|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  | **MSG-134** | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  | **ETC FRA Technical Specifications** | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  |  |  |
|  |  | Version: |  |
|  |  | Date: |  |
|  | Status: |  |
|  | Usage: |  |
|  | Author: |  |
|  | Type: | |  |

***TABLE OF CONTENTS***

1. Introduction 6

1.1 Purpose 6

1.2 reference documents 6

1.3 TerminologY 6

1.3.1 Acronyms 6

1.3.2 Definitions 7

1.4 Requirement management 8

2. General overview 9

2.1 Basic concepts 9

2.2 Certification framework architecture 9

2.3 Test Case architecture 12

2.3.1 Test Case definition 12

2.3.2 Test case configuration 13

2.3.3 Test case execution 17

2.3.4 Technical environment and dependencies 17

2.4 implementation rules 17

3. Specifications 18

3.1 Presentation 18

3.2 Common component "ETC\_FRA\_Common" 18

3.2.1 Role 18

3.2.2 Implementation 18

3.2.2.1 FCTT\_NG reused code 18

3.2.3 Compilation 19

3.2.4 Configuration 19

3.2.5 Execution 19

3.3 Test case "CS Verification" 20

3.3.1 Role 20

3.3.2 Implementation 20

3.3.2.1 Test case main class 20

3.3.2.2 Data model management class 21

3.3.2.3 Parameters management class 21

3.3.2.4 FCTT\_NG reused code 21

3.3.3 Compilation 22

3.3.4 Configuration 22

3.3.4.1 « IVCTsut » subdirectory 23

3.3.4.2 "TestSchedules" subdirectory 23

3.3.5 Execution 23

3.3.5.1 Nominal case (outside Eclipse) 24

3.3.5.2 Development case (for Eclipse debugging) 24

3.3.6 Results 25

3.4 Test Case « HLA Declaration Management » 25

3.4.1 Role 25

3.4.2 Implementation 26

3.4.2.1 Test case main class 26

3.4.2.2 Data model management class 26

3.4.2.3 Parameters management class 27

3.4.2.4 FCTT\_NG reused code 27

3.4.3 Compilation 28

3.4.4 Configuration 28

3.4.5 Execution 28

3.4.6 Results 28

3.5 Test Case « HLA Object Management » 30

3.5.1 Role 30

3.5.2 Implementation 30

3.5.2.1 Test case main class 30

3.5.2.2 Data model management class 31

3.5.2.3 Parameters management class 31

3.5.2.4 FCTT\_NG reused code 31

3.5.3 Compilation 31

3.5.4 Configuration 31

3.5.5 Execution 31

3.5.6 Results 31

3.6 Test Case « HLA Services Verification » 34

3.6.1 Role 34

3.6.2 Implementation 34

3.6.2.1 Test case main class 34

3.6.2.2 Data model management class 34

3.6.2.3 Parameters management class 35

3.6.2.4 FCTT\_NG reused code 35

3.6.3 Compilation 35

3.6.4 Configuration 35

3.6.5 Execution 35

3.6.6 Results 35

4. Hardware resources 39

5. Appendices 40

5.1 SCHEMATRON rules file 40

5.2 « ETC\_FRA\_COMMON.gradle » Gradle generation file 53

5.3 « TS\_CS\_VERIFICATION.gradle » Gradle generation file 54

5.4 ETC CS\_Verification execution logs 56

5.4.1 Nominal case 56

5.4.2 Failed case 57

***Figures***

[Figure 1: IVCT Framework overview 10](#_Toc480792095)

[Figure 2: IVCT / Test Case dependencies 12](#_Toc480792096)

[Figure 3: ETC FRA configuration overview 14](#_Toc480792097)

[Figure 4: Configuration file tree for a specific test case of a federate to be tested (SuT) 15](#_Toc480792098)

[Figure 5: Installation tree for a test case containing an ETC list file 16](#_Toc480792099)

[Figure 6: Configuration file tree for "TS\_CS\_Verification" 22](#_Toc480792100)

[Figure 7: SuT configuration folder for different ETCs 23](#_Toc480792101)

[Figure 8: File trees of a Test Suite for the "CS Verification" ETC 23](#_Toc480792102)

[Figure 9: "Log" window 24](#_Toc480792103)

page

***Tables***

[Table 1: Minimal hardware configuration for ETC FRA 39](#_Toc480792104)

# Introduction

## Purpose

This document is the technical specifications for the IVCT test cases (also called ETC FRA) developed by France, as a part of the effort of development to produce a NATO certification capability within NATO group MSG-134.

