Tutorial: Automated Interoperability Testing Using TTCN-3

Thomas Rings
University of Göttingen
Germany

Stephan Schulz
Conformiq Software OY
Finland

Luca Serazio
Telecom Italia
Italy

TTCN-3 User Conference 2010, Beijing, China

© ETSI 2010. All rights reserved
Acknowledgements

- Specialists Taskforce (STF) 370
  - Developed the presented materials
  - Team of 12 specialists with various background from industry and academia
  - Work started in February 2009 and ends in July 2010

- Committee for Methods for Testing and Specification (MTS) at the European Telecommunications Standards Institute (ETSI)
  - Initiation, guidance and monitoring of the project

- STF 370 has been funded in part by the European Commission (EC) 2008 ICT Standardisation Action Plan SA/ETSI/2008-10
Outline

1. Methodology and Framework for Automated Interoperability Testing

2. Development Process for an Interoperability Test System
“Interoperability is the ability of two systems to interoperate using the same communication protocol”

Context of
- Complex and distributed systems
- Multi-vendor, multi-network, multi-service environment
- Interoperability test events (e.g. ETSI Plugtest)

Goals of automated interoperability testing
- Assure interoperability
- Assess and validate that systems follow standards
- Assess and validate standards
“Interoperability is the ability of two systems to interoperate using the same communication protocol”

- Provide users a specific End to End (E2E) functionality/service
- Exchange information across standardized interfaces

- E2E functional testing
- Conformance checks
### Motivation for a Methodology (2/2)

<table>
<thead>
<tr>
<th>E2E testing</th>
<th>Conformance testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>❐ may require complex set up</td>
<td>❐ can be expensive (lots of tests)</td>
</tr>
<tr>
<td>❐ can verify correct service provision to end users</td>
<td>❐ can not guarantee system interoperability especially for the application layer</td>
</tr>
<tr>
<td>❐ does not ensure adherence to standard specifications</td>
<td></td>
</tr>
</tbody>
</table>

Need for a methodology to enable automated interoperability testing with conformance checking.

EUT: Equipment Under Test
SUT: System Under Test
Verdicts in Interoperability Testing

- **E2E interoperability verdict**
  - Pass: all EUTs interoperate as required by the test
  - Fail: at least one EUT does not interoperate as required by the test

- **Conformance verdict**
  - Pass: Fulfils the conformance test objective
  - Fail: non-conformance to a normative requirement
  - Inconclusive: neither a pass nor a fail verdict can be given
Automation of Interoperability Testing (1/2)

- **Testing tasks that can be automated**
  - Configuration of EUTs
  - Monitoring of relevant standardized interfaces
  - Emulation of equipment
  - Test execution

- **Degree of possible automation**
  - Ideally all the tasks mentioned above should be automated
  - Usually automation of a subset
Automation of Interoperability Testing (2/2)

- **Possible limitations**
  - Accessibility of SUT interfaces
  - Complexity or cost of implementing interface access and checks
  - Unstable specifications

- **Benefits**
  - Reduces time needed for test execution and evaluation
  - Avoids repetitive manual activity
  - Saves costs related to human experts and test bed occupation
  - Reduces time to market
A Framework for Automated Interoperability Testing

Means of Interoperability Testing

Test Coordinator

Test Oracle

Application Support Nodes

EUT 1

EUT 2

SUT

EUT N

Equipment User

Interconnecting Network

Configuration interface

Monitoring interface

Stimulating interface
Outline

1. Methodology and Framework for Automated Interoperability Testing

2. Development Process for an Interoperability Test System
World Class Standards

Development Process for an Interoperability Test System

- Language independent
- In this tutorial, explained and exemplified with specific technologies
  - IP Media Subsystem (IMS)
  - Testing and Test Control Notation Version 3 (TTCN-3)
Introduction to IMS

- IMS = IP Multimedia Subsystem
  - 3GPP standard
  - One of the key enablers of next generation networks
- Access independent platform
- A peer-to-peer architecture
  - Can be split into user, control and service layer
  - Signalling mainly based on the Session Initiation Protocol (SIP)
- Call Session Control Function (CSCF)
  - SIP servers or proxies used to process SIP signaling packets in the IMS.
World Class Standards

