Client:Node

socket

Request (sessionId := 1)

Accept (sessionId := 1)

PASS

Accept (sessionId := ?)

DEMO

FAIL

alternative

interrupt

default

Tick

Tock

Reject

INCONCLUSIVE

DEMO
Where does TDL fit in?

Front-end tool  Back-end tool  Artefact (output)  (Partially) Covered in reference implementation

Textual Editor (incl. ES 203119-4)  UML-based Graph. Editor  TDL Model Analyser  TDL Test Generator

Graphical Editor (ES 203119-2)  Graph. Viewer & Doc. Gen.  Test Plan  Test Code Generator

TDL Exchange Format (ES 203119-3)

C-code, TTCN-3
Where does TDL fit in?

Keyword-Driven Testing

MBT

Generation

Standards

Documentation

Visualisation

Representation

ITS

Interoperability

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  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 )
ensur{e 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 )
}

7.2.2.3.2  Test procedure sequence

Table 7.2.2.3.2.1: 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>
From 3GPP TS 36.523-1 V10.2.0 (2012-09):

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 {
  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 7.2.2.3.3.2-2 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>

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; (RLC SDU#1)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Check: Does the UE transmit RLC SDU#2?</td>
<td>--&gt; (RLC SDU#2)</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>Check: Does the UE transmit RLC SDU#3?</td>
<td>--&gt; (RLC SDU#3)</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>The UE transmits RLC SDU#4.</td>
<td>--&gt; (RLC SDU#4)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

7.2.2.3.3 Specific message contents

None.
From 3GPP TS 36.523-1 V10.2.0 (2012-09):

```c
// Test description definition
Test Description TD_7.1.3.1 uses configuration defaultTC {
    // Pre-conditions and preamble from the source document
    perform action preCondition with { PRECONDITION ; } ;
    perform action preamble with { PREAMBLE ; } ;

    // Test sequence
    SS.g sends pdcch (c_rnti=ue) to UE.g with {
        STEP : "1" ;
        PROCEDURE : "SS transmits a downlink assignment
                      including the C-RNTI assigned to
                      the UE" ;
    } ;
    SS.g sends mac_pdu to UE.g with {
    UE.g sends harq_ack to SS.g with {
    set verdict to PASS ;
    SS.g sends pdcch (c_rnti=unknown) to UE.g with {
    SS.g sends mac_pdu to UE.g with {
```
From ETSI TS 186 011-2 V3.1.1 (2011-06):

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>
<tr>
<td>References</td>
<td>Test Purpose</td>
</tr>
<tr>
<td></td>
<td>TP_IMS_4002_1</td>
</tr>
<tr>
<td>Use Case ref.:</td>
<td>UC_05_1</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)

Test Sequence:
1. User A sends message to User B with at least 1 500 characters
2. Verify that user B receives message from user A

Conformance Criteria:
1. TP.IMS_4002_01 in CFW step 4 (MESSAGE) ensure that {
   when { UE_A sends a MESSAGE to UE_B containing a Message_Body greater than 1 300 bytes }
   then { IMS_B receives the MESSAGE containing the MESSAGE_Body greater than 1 300 bytes }
}

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

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

**Interoperability Test Description**

| Identifier: | TD_IMS_MESS_0001 |
| Summary: | IMS network shall support SIP messages greater than 1 500 bytes |
| Configuration: | CF_INT_CALL |
| SUT | IMS_B |

<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>

**Step Sequence**

<table>
<thead>
<tr>
<th>Step</th>
<th>User A</th>
<th>User B</th>
<th>IMS_A</th>
<th>IBCF_A</th>
<th>IMS_B</th>
<th>IBCF_B</th>
<th>User A</th>
<th>User B</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UE_A</td>
<td></td>
<td>IMS_A</td>
<td>IBCF_A</td>
<td>IMS_B</td>
<td>IBCF_B</td>
<td>UE_A</td>
<td>UE_B</td>
<td>User A sends an instant message to user B</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UE_A</td>
<td>IMS_A</td>
<td>UE_A sends MESSAGE to IMS_A</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A sends MESSAGE to IBCF_A</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IBCF_A sends MESSAGE to IBCF_B</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IBCF_B sends MESSAGE to IMS_B with via header indicating TCP</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B sends MESSAGE to UE_B</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>User B is informed about the instant message</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>UE_B sends 200 OK to IMS_B</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B sends 200 OK to IBCF_B</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IBCF_B sends 200 OK to IBCF_A</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IBCF_A sends 200 OK to IMS_A</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A sends 200 OK to UE_A</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>Optional: User A is presented a delivery report</td>
</tr>
</tbody>
</table>

*Check:*

- User B receives message from user A

**Reference**

- ETSI TS 186 011-2 V3.1.1 (2011-06)

**Interoperability Test Description**

**Step 2:**

<table>
<thead>
<tr>
<th>Step</th>
<th>User A</th>
<th>User B</th>
<th>IMS_A</th>
<th>IBCF_A</th>
<th>IMS_B</th>
<th>IBCF_B</th>
<th>User A</th>
<th>User B</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UE_A</td>
<td></td>
<td>IMS_A</td>
<td>IBCF_A</td>
<td>IMS_B</td>
<td>IBCF_B</td>
<td>UE_A</td>
<td>UE_B</td>
<td>User A sends an instant message to user B</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A sends MESSAGE to IBCF_A</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IBCF_A sends MESSAGE to IBCF_B</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IBCF_B sends MESSAGE to IMS_B with via header indicating TCP</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B sends MESSAGE to UE_B</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>User B is informed about the instant message</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>UE_B sends 200 OK to IMS_B</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B sends 200 OK to IBCF_B</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IBCF_B sends 200 OK to IBCF_A</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IBCF_A sends 200 OK to IMS_A</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A sends 200 OK to UE_A</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_A</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>IMS_B</td>
<td>Optional: User A is presented a delivery report</td>
</tr>
</tbody>
</table>

**Reference**

- ETSI TS 186 011-2 V3.1.1 (2011-06)
From ETSI TS 186 011-2 V3.1.1 (2011-06):
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
  }
}
Keyword-Driven Testing
• **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)
Generation

Representation

Visualisation
• 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
MBT Representation

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

ETSI STANDARDS
ETSI ES 203 119-2
V1.2.1 (2016-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 3: Exchange Format

ETSI STANDARDS
ETSI ES 203 119-3
V1.2.1 (2016-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

ETSI STANDARDS
ETSI ES 203 119-1
V1.3.1 (2016-09)

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

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

- New technology, growing rapidly
- Open-source reference implementation
  - 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)

Why TDL?

- For users
  - separate test specification from test implementation
  - amenable to tool-supported verification
  - adjustable to stakeholders
  - focus on what to test vs how

- For tool vendors
  - universal standardised exchange format
  - support for customers from different domains
  - reuse of and integration with existing tools
  - focus on core expertise, add value through interoperability

Where does TDL fit in?

Keyword-Driven Testing

MBT Generation Standards Documentation Interoperability

Representation Rail Visualisation ITS Conformance

Where would you consider using TDL?

tdl.etsi.org
What would you want to see in TDL?
Testing and Modeling with TDL

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

dl.etsi.org
© ETSI 2016. All rights reserved