This document describes:

* An overview of the architecture of the certification Framework (see §)
* The detailed software architecture of the implemented test cases (certification tool, see §3)
* The hardware resources required to run this software suite (see §4)

Note:

* This document bring some technical details to the functional specifications and aims at doing the development

The document contains some specific symbols to indicate the following types of information:

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| **http://www.atome77.com/images/icones/icone-information.jpg** | Information |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | Warning |

## reference documents

|  |  |  |
| --- | --- | --- |
| **ID** | **Reference** | **Document name** |
| [R1] | MSG-134 Technical Activity Proposal | Technical Activity Proposal MSG-134 NATO Distributed Simulation Architecture & Design, Compliance Testing and Certification |
| [R2] | AC/323 (SGMS)D/2 | NATO Modeling and Simulation Master Plan version 1.0 |
| [R3] | IEEE 1516.1-2010 | IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) – Federate Interface Specification |
| [R4] | IEEE 1516.2-2010 | IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) – Object Model Template (OMT) Specification |
| [R5] | IEEE 1516-2010 | IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) – Framework and Rules |

## TerminologY

### Acronyms

| Acronym | Meaning |
| --- | --- |
| **API** | Application Programming Interface |
| **ATC** | Abstract Test Case |
| **CA** | Certification Agent |
| **ETC** | Executable Test Case |
| **FCTT** | Federate Compliance Testing Tool: tool used by France to certify against HLA 1.3 et 1516-2000 standards |
| **FCTT NG** | Federate Compliance Testing Tool New Generation: tool used by France to certify against 1516-2010 standard |
| **FOM** | Federation Object Model |
| **FRA** | France |
| **FUT** | Federate Under Test: |
| **HLA** | High Level Architecture |
| **HMI** | Human-Machine Interface |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IEEE 1516-2010** | Version of HLA standard from IEEE approved in August 2010 |
| **IP** | Internet Protocol |
| **IR** | Interoperability Requirement |
| **IVCT** | Integration Verification and Certification Tool |
| **JSON** | JavaScript Object Notation |
| **M&S ou MS** | Modelling & Simulation |
| **MOM/MIM** | Management Object Model (MOM) & Initialization Module (MIM) |
| **MSG** | Modelling & Simulation Group (NATO, see also NMSG) |
| **NATO** | North Atlantic Treaty Organization |
| **NMSG** | NATO Modelling & Simulation Group |
| **OMT** | HLA Object Model Template |
| **RSI** | Report Service Invocation |
| **RTI** | Run-Time Infrastructure |
| **SOM** | Simulation Object Model |
| **SuT** | System Under Test |
| **SuTE** | System Under Test Environment |
| **TC** | Test Case |

### Definitions

| Name | Definition |
| --- | --- |
| **Certification Agent** | Person who manages the entire application certification process |
| **Callback** | HLA / RTI terminology corresponding to the implementation of a programming interface to retrieve information from the RTI |

## Requirement management

The requirements of this document (design requirement) are uniquely identified. They are named [IR‑XXX‑ #] with:

* IR meaning "Interoperability Requirement"
* XXX being a string representing the type of requirement:
  + "BP" for Best Practice requirements
  + "DOC" for Documentation Requirements
  + "SOM" for HLA requirements
* # is the requirement number for a type

The identifier is followed by the description of the requirement.

Example of requirement:

[IR-SOM-0002] SuT CS / SOM must be consistent.

The main objective is to ensure full traceability between the requirements of this specification and the test documentation in the validation phase.

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The requirements mentioned in this document have been written by "MSG-134" [R1], no technical requirement were added. |

# General overview

## Basic concepts

IVCT is a software framework for conducting system certifications (SuT).

The test cases are executed using components (ETC) working within IVCT.

Each ETC is an implementation of an abstract test case (ATC) describing how to verify a set of interoperability requirements (IR).

A set of successfully passed ETC allows obtaining a badge (CB).

## Certification framework architecture

The overall architecture of the IVCT Framework can be schematized as follows:

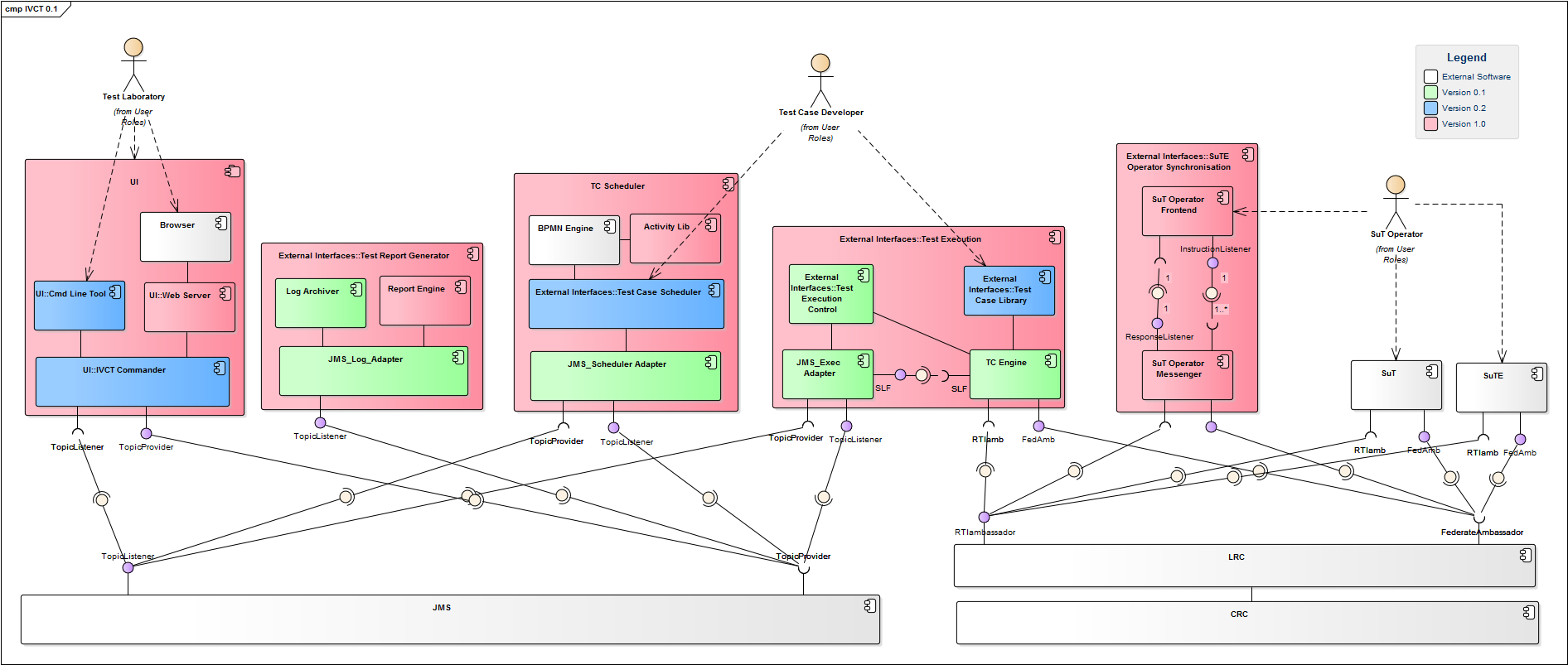


Figure 1: IVCT Framework overview

In this architecture, the implementation of a test case corresponds to the development of a "Test Case Library".

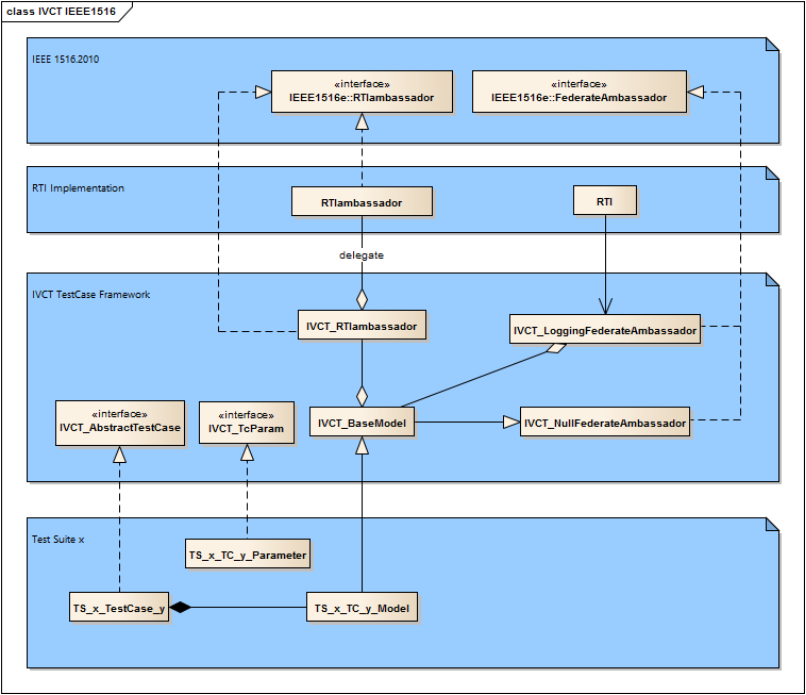
The other major components for the development and use of a test case are:

* "Cmd Line Tool" and "IVCT Commander" for configuration (reading and transmitting configuration parameters of a test case) and execution (sending orders to "TC Engine")
* "TC Engine", which receives orders from the two previous components, dynamically loads the test case (external library, i.e. Java .jar file) and calls the methods to activate that test case

## 

## Test Case architecture

From a "Development" point of view, a test case (here called "Test Suite"[[1]](#footnote-1)) is situated as follows:



Test Case

Figure 2: IVCT / Test Case dependencies

### Test Case definition

As seen in , defining a test case requires deriving the class "IVCT\_BaseModel" from the Framework into a specific class. The role of this class is twofold:

* To contain the data model that will be manipulated by the test case, hence the name "Model"
* To ensure dialogue with HLA federation

To perform this second role, the implementation class of the test case will:

* Use an instance of class IVCT\_RTIambassador (RTI ambassador), provided by the Framework, in order to send information to the RTI,
* Implement callbacks of the following types, in order to receive RTI information:
  + "receiveInteraction" to handle interactions receipt notifications
  + "discoverObjectInstance" to handle object creation notifications
  + "removeObjectInstance" to handle object removal notifications
  + "reflectAttributeValues" to handle notifications about changing attribute values

This class is therefore "directly plugged" on the HLA bus (via the IVCT).