IMS Core Networks

- Interfaces between logical entities are standardized
Introduction to TTCN-3

- Internationally standardized testing language
- Developed and maintained by the ETSI Technical Committee MTS
- Specifically designed for testing and certification
- Can be applied to a variety of application domains and types of testing
- Proven to work in very large and complex industrial tests, e.g., of 3G network elements
  - TTCN-3 test suites for IMS, LTE and SIP

Source: www.ttcn-3.org
TTCN-3
Conceptual Test System Architecture

- The test system architecture is standardized via two interfaces
  - TTCN-3 Control Interface (TCI)
  - TTCN-3 Runtime Interface (TRI)
Development of Interoperability Test Systems in Practice

Prerequisites
- Interoperability Testing Framework
  - Test Descriptions
  - Test Architecture
  - Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)
- Specify Test Cases
  - Equipment Operation Repository
  - Conformance Check Repository
- Implement Codec and Adaptation Functions

Abstract Test Suite (ATS)
- Validate

Limitations
About ETSI’s Interoperability Testing Framework

- A set of guidelines how to specify interoperability tests
- Realizes concepts and general architecture defined in the ETSI automated Interoperability Testing (IOT) methodology in TTCN-3
- Separates conformance and interoperability verdict management
- Supports live vs. offline interoperability test execution

<table>
<thead>
<tr>
<th>Capture Mode</th>
<th>offline</th>
<th>live</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message checking</td>
<td>Performed on traffic capture files <strong>after</strong> testing</td>
<td>Performed on live capture <strong>during</strong> testing</td>
</tr>
<tr>
<td>Equipment operation</td>
<td>Disabled</td>
<td>manually or automatically</td>
</tr>
</tbody>
</table>
About ETSI’s Interoperability Testing Framework

- Basis for test component type, test interface, and test parameter definition
- Can be implemented in a library based design
  - Libiot library
    - basic functionalities of IOT entities, i.e., test coordinator, equipment user, interface monitor, test oracle
A Framework for Automated Interoperability Testing

Means of Interoperability Testing

Test Oracle

Test Coordinator

Application Support Nodes

EUT 1

EUT 2

EUT N

Equipment User

Interface Monitor

Interconnecting Network

SUT

Configuration interface

Monitoring interface

Stimulating interface
type component OracleServer extends ServerSyncComp {
    port VerdictPort vPort;
    var VerdictType vc_e3e_verdict := {none, "init"};
    var VerdictType vc_conf_verdict := {none, "init"};
}

type component OracleClient extends ClientSyncComp {
    port VerdictPort vPort;
}

type component TestCoordinator extends OracleServer {
    var ComponentIdList vc_compIds;
    port AdapterConfigPort acPort;
}
A Framework for Automated Interoperability Testing

Means of Interoperability Testing

- Test Oracle
- Test Coordinator
- Application Support Nodes
- Equipment User
- Interface Monitor
- Interconnecting Network

- Configuration interface
- Monitoring interface
- Stimulating interface
Interoperability Testing Framework
Test Entities Definitions

type component EquipmentUser extends OracleClient{
    port EquipmentAccessPort eaPort;
    timer t_equipment;
}

type component InterfaceMonitor extends OracleClient {
    timer tc_wait := PX_MAX_MSG_WAIT;
    var charstring vc_interfaceName := "UNDEFINED";
    port AdapterConfigPort acPort;
}
Interoperability Testing Framework
Test Parameters

- Libraries are customized by setting test parameters
  - In the case of TTCN-3: module parameters
- Configuration information for each EUT
  - IP address(es), port(s), domain name
- Time related parameters
  - Timeout or duration time
- Test session pairing parameters
  - Test case specifications independence of specific EUT configuration information
Development of Interoperability Test Systems in Practice
Prerequisites and Inputs

Prerequisites
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)

Abstract Test Suite (ATS)
- Specify Test Cases
- Implement Codec and Adaptation Functions
- Equipment Operation Repository
- Conformance Check Repository

Validate
Test Architecture

Example: Interworking IMS Core Networks

- A test architecture is an abstract description of logical entities as well as their interfaces and communication links involved in a test.
Development of Interoperability Test Systems in Practice

Prerequisites and Inputs

Prerequisites
- Executable Test Suite (ETS)
- Interoperability Test System Design

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Abstract Test Suite (ATS)
- Specify Test Cases

Executable Test Suite (ETS)
- Implement Codec and Adaptation Functions
- Conformance Check Repository
- Equipment Operation Repository

