CERTIF: a testing methodology and a test bench to assess the Compliance to software defined radio standards

15 November 2018
Agenda

- SDR compliance assessment: the needs
  - Testing methodology
    - From the SDR requirements to the tests
    - Test design process
    - Compliance checkpoints definition
    - Modeling
    - Testing generation
  - The test bench
  - Test of SDR components: an example
  - Conclusion / Q&A
SDR Compliance Assessment

The needs

- COTS
- Scalability
- Applications portability
- Hardware abstraction
- Software modularity & reusability

Needs to assess the compliance to these SDR standards

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Application 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services/devices</td>
<td></td>
</tr>
<tr>
<td>Operational environment</td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Application 3</th>
<th>Application 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services/devices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2018
Assumptions on the nature of the systems under test

- The Software radio platforms
  
  The system under test is an equipment running under an Operational Environment including GPP and/or DSP and FPGA

- The Applications
  
  The system under test is a set of source code files that compiles including C/C++, IDL and VHDL

Assumptions on the compliance check method

- The Software radio platforms
  
  The compliance analysis is performed through dynamic tests by calling platform interfaces

- The Applications
  
  The compliance analysis is performed through source code static analysis. A porting stage is not needed.
SDR conformance assessment: the needs

Testing methodology
- From the SDR requirements to the tests
- Test design process
- Compliance checkpoints definition
- Modeling
- Testing generation

The test bench

Test of SDR components: an example

Conclusion / Q&A
Why going from requirements to tests?

SDR Standard Requirements
(RD & RS APIs, OE, Connectivity, AEPs Profiles, etc.)

Requirements coverage

Tests status ➔ level of compliance

SDR Standard compliance Test repository
Test Strategy

Overview

SDR Standard Requirements
(RD & RS APIs, OE, Connectivity, AEPs Profiles, etc.)

Compliance checkpoints definition

Static Analysis based on rules
- Static Tests

Model-Based Testing
- Functional Behavior Tests

SDR Standard compliance Test repository

Traceability

Copyright © 2018
Static analysis rules are based on source code language and provide/use functions.

This test strategy is applied to the check of the compliance of the applications to the interfaces defined by the software radio standard.

- **Provide**: The aim is to find an implementation of each function of the interface.
  - CORBA into C/C++
  - C/C++
  - VHDL

- **Use**: The aim is to find a call of each function of the interface.
  - CORBA into C/C++
  - C/C++
  - VHDL
Test Strategy
Functional Test design Process

- SDR Standard
- Test Architect
- Requirements Extraction
- Compliance Checkpoints definition
- MBT Model
- Tests Specifications
- Requirements
- Compliance checkpoints
- Abstracts Tests scripts (language dependant)
- Adaptation Layer Specification
- Automation Engineer
Test Strategy

Compliance checkpoints definition

- Compliance checkpoint defines the test objectives
  - Success case(s) or Error case(s) definition
  - Definition of test success criteria
  - Definition of the applicability of the test

Sample on the startTone() function of Audio Device

<table>
<thead>
<tr>
<th>Requirement Identifier</th>
<th>Requirement Text</th>
<th>RCC Identifier</th>
<th>RCC Applicability</th>
<th>Component</th>
<th>RCC Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>The startTone operation provides the user the ability to start the generation of a previously created tone/beep to the device user. - Synopsis: void startTone( In unsigned short toneld ) raises(InvalidToneld); - Return Value: None - State: ENABLED CP::Device::-operationalState.</td>
<td>JTRS_AD_PROVIDE_START_TONE_SUCCESS_001</td>
<td>Platform</td>
<td>GPP</td>
<td>* Success case * the tone or beep identification number is valid * Check the tone is started</td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTRS_AD_PROVIDE_START_TONE</td>
<td>JTRS_AD_PROV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Test design based on the behavior of the system under Test (Model Based Testing)
UML Class diagrams design for abstract test implementation

Test Strategy
Modeling

Functions to call on setup before the test body

Test body

Functions to call on Tear down (return to initial state)

Constraints expression

Automatic Tests Generation

PostCondition (OCL Language)

Precondition (OCL Language)

--- SUT SUCCESS case

if (adcчки.currentAmpEnabled = Enum_Boolean_with_NONE::Enum_Boolean_TRUE) then
  true --- SPDG: TRUE= on
else if (adcчки.currentAmpEnabled = Enum_Boolean_with_NONE::Enum_Boolean_FALSE) then
  if (adcчки.defaultAmpEnabledChanged = true) then
    true
  else false
endif and
  true --- SPDG: FALSE= off
else false

Copyright © 2018
● Constraints design in OCL language allows test generation based on component behavior

● Abstract Tests could be exported into different programming languages
  - C/C++, JAVA, etc ...