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | When no interaction with the RTI is necessary, the IVCT\_RTIambassador class is not instantiated and no callback is implemented. |

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | The derived class has to implement all the concepts required for the required done for that test case and which are specific to it. |

It is then required to implement the two following interfaces:

* "IVCT\_TcParam" to define the execution parameters needed for the test case, such as the RTI connection parameters. These parameters are defined in a JSON Test Case specific configuration file which is read by the Cmd Line Tool component and then sent to the test case. The implementation class of "IVCT\_TcParam" will then retrieve the values of the parameters and provide the accessors to these parameters as for example:
  + "getFederationName" to return the name of the federation
  + "getRtiHost" to return the RTI connection address

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | This interface implementation has also to manage any other parameter required for the required done and which are specific to the test case. |

* "IVCT\_AbstractTestCase" to define an ETC. This interface requires to define the following methods:
  + "getIVCT\_BaseModel" to initialize the ETC by creating an instance of the previously defined model
  + "logTestPurpose" to log the purpose of the ETC
  + "performTest" to execute the steps of the ETC
  + "preambleAction" to prepare the execution of the ETC (RTI connection,...)
  + "postambleAction" to terminate the execution of the ETC (disconnection RTI,...)

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | All these methods are called by the framework at runtime. |

### Test case configuration

The Framework configuration is based on the following environment variables:

* "IVCT\_CONF" must contain the absolute path of the folder where the "IVCTconfig.xml" file is located, which itself describes the absolute location of the folders where SuT configurations are located
* "IVCT\_HOME" must contain the absolute path of the folder where the IVCT Framework is located
* "IVCT\_TS\_HOME" must contain to the absolute path of the folder where the "IVCTtestsuites.xml" file is located, which itself describes the Java packages of the dynamic libraries of the ETCs

Example of "IVCTconfig.xml" file content:

**<?xml version="1.0" encoding="UTF-8"?>**

**<ivctConfig>**

**<pathNames>**

**<sutDir>D:\\Users\\HLA\\Documents\\GitHub\\ETC\_FRA\_Config\\IVCTsut</sutDir>**

**</pathNames>**

**</ivctConfig>**

Overview:

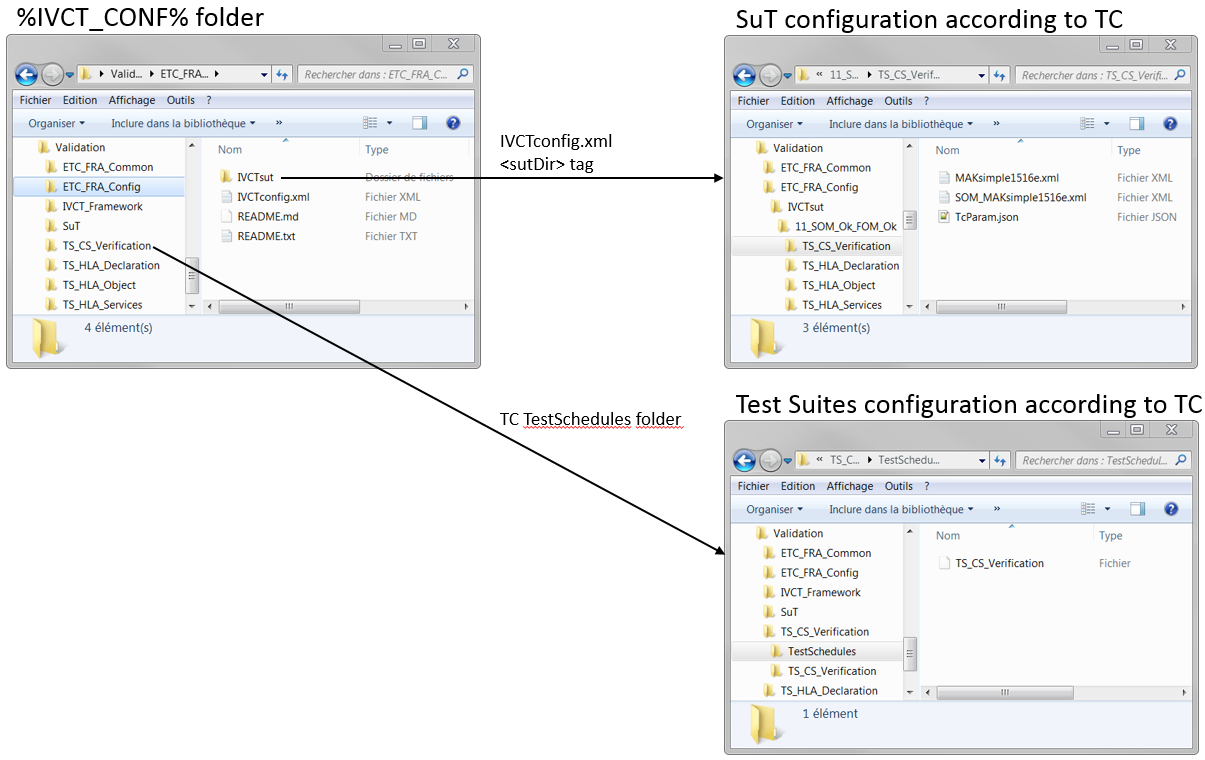


Figure 3: ETC FRA configuration overview

The folder specified in the "<sutDir>" tag contains one subdirectory per federate to be tested (SuT named 11\_SOM\_Ok\_FOM\_Ok in the screenshot below). This subdirectory contains itself a subfolder for each test case that includes a JSON file ("TcParam.json") describing the parameters of the test case. If necessary, other SuT-specific configuration files for this test case are added to the subfolder.