Validate

Limitations
- Test Descriptions
- Test Architecture
- Interoperability Testing Framework
- Limitations
Ideally unambiguous, informal descriptions of test cases usually written in English prose

Capture

- All equipment required for a test
- Pre-conditions of a test
- Equipment operation and observation during a test
- Protocol messages or procedures to be checked

An interoperability test description is derived from and includes an interoperability test purpose

- Can also include conformance test purposes
  - Interoperability tests with conformance checks
## Interoperability Test Description

### Example: IMS Call Cancelation by Calling User

<table>
<thead>
<tr>
<th>Identifier:</th>
<th>TD_IMS_CALL_0014</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOP TP:</td>
<td>IMS network handles correctly cancels calling user call before its establishment</td>
</tr>
<tr>
<td>Test Architecture:</td>
<td>Interworking IMS Core Networks (CF_INT_CALL)</td>
</tr>
<tr>
<td>Specification Reference</td>
<td>TS 124 229 [1], clause 5.4.3.2 ¶49</td>
</tr>
</tbody>
</table>

### Pre-test conditions:
- Home Subscriber Server (HSS) of IMS_A and of IMS B is configured according to table 1
- UE_A is registered in IMS_A using any user identity
- UE_B is registered in IMS_B using any user identity

### Test Sequence:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User A calls User B</td>
</tr>
<tr>
<td>2</td>
<td>Verify that User B is informed of incoming call of User A</td>
</tr>
<tr>
<td>3</td>
<td>Verify that User A is informed that UE_B is ringing</td>
</tr>
<tr>
<td>4</td>
<td>User A cancels the call, before User B answers or before network timeout</td>
</tr>
<tr>
<td>5</td>
<td>Verify that user B is informed that call has been cancelled</td>
</tr>
<tr>
<td>6</td>
<td>Verify that user A is informed that call is terminated</td>
</tr>
</tbody>
</table>

### Conformance Criteria:

<table>
<thead>
<tr>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>TP_IMS_5107_03 in CFW step 16 (CANCEL):</td>
</tr>
<tr>
<td>ensure that</td>
</tr>
<tr>
<td>when { UE_A sends CANCEL to UE_B }</td>
</tr>
<tr>
<td>then { IMS_B receives the CANCEL</td>
</tr>
<tr>
<td>containing no Route_header</td>
</tr>
<tr>
<td>indicating the S-CSCF_SIP_URI of IMS_A }</td>
</tr>
</tbody>
</table>
Test Sequence

UE_A
Call User B
Cancel Call
Call terminated

IMS_A
Call User B
Cancel Call

IMS_B
Call User B
Ringing
Cancel Call
Call canceled

UE_B
## Interoperability Test Description

**Example: IMS Call Cancelation by Calling User**

<table>
<thead>
<tr>
<th>Identifier:</th>
<th>TD_IMS_CALL_0014</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IOP TP:</strong></td>
<td>IMS network handles correctly cancels calling user call before its establishment</td>
</tr>
<tr>
<td><strong>Test Architecture:</strong></td>
<td>Interworking IMS Core Networks (CF_INT_CALL)</td>
</tr>
<tr>
<td><strong>Specification Reference:</strong></td>
<td>TS 124 229 [1], clause 5.4.3.2 ¶49</td>
</tr>
</tbody>
</table>
| **Pre-test conditions:** | • Home Subscriber Server (HSS) of IMS_A and of IMS B is configured according to table 1  
• UE_A is registered in IMS_A using any user identity  
• UE_B is registered in IMS_B using any user identity |

<table>
<thead>
<tr>
<th>Test Sequence:</th>
<th>Step</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User A calls User B</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Verify that User B is informed of incoming call of User A</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Verify that User A is informed that UE_B is ringing</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>User A cancels the call, before User B answers or before network timeout</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Verify that user B is informed that call has been cancelled</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Verify that user A is informed that call is terminated</td>
<td></td>
</tr>
</tbody>
</table>

**Conformance Criteria:**

Check

1. TP_IMS_5107_03 in CFW step 16 (CANCEL):
   
   ensure that {
     when { UE_A sends CANCEL to UE_B }  
     then { IMS_B receives the CANCEL  
       containing no Route_header  
       indicating the S-CSCF_SIP_URI of IMS_A } 
   }
Development of Interoperability Test Systems in Practice

Prerequisites and Inputs

Prerequisites
- Executable Test Suite (ETS)
- Interoperability Test System Design
  - Define Test Configuration
  - Define Message Structures
  - Define Test Parameters
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Executable Test Suite (ETS)