● Tests specification could be exported into different formats
  - Database export, XML files, etc ...
Example of C++ test with a start function of a radio Device

- Each generated function is a single test step
- Each test is an assembly of single steps

```cpp
bool JTRS_AD_PROVIDE_API_START_1::setUp()
{
  current_result = m_adapter->api_set_API_Params(<params>);
  current_result = m_adapter->api_get_API_Params(<params>);
  current_result = m_adapter->check_API_Params(<params>);
  current_result = m_adapter->prepare_StartRecording(<params>);
  current_result = m_adapter->api_create(<params>);
  current_result = m_adapter->check_create_Record_ReturnedId(<params>);
  return current_result;
}

bool JTRS_AD_PROVIDE_API_START_1::test()
{
  current_result = m_adapter->api_start(<params>);
  current_result = m_adapter->check_StatusForStarted(<params>);
  return current_result;
}

bool JTRS_AD_PROVIDE_API_START_1::tearDown()
{
  current_result = m_adapter->api_stopAll(<params>);
  current_result = m_adapter->api_destroy(<params>);
  current_result = m_adapter->api_set_API_Params(<default_params>);
  current_result = m_adapter->api_get_API_Params(<default_params>);
  current_result = m_adapter->check_API_Params(<default_params>);
  current_result = m_adapter->bench_tearDown();
  return current_result;
}
```
SDR compliance assessment: the needs

Testing methodology
- From the SDR requirements to the tests
- Test design process
- Compliance checkpoints definition
- Modeling
- Testing generation

The test bench

Test of SDR components: an example

Conclusion / Q&A
The Test Bench

Description

Model based testing tools

Test management tool

Static analysis tools

Dynamic analysis tool (tests harnesses)

Measurement tools

System under test (application / waveform)

System under test (platform / device - service)

Standards
The Test Bench
Objectives

- Maximum automation of test campaigns
- In the same spirit, automatic management of measurement tools
- No dependency between Test management software and test execution software
- Portability of test execution software to address different systems under test
SDR compliance assessment: the needs

Testing methodology
- From the SDR requirements to the tests
- Test design process
- Compliance checkpoints definition
- Modeling
- Testing generation

The test bench

Test of SDR components: an example

Conclusion / Q&A
Test of SDR components

Example of Radio Device API Platform testing

- Test request from Test management Software

![Diagram showing Test Execution software, Domain Manager, Device Manager, System under Test, and Measurement Tools]

- Launch Test Test_API_Func_001
Test of SDR components
Example of Radio Device API Platform testing

- Test request on Radio Device Harness
Test of SDR components
Example of Radio Device API Platform testing

- Test Set Up

Test Execution software

Domain Manager

Application

Radio Device

Tests

Harness

Prepare Capture

Device Manager

GPP Device

Test control

Software

Measurement Tools

System under Test

Naming Service

SetUp():
Set_API_Params()
API_Create()
Etc..
Test of SDR components
Example of Radio Device API Platform testing

- Test main initial phase

Test Execution software

Domain Manager
- Application
- Radio Device Tests Harness
- Capture
- Device Manager
- GPP Device
- Test control Software

System under Test
- Naming Service
- Domain Manager
- Device Manager
- Radio Device

Test() : API_start

Measurement Tools

Test Management Software
Test of SDR components

Example of Radio Device API Platform testing

- Test main capture phase

Diagram showing the integration of various components including:

- Domain Manager
- Application
- Radio Device Tests Harness
- Device Manager
- GPP Device
- Check Capture
- Test Control Software

Also includes:

- Measurement Tools
- Transmission
- Test Execution software

System under Test includes:

- Naming Service
- Domain Manager
- Device Manager
- Radio Device

Copyright © 2018
Test of SDR components

Example of Radio Device API Platform testing

- Test Tear Down

Test Execution software

Domain Manager

Application

Radio Device

Tests

Harness

Device Manager

GPP Device

Test control

Software

System under Test

TearDown() : API_Destroy ()

Etc ..

Naming Service

Domain Manager

Device Manager

Radio Device

Capture Measurement Tools

Test Management Software

Copyright © 2018
Test of SDR components

Example of Radio Device API Platform testing

- Result Return from Harness
Test of SDR components
Example of Radio Device API Platform testing

- Result return to Test Management Software
Agenda

- SDR compliance assessment: the needs
- Testing methodology
  - From the SDR requirements to the tests
  - Test design process
  - Compliance checkpoints definition
  - Modeling
  - Testing generation
- The test bench
- Test of SDR components: an example

Conclusion / Q&A
Conclusion
Results & Perspectives

Results
- 96 % of automated tests among more than 500 tests

Perspectives
- Extensibility of the test bench
  - Evolution of the standard
  - New application or waveform
  - New service or device
  - Middleware adaptation

- Testing methodology and some testing tools reusable for other SDR standards
Questions