Here is an example of a test case directory content corresponding to a SuT:

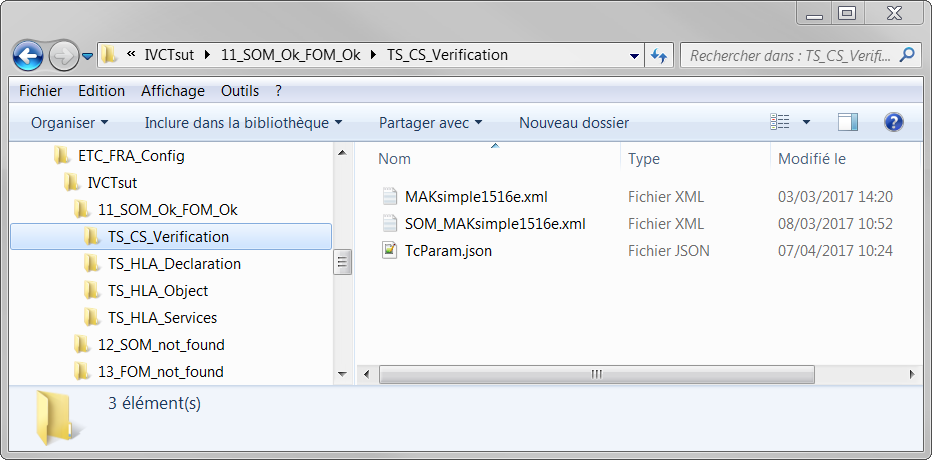


Figure 4: Configuration file tree for a specific test case of a federate to be tested (SuT)

The JSON parameter description file has the following content:

**{**

**"sutName" : "TestFederate"**

**"resultDirectory" : "…\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification"**

**"fomFiles" : [**

**{ "fileName" : "…\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml" }**

**]**

**"somFiles" : [**

**{ "fileName" : "…\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml" }**

**]**

**}**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The file paths in JSON file are "absolute" and, in the previous illustration, the beginning of the paths has been replaced with "..." for greater clarity. |

The folder specified by the "IVCT\_TS\_HOME" variable contains the "IVCTtestsuites.xml" file which indicates IVCT Framework how to access to ETCs:

**<?xml version="1.0" encoding="UTF-8"?>**

**<ivctConfig>**

**<testSuites>**

**<testSuite>**

**<name>TS\_CS\_Verification</name>**

**<packageName>nato.ivct.etc.fr.tc\_cs\_verification</packageName>**

**<tsRunFolder>TS\_CS\_Verification\\TS\_CS\_Verification\\bin</tsRunFolder>**

**</testSuite>**

**<testSuite>**

**<name>TS\_HLA\_Declaration</name>**

**<packageName>nato.ivct.etc.fr.tc\_hla\_declaration</packageName>**

**<tsRunFolder>TS\_HLA\_Declaration\\TS\_HLA\_Declaration\\bin</tsRunFolder>**

**</testSuite>**

**<testSuite>**

**<name>TS\_HLA\_Object</name>**

**<packageName>nato.ivct.etc.fr.tc\_hla\_object</packageName>**

**<tsRunFolder>TS\_HLA\_Object\\TS\_HLA\_Object\\bin</tsRunFolder>**

**</testSuite>**

**<testSuite>**

**<name>TS\_HLA\_Services</name>**

**<packageName>nato.ivct.etc.fr.tc\_hla\_services</packageName>**

**<tsRunFolder>TS\_HLA\_Services\\TS\_HLA\_Services\\bin</tsRunFolder>**

**</testSuite>**

**</testSuites>**

**</ivctConfig>**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The "<name>" tag corresponds to the test case installation subfolder under the folder specified by IVCT\_TS\_HOME environment variable.  The "<packageName>" and "<tsRunFolder>" tags refer respectively to the Java package name and the ETC runtime directory (based on the value of the IVCT\_TS\_HOME environment variable). |

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The file path separator used is "\\" to avoid any "Escape" sequence interpretation. For example, "\b" is interpreted as a "backspace" character. |

The test case installation subfolder contains a (freely-named) file listing the ETCs to be executed. The name of an ETC is a Java class that implements the "AbstractTestCase" interface.