Abstract Test Suite (ATS)
- Specify Test Cases
  - Equipment Operation Repository
  - Conformance Check Repository
- Implement Codec and Adaptation Functions

Validate
World Class Standards

Limitations
Example: IMS Core Networks

- Authentication and security
- Accessibility
- Equipment configuration and operation
Development of Interoperability Test Systems in Practice

Interoperability Test Design

Prerequisites

- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design

- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)

- Specify Test Cases
  - Equipment Operation Repository
  - Conformance Check Repository
- Implement Codec and Adaptation Functions

Abstract Test Suite (ATS)

Validate

Specify Test Cases

Implement Codec and Adaptation Functions

Define Test Configuration

Define Message Structures

Define Test Parameters
Interoperability Test Design
General Guidelines

- Use naming conventions
- Use modularization → Library concept
- Develop function design
- Abstract handling of proprietary interfaces
- Configure message skipping in monitoring
- Manage of EUT interface information
- Document code
World Class Standards

Development of Interoperability Test Systems in Practice
Interoperability Test Design

**Prerequisites**
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

**Interoperability Test System Design**
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

**Executable Test Suite (ETS)**
- Specify Test Cases
  - Equipment Operation Repository
  - Conformance Check Repository
- Implement Codec and Adaptation Functions

**Abstract Test Suite (ATS)**
- Validate
World Class Standards

Test Configuration: Abstract Port Mappings/Connections

- **Test System Interface (TSI)**
  - Adapter configuration port (acPort)
  - Data port (dPort)
  - Equipment access port (eaPort)

- **IMS Test Coordinator** acts as Main Test Component (MTC)
- **IMS coordination port** (icpPort) for forwarding messages to MTC
- **Verdict port** (vPort) for communicating local verdicts
Test Configuration

- A concrete instance of a test architecture defined on the basis of test entities, communication links and their connection
- Identifies EUTs, monitor points, and test entities
- Defines entity instance which should realize the functionality of one or more test entity types
World Class Standards

Test Configuration
Example: Interworking IMS Core Networks
Test Configuration
Example: IMS IOT with TTCN-3

type component IotSystemInterface {// system component
        port DataPort dPort;
        port EquipmentAccessPort eaPort;
        port AdapterConfigPort acPort;
    }

type component ImsInterfaceMonitor extends InterfaceMonitor {
        port DataPort dPort;
        port ImsCoordinationPort icpPort;
    }
Test Configuration
Example: IMS IOT with TTCN-3

type port ImsCoordinationPort message {
    inout SipMessage;
}

// Test Configuration: Interworking IMS Core Networks

type record CF_INT_CALL {
    ImsInterfaceMonitor gmA,
    ImsInterfaceMonitor mw,
    ImsInterfaceMonitor gmB optional
}
Development of Interoperability Test Systems in Practice

Interoperability Test Design

Prerequisites
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)
- Specify Test Cases
  - Equipment Operation Repository
  - Conformance Check Repository
- Implement Codec and Adaptation Functions

Abstract Test Suite (ATS)

Validate
Message Structures
Example: TTCN-3 Types for a SIP Cancel

type record Request {  //SIP Request from IETF RFC 3261
  RequestLine requestLine,
  MessageHeader msgHeader,
  MessageBody messageBody optional,
  Payload payload optional
}

type record RequestLine {
  Method method,
  SipUrl requestUri,
  charstring sipVersion
}

type enumerated Method {
  ACK_E,
  BYE_E,
  CANCEL_E,
  INVITE_E, // ...
}
Interoperability Test Design

Prerequisites
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)
- Specify Test Cases
- Implement Codec and Adaptation Functions

Abstract Test Suite (ATS)
- Equipment Operation Repository
- Conformance Check Repository

Validate
Test Parameter Examples

- Specify EUT interface information
  - IP addresses and ports of interfaces, domain names, user identities
- Specify time limit parameter
- Enable and disable observations of specific interfaces
  - Interface(s) may not be accessible
  - Enable/disable execution of associated monitor components in test
- Select capture mode for lower tester adapter, i.e., offline or live
- Configure general lower test adapter settings
  - Dependent on capture mode selection
A module parameter captures all information for each supported interface of each product participating in an interoperability event.

```plaintext
modulepar ProductList PX_PRODUCTS := {
    // productIndex = 0
    productName := "Super IMSCore",
    monitorInterfaces := {
        interfaceName := "Mw",
        interfaceInfo := {
            IpInterfaceInfo := {
                domainName := "pcscf.core.etsi",
                IpAddress := "192.86.1.97",
                portNumbers := {5060}
            },
            domainName := "icscf.core.etsi",
            IpAddress := "192.86.1.98",
            portNumbers := {5060}
        }
    }
}
```
The Development of an Interoperability Test System in Practice – The Abstract Test Suite (ATS)

Prerequisites:
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design:
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS):
- Specify Test Cases
- Implement Codec and Adaptation Functions
- Conformance Check Repository
- Equipment Operation Repository

Abstract Test Suite (ATS):
- Validate

Limitations: 

World Class Standards
Abstract Test Suite (ATS) Specification

- Specific TTCN-3 definitions
  - test configuration management, test case statements, test purpose checking functions
- Imports interoperability testing framework libraries
- Imports (or defines) technology specific libraries (e.g. protocols libraries)
- ATS consists of modules:
  - TestConfiguration, TestSystem, TypesAndValues
  - Functions, Templates, TestSteps, TestCases, TestControl
The Development of an Interoperability Test System in Practice – The Abstract Test Suite (ATS)

Prerequisites
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)
- Specify Test Cases
- Implement Codec and Adaptation Functions
- Equipment Operation Repository
- Conformance Check Repository
- Validate

Abstract Test Suite (ATS)
World Class Standards

Specification of Test Cases
The TTCN-3 Testcase Statement

- Structure follows conventional test case implementation
  - ImsTestCoordinator acts as MTC
  - IotSystemInterface represents the abstract interface to upper and lower test adapters

```plaintext
testcase TC_IMS_CALL_0014() runs on ImsTestCoordinator
system IotSystemInterface {
  // 1. Test configuration setup
  // 2. Preamble
  // 3. Test body
  // 4. Postamble
  // 5. Tear down
}
```
Specification of Test Cases
Test Configuration Setup (1/5)
Specification of Test Cases
Test Configuration Setup (2/5)

- Lower test adapter is configured (e.g., for use of live mode)

![Diagram showing the configuration process between MTC and System, with messages `m_generalConfigurationReq_live`, `m_generalConfigurationRsp_success`, and the response "General configuration succeed".]
Specifying Test Cases

Test Configuration Setup (3/5)

Legend:
- Monitor interface
- Equipment operation interface
- Control interface
- SIP interface
Specification of Test Cases
Test Configuration Setup (4/5)
After all lower test adapter configurations are completed, traffic capture processing is started by the MTC.
testcase TC_IMS_CALL_0014()
runs on ImsTestCoordinator
system IotSystemInterface {
    // 1. Test configuration setup
    // 2. Preamble
    // 3. Test body
    // 4. Postamble
    // 5. Tear down
}
# Specification of Test Cases

## Interoperability Test Description - Preamble

<table>
<thead>
<tr>
<th>Identifier:</th>
<th>TD_IMS_CALL_0014</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOP TP:</td>
<td>IMS network handles correctly cancels calling user call before its establishment</td>
</tr>
<tr>
<td>Test Architecture:</td>
<td>Interworking IMS Core Networks</td>
</tr>
<tr>
<td>Specification Reference</td>
<td>TS 124 229 [1], clause 5.4.3.2 ¶49</td>
</tr>
</tbody>
</table>

### Pre-test conditions:
- HSS of IMS_A and of IMS B is configured according to table 1
- UE_A is registered in IMS_A using any user identity
- UE_B is registered in IMS_B using any user identity

<table>
<thead>
<tr>
<th>Test Sequence:</th>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>User A calls User B</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Verify that user B is informed of incoming call of User A</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Verify that user A is informed that UE_B is ringing</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>User A cancels the call, before User B answers or before network timeout</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Verify that user B is informed that call has been cancelled</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Verify that user A is informed that call is terminated</td>
</tr>
</tbody>
</table>

### Conformance Criteria:

1. TP_IMS_5107_03 in CFW step 16 (CANCEL):
   - Ensure that:
     - When { UE_A sends CANCEL to UE_B }
     - Then { IMS_B receives the CANCEL containing no Route_header indicating the S-CSCF_SIP_URI of IMS_A }
Successful Registration of UE A in IMS A