Example of a file containing the list of ETCs to run:

**TC\_001\_Files\_Check**

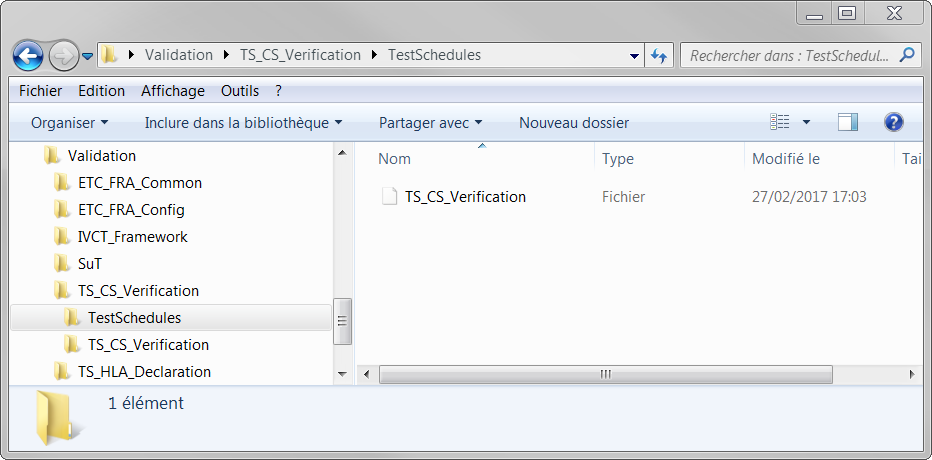


Figure 5: Installation tree for a test case containing an ETC list file

Summary:

* Test case file tree (« TC »):

%IVCT\_CONF%\{IVCTconfig.xml<sutDir> tag}\<*SuT name*>\<*TC name*>

* Test suites file tree (« TS »):

%IVCT\_TS\_HOME%\<*TC name*>\TestSchedules\<*TS name*>

Commands to enter are the following:

* ssut <*SuT name*>
* st <*TC name*>
* sts <*TS name*>

### Test case execution

A test case is called by the "TC Engine" component commanded by the user interface "Cmd Line Tool" of the Framework IVCT. For the IVCT Framework to do this :

* The environment variable "IVCT\_TS\_HOME" must contain the "IVCTtestsuites.xml" file and the external Java libraries of the test cases
* The "CLASSPATH" variable (like any Java application) correctly filled

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | The "CLASSPATH" variable must be set using the "IVCT\_TS\_HOME" variable in the user interface launching script named "%IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat" ("Cmd Line Tool" component) of the IVCT Framework. |

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | These two previous variables must be correctly set and any inconsistency makes the test case impossible to execute (Java "ClassNotFoundException" error). |

### Technical environment and dependencies

The development of ETC FRA was executed under Windows 7.

The technical basis of ETC FRA is Java 8.

Note that ETC FRA depend on:

* Ellipse parser in version 2.6
* RTI commercial products (MÄK RTI and Pitch RTI)

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | Both RTIs are the only licensed software components on which ETCs rely. |

## implementation rules

This chapter defines the main rules for the implementation of test cases observed during their development. These rules are those adopted collectively at NATO MSG-134 level. They are described at the following address on the GitHub site dedicated to the development of IVCT:

<https://github.com/MSG134/IVCT_Framework/wiki/Executable-Test-Case-Development-Rules>.

# Specifications

## Presentation

The objective of the ETC FRA is to allow the certification of an HLA federate against IEEE 1516-2010 standard (cf. documents [R3], [R4] and [R5]).

The two principles used to create the source code for executable test cases (ETC FRA) are as follows:

* They are integrated into the IVCT, so their general architecture respects the rules imposed by that Framework
* They reuse the code of the FCTT NG to provide the same functionalities

In order to respect the distribution logic of ETCs developments jointly defined between nations within MSG-134 (cf. <https://smart-wiki.smart-lab.se/display/msg134/Progress>[[2]](#footnote-2) for identification of test cases developed by different nations), four ETCs are developed:

* CS Verification: "Static" verification of federate FOM and SOM files
* HLA Declaration Management: Verification of the consistency between the publications / subscriptions declared in SOM and those done by the federate at runtime
* HLA Object Management: Verification of the consistency between the updates / receptions of object attributes and interactions parameters declared in SOM and those performed by the federate at runtime
* HLA Services Verification: Verification of the consistency between the services declared as used in SOM and those actually used by the federate at runtime

In addition, a common component named ETC\_FRA\_COMMON is shared between these four ETCs.

## Common component "ETC\_FRA\_Common"

### Role

This component is not a test case, but gathers all the common code that is used by the ETC FRA test cases. Its existence avoids code duplication.

### Implementation

This component is implemented in a Gradle project tree:

* The project is named "ETC\_FRA\_Common"
* Packages are retrieved from the FCTT NG (see §3.2.2.1 for the complete list)

#### FCTT\_NG reused code

The "FCTTFilesCheck" class is reused from the source code of the "FCTT\_NG" with its Java package "fr.dga.fctt\_ng.gui.configuration.controller.validation" but without the class "FCTTValidatorConfiguration".

Schematically, the "FCTTFilesCheck" class is modified:

* To use IVCT classes and log system
* To no longer use the threading capabilities of FCTT\_NG

To compile this "FCTTFilesCheck" class and its package, it is then necessary to add the following FCTT\_NG packages:

* « fr.dga.fctt\_ng.federate »
* « fr.dga.fctt\_ng.gui.configuration.controller.validation »
* « fr.dga.fctt\_ng.gui.configuration.model.validation »
* « fr.dga.fctt\_ng.gui.configuration.model.validation.parser1516e »
* « fr.dga.fctt\_ng.gui.configuration.model.validation.parser1516e.fomparser »
* « fr.dga.fctt\_ng.gui.configuration.model.validation.schematron.generated »
* « fr.dga.fctt\_ng.gui.mainWindow.model »
* « fr.dga.fctt\_ng.gui.resultData.model »
* « fr.dga.fctt\_ng.gui.resultServices.model »
* « fr.dga.fctt\_ng.gui.utils »

To be shared between the ETC FRA, these packages are respectively renamed:

* « nato.ivct.etc.fr.fctt\_common.configuration.controller.validation »
* « nato.ivct.etc.fr.fctt\_common.configuration.model.validation »
* « nato.ivct.etc.fr.fctt\_common.configuration.model.validation.parser1516e »
* « nato.ivct.etc.fr.fctt\_common.configuration.model.validation.parser1516e.fomparser »
* « nato.ivct.etc.fr.fctt\_common.configuration.model.validation.schematron.generated »
* « nato.ivct.etc.fr.fctt\_common.federate »
* « nato.ivct.etc.fr.fctt\_common.mainWindow.model »
* « nato.ivct.etc.fr.fctt\_common.resultData.model »
* « nato.ivct.etc.fr.fctt\_common.resultServices.model »
* « nato.ivct.etc.fr.fctt\_common.utils »

The Java "Class Dependency Analyzer" tool was used to identify unused classes.

### Compilation

To generate that component in Gradle environment, the "ETC\_FRA\_Common\ETC\_FRA\_Common" development subdirectory that includes the "ETC\_FRA\_Common.gradle" file. It contains all the dependencies of the module:

* For its compilation with the "compile" directives
* For its execution with "runtime" directives

The "Gradle" configuration file is provided in §5.2 of the appendix.

To start the compilation from a command prompt window launched from ETC\_FRA\_COMMON directory, run the following commands:

* gradlew eclipse (to create the Eclipse project, to be open under Eclipse)
* gradlew install (for compiling and generating .jar files)

The generation produces an "ETC\_FRA\_Common-X.Y.Z.jar" file located in the "ETC\_FRA\_Common\ETC\_FRA\_Common\build\lib" directory that will be used by the test cases.

### Configuration

Not applicable (this component is not a test case).

### Execution

Not applicable (this component is not executed).

## Test case "CS Verification"

### Role

This test case is equivalent to the "FCTT\_NG" configuration verification step. The role of this test case is to check:

* The existence of SOM and FOM files
* The validity of SOM and FOM files
* Internal and overall consistency of SOM and FOM files with respect to HLA standard

The interoperability requirements associated with this test case are as follows:

[IR-DOC-0001]: SuT interoperability capabilities shall be documented in a Conformance Statement including a SOM and a FOM with a minimum set of supporting FOM modules

[IR-SOM-0001]: SuT CS/SOM shall be valid

[IR-SOM-0002]: SuT CS/SOM shall be consistent

### Implementation

This test case is implemented in a Gradle project tree:

* The project is named "TS\_CS\_Verification"
* The source file for the test case is named "TC\_001\_Files\_Check.java"
* The source file for the test case parameter manager is named "CS\_Verification\_TcParam.java"
* The source file for the test case model is named "CS\_Verification\_BaseModel.java"

#### Test case main class

The test case is a class implementing the "IVCT\_AbstractTestCase" interface (see §2.3.1). The "tc\_cs\_verification\TC\_001\_Files\_Check.java" source file implements the "TC\_001\_Files\_Check" class and its methods:

* "getIVCT\_BaseModel" to initialize the ETC by creating:
  + an instance of the "CS\_Verification\_BaseModel" model
  + an instance of the "CS\_Verification\_TcParam" parameters
  + a "FCTTFilesCheck" business instance
* "logTestPurpose" to log the purpose of the test-case
* "preambleAction" to prepare the execution of the test case (nothing to do, there is no connection to the RTI required for this test case)
* "performTest" to sequence the test case steps:
  + Enable the "FCTTFilesCheck" instance to:
    - Load SOM and FOM files
    - Analyze the FOM files (validity)
    - Analyze SOM files (validity)
    - Analyze the inclusion of SOM(s) in the FOM(s)
    - Analyze consistency of sharing property between SOM(s) and FOM(s)
    - Check the consistency between the services used and the published and subscribed objects and interactions (cf. rules file in the appendix to §5.1)
  + Generate a report file
* "postambleAction" to complete the test case (there is no disconnection from the RTI required for this test case) and generate the results file

#### Data model management class

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | For this "CS Verification" test case, no data exchange is performed, so there is no required connection to the RTI. |

The "tc\_lib\_cs\_verification\CS\_Verification\_BaseModel.java" source file implements the "CS\_Verification\_BaseModel" class which inherits from "IVCT\_BaseModel" (see §2.3.1).

Since no connection to the RTI is required, this class only implements validation of FOM and SOM files. Interaction, object, and attribute management functions perform no action.