MTC System User A

\[
\text{vPort} \quad \text{eaPort} \quad \text{m_EO_Request} \quad \text{eaPort} \quad \text{mw_EO_Response} \quad \text{eaPort} \quad \text{“UE registration successful”} \quad \text{vPort}
\]

E2E: Pass

E2E verdict = pass;

done

...
function f_userRegistration(
    in charstring p_publicId, in charstring p_privateId,
    in charstring p_pw)
runs on EquipmentUser {
    f_sendEquipmentCmd(valueof(
        m_EO_Request(c_UES_REGISTRATION, {p_publicId, p_privateId, p_pw}))); // LibIot function
}
testcase TC_IMS_CALL_0014()
runs on ImsTestCoordinator
system IotSystemInterface {
    // 1. Test configuration setup
    // 2. Preamble
    // 3. Test body
    // 4. Postamble
    // 5. Tear down
}
### Interoperability Test Description

<table>
<thead>
<tr>
<th>Identifier:</th>
<th>TD_IMS_CALL_0014</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOP TP:</td>
<td>IMS network handles correctly cancels calling user call before its establishment</td>
</tr>
<tr>
<td>Test Architecture:</td>
<td>Interworking IMS Core Networks</td>
</tr>
<tr>
<td>Specification Reference</td>
<td>TS 124 229 [1], clause 5.4.3.2 ¶49</td>
</tr>
</tbody>
</table>
| Pre-test conditions: | • HSS of IMS_A and of IMS_B is configured according to table 1  
                          • UE_A is registered in IMS_A using any user identity  
                          • UE_B is registered in IMS_B using any user identity |

<table>
<thead>
<tr>
<th>Test Sequence:</th>
<th>Step</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>User A calls User B</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Verify that user B is informed of incoming call of User A</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Verify that user A is informed that UE_B is ringing</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>User A cancels the call, before User B answers or before network timeout</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Verify that user B is informed that call has been cancelled</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Verify that user A is informed that call is terminated</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conformance Criteria:</th>
<th>Check</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 TP_IMS_5107_03 in CFW step 16 (CANCEL):</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ensure that</td>
<td></td>
</tr>
<tr>
<td></td>
<td>when { UE_A sends CANCEL to UE_B }</td>
<td></td>
</tr>
<tr>
<td></td>
<td>then { IMS_B receives the CANCEL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>containing no Route_header</td>
<td></td>
</tr>
<tr>
<td></td>
<td>indicating the S-CSCF_SIP_URI of IMS_A }</td>
<td></td>
</tr>
</tbody>
</table>
// 4. Test body

f_mtc_userInitiateCall(v_ueA, v_userInfoB); // Step 1

f_mtc_userCheckRinging(v_ueB); // Step 2

f_mtc_userCheckPeerIsRinging(v_ueA); // Step 3

f_mtc_userTriggerCancelCall(v_ueA); // Step 4

f_mtc_check_TP_IMS_5107_03_gm(v_gmA, false); // Check1

f_mtc_check_TP_IMS_5107_03_mw(v_mw, false); // Check1

f_mtc_userCheckCallCancelled(v_ueB); // Step 5

f_mtc_userCheckCallEnded(v_ueA); // Step 6
Successful call initiation by UE A

MTC

System

User A

start

m_EO_Request

ePort

ePort

mw_EO_Response

ePort

ePort

“UE call initiation successful”

ePort

ePort

E2E: Pass

done

vPort

vPort

E2E verdict = pass;

...
Specification of Test Cases
Test Body (3/9) – Equipment Operation

- Usage of a generic equipment operation function
  - Sends a IMS UE operation command
  - Sets local test component E2E verdict and reports it to MTC

```c
function f_userCallInitiation(charstring p_publicID)
    runs on EquipmentUser {
        f_sendEquipmentCmd(valueof // LibIot function
            m_EO_Request(c_UUE_CALL_INITIATION, {p_publicID}))
    }
```

### Interoperability Test Description

<table>
<thead>
<tr>
<th>Identifier:</th>
<th>TD_IMS_CALL_0014</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOP TP:</td>
<td>IMS network handles correctly cancels calling user call before its establishment</td>
</tr>
<tr>
<td>Test Architecture:</td>
<td>Interworking IMS Core Networks</td>
</tr>
<tr>
<td>Specification Reference</td>
<td>TS 124 229 [1], clause 5.4.3.2 ¶49</td>
</tr>
</tbody>
</table>
| Pre-test conditions: | • HSS of IMS_A and of IMS B is configured according to table 1  
                         • UE_A is registered in IMS_A using any user identity  
                         • UE_B is registered in IMS_B using any user identity |