#### Parameters management class

The test case parameters are located in a class implementing "IVCT\_TcParam" interface (see §2.3.1). The "tc\_lib\_cs\_verification\CS\_Verification\_TcParam.java" file implements the "CS\_Verification\_TcParam" class and its methods for managing the parameters (defined in the corresponding JSON configuration file, see §2.3.2):

* "getFomFiles" for the list of federate FOM files to check
* "getSomFiles" for the list of federate SOM files to check
* "getResultDir" for the result generation folder

Example of "CS Verification"JSON parameter description file:

**{**

**"sutName" : "TestFederate"**

**"resultDirectory" : "D:\TS\_CS\_Verification\Config1"**

**"fomFiles" : [**

**{ "fileName" : "D:\TS\_CS\_Verification\Config1\fomfile1.xml" }**

**{ "fileName" : "D:\TS\_CS\_Verification\Config1\fomfile2.xml" }**

**]**

**"somFiles" : [**

**{ "fileName" : "D:\TS\_CS\_Verification\Config1\somfile1.xml" }**

**{ "fileName" : "D:\TS\_CS\_Verification\Config1\somfile2.xml" }**

**]**

**}**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | FOM and SOM files lists are described here as JSON arrays. |

#### FCTT\_NG reused code

The approach is as follows:

* The "nato.ivct.etc.fr.tc\_lib\_cs\_verification" test case package contains an implementation of the "IVCT\_TcParam" interface named "CS\_Verification\_TcParam" and a model class derived from "IVCT\_BaseModel" (see §2.3.1) named "CS\_Verification\_BaseModel"
* The class constructor "CS\_Verification\_BaseModel" creates an instance of "FCTTFilesCheck" reused from the "FCTT\_NG" source code

The class "FCTTFilesCheck" is defined in the "ETC\_FRA\_COMMON" common component.

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | In order for the source code from FCTT\_NG to run, resource files of FCTT\_NG have to be reused too. The resources of FCTT\_NG include XML and XSD files used for verification of FOM and SOM files. Some resources are situated in the project directory "src\main\resources", causing these files to be automatically integrated into the generated Java library. Other resources are situated in the project directory "bin\resources". |

### Compilation

To generate that test case in Gradle environment, the "TS\_CS\_Verification\TS\_CS\_Verification" development subdirectory that contains the "TS\_CS\_Verification.gradle" file which is a listing of all the dependencies of the module:

* For its compilation with the "compile" directives
* For its execution with "runtime" directives

The "Gradle" configuration file is provided in §5.3 of the appendix.

To start the compilation, from a command prompt window launched from the "TS\_CS\_Verification" directory, run the following commands:

* gradlew eclipse (to create the Eclipse project, then open under Eclipse)
* gradlew install (for compiling and generating .jar files)

The test case generation produces a "TS\_CS\_Verification-X.Y.Z.zip" file to be unpacked in the "TS\_CS\_Verification\TS\_CS\_Verification\build\distributions" directory to allow the execution of the "TS\_CS\_Verification-X.Y.Z.jar" file by the IVCT.

### Configuration

The test case relies on "%IVCT\_CONF%\IVCTconfig.xml" configuration file and the Test Suites files of the "TestSchedules" folder:

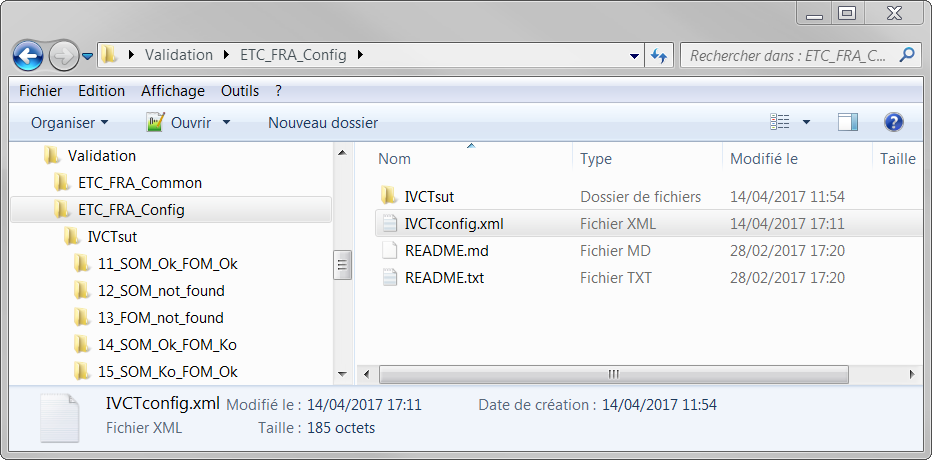


Figure 6: Configuration file tree for "TS\_CS\_Verification"

Sample contents of the « IVCTconfig.xml » file:

**<?xml version="1.0" encoding="UTF-8"?>**

**<ivctConfig>**

**<pathNames>**

**<sutDir>D:\\Users\\HLA\\Documents\\GitHub\\ETC\_FRA\_Config\\IVCTsut</sutDir>**

**</pathNames>**

**</ivctConfig>**

#### « IVCTsut » subdirectory