### Test Sequence:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User A calls User B</td>
</tr>
<tr>
<td>2</td>
<td>Verify that user B is informed of incoming call of User A</td>
</tr>
<tr>
<td>3</td>
<td>Verify that user A is informed that UE_B is ringing</td>
</tr>
<tr>
<td>4</td>
<td>User A cancels the call, before User B answers or before network timeout</td>
</tr>
<tr>
<td>5</td>
<td>Verify that user B is informed that call has been cancelled</td>
</tr>
<tr>
<td>6</td>
<td>Verify that user A is informed that call is terminated</td>
</tr>
</tbody>
</table>

### Conformance Criteria:

<table>
<thead>
<tr>
<th>Check</th>
<th>Condition</th>
</tr>
</thead>
</table>
| 1     | TP_IMS_5107_03 in CFW step 16 (CANCEL):  
       | ensure that {  
       | when { UE_A sends CANCEL to UE_B }  
       | then { IMS_B receives the CANCEL  
       | containing no Route_header  
       | indicating the S-CSCF_SIP_URI of IMS_A } |
// 4. Test sequence / test body
f_mtc_userInitiateCall(v_ueA, v_userInfoB);  // Step 1
f_mtc_userCheckRinging(v_ueB);  // Step 2
f_mtc_userCheckPeerIsRinging(v_ueA);  // Step 3
f_mtc_userTriggerCancelCall (v_ueA);  // Step 4
f_mtc_check_TP_IMS_5107_03_gm(v_gmA, false);  // Check1
f_mtc_check_TP_IMS_5107_03_mw(v_mw, false);  // Check1
f_mtc_userCheckCallCancelled(v_ueB);  // Step 5
f_mtc_userCheckCallEnded(v_ueA);  // Step 6
function f_mtc_check_TP.IMS.5107.03.gm(
    ImsInterfaceMonitor p_monitorCompRef, boolean p_checkMessage)
runs on ImsTestCoordinator {
    p_monitorCompRef.start(
        f_imsIot_receive(
            {mw_SipRequest(mw_CANCEL_Request_Base(?))}, {},
            {0, omit}, "TP.IMS.5107.03", false, p_checkMessage)
    );
    p_monitorCompRef.done;
}
Specification of Test Cases
Test Body (7/9) – Generic receive example

```c
f_imsIot_receive(
    {mw_SipRequest(mw_CANCEL_Request_Base(?))}, {},
    {0, omit}, "TP_IMS_5107_03", false, p_checkMessage
);
```

Template definitions:
- **Base templates define very basic SIP message checks**
  - Example: mw_CANCEL_Request_Base
- **Complex checks are specified in templates derived from base templates**
  - Example: mdw_TP_IMS_5107_03_mw (modifies mw_CANCEL_Request_Base)

Message skipping | Test Purpose identifier | Message forwarding? | Check or Consume?
--- | --- | --- | ---
Specification of Test Cases
Test Body (8/9) – Mw Conformance Check

```java
function f_mtc_check_TP_IMS_5107_03_mw(
    ImsInterfaceMonitor p_monitorCompRef, boolean p_checkMessage)
runs on ImsTestCoordinator {
    var template SipUrl v_scscfImsAUrl :=
        mw_SipUrl_Host(f_GetEUTScscfAddress(PX_EUT_A));
    p_monitorCompRef.start(
        f_imsIoT_receive(
            {mw_SipRequest(
                mdw_TP_IMS_5107_03_mw(?, v_scscfImsAUrl))},
            {}, {0, omit}, "TP_IMS_5107_03", false, p_checkMessage));
    p_monitorCompRef.done;
}
```
 Specification of Test Cases
Test Body (9/9) – Mw Conformance Check

- Message template for Mw conformance check of TP_IMS_5107_03

<table>
<thead>
<tr>
<th>Conformance Criteria</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP_IMS_5107_03 in CFW step 16 (CANCEL): ensure that { when { UE_A sends CANCEL to UE_B } then { IMS_B receives the CANCEL containing no Route_header indicating the S-CSCF_SIP_URI of IMS_A } }</td>
</tr>
</tbody>
</table>

```haskell
template CANCEL_Request mdw_TP_IMS_5107_03_mw (template CallId p_callId, template SipUrl p_SCSCF_SIP_URI)
modifies mw_CANCEL_Request_Base := {
  msgHeader := {
    route := (omit,
      { fieldName := ROUTE_E,
        routeBody := {
          *, complement(mw_routeBody(p_SCSCF_SIP_URI)), *
        } }) }) }
```
Specification of Test Cases
Testcase Statement – Postamble & Tear Down

testcase TC_IMS_CALL_0014 ()
runs on ImsTestCoordinator
system IotSystemInterface {
    // 1. Test configuration setup
    // 2. Preamble
    // 3. Test body
    // 4. Postamble
    // 5. Tear down
}
Specification of Test Cases
Postamble

MTC

System

User A

start

m_EO_Request

mw_EO_Response

“UE de-registration successful”

ePort

ePort

ePort

ePort

E2E: Pass

E2E verdict = pass;

done

vPort

vPort

…

…

…

Footer text (edit in View : Header and Footer)
Specification of Test Cases
Tear down of test configuration

- Re-establish initial state of the test environment
  - Stop traffic capture in lower test adapter
  - Disconnect and unmap test component ports
  - Removes any selection requirements for next test to be executed

Diagram:

- MTC
  - acPort
  - m_stopTrafficCaptureReq
  - m_stopTrafficCaptureRsp_any

- System
  - acPort
  - ...
The Development of an Interoperability Test System in Practice – The Executable Test Suite (ETS)

Prerequisites
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)

Abstract Test Suite (ATS)
- Specify Test Cases
- Equipment Operation Repository
- Conformance Check Repository
- Implement Codec and Adaptation Functions

Validate
Executable Test Suite (ETS)

- Requires
  - Abstract Test Suite, Codecs, Adaptation functions
- The IMS interoperability test system follows the standardized architecture of a TTCN-3 test system
  - Codec for SIP and extensions required by IMS
  - Lower test adapter for capturing IP traffic
  - Upper test adapter for converting equipment operation requests into instructions for equipment operators

Diagram:
- TTCN-3 Executable
- Test Coordinator
- Interface Monitor
- Equipment User
- System Adapter
- Lower Test Adapter
- Upper Test Adapter
- System Under Test
Example of Configuration of the Test Adapter

- **MTC**
  - m_genConfigReq_Live
  - acPort
  - mw_genConfigRsp
  - acPort
  - "General config successful"
  - acPort
  - m_startTrafficCaptureReq
  - acPort
  - mw_startTrafficCaptureRsp
  - acPort
  - ...

- **System**
  - acPort
  - mw_genConfigRsp
  - acPort
  - ...

- **Gm A**
  - m_SetFilterReq
  - acPort
  - mw_SetFilterRsp
  - acPort
  - "Gm A config successful"
  - mw_SetFilterRsp
  - acPort
  - ...

- **Mw**
  - acPort
  - ...

"Mw config successful"
The Development of an Interoperability Test System in Practice – Validation of the Tests

Prerequisites
- Interoperability Testing Framework
- Test Architecture
- Test Descriptions
- Limitations

Interoperability Test System Design
- Define Test Configuration
- Define Message Structures
- Define Test Parameters

Executable Test Suite (ETS)
- Specify Test Cases
- Equipment Operation Repository
- Conformance Check Repository
- Implement Codec and Adaptation Functions

Abstract Test Suite (ATS)
- Validate
Validation of the Test System

- Assures that the test system reproduces at its interface faithfully the behaviour specified for each test description
- Independent review against the test description
- Execution of the test system against a real SUT
  - Review of test results
- Validation results in improvements of the test specification and corrections of errors in the system adapter
Summary

- Methodology for automated interoperability testing of distributed systems
- Process for the development of interoperability test systems
- IMS Interoperability Test Suite has been applied successfully at the 3rd IMS Plugtest™ in 2009 and will be in future events
- Application to other areas
- Further details see the see references
References


[2] ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Parts 1: TTCN-3 Core Language" (also published as ITU-T Recommendation series Z.140)

[3] ETSI TS 123 228 (V7.15.0): “Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; IP Multimedia Subsystem (IMS); Stage 2 (3GPP TS 23.228 version 7.15.0 Release 7)”


Contact

Thomas Rings

Address

University of Göttingen
Institute of Computer Science
Goldschmidtstraße 7
37077 Göttingen

Email
rings@cs.uni-goettingen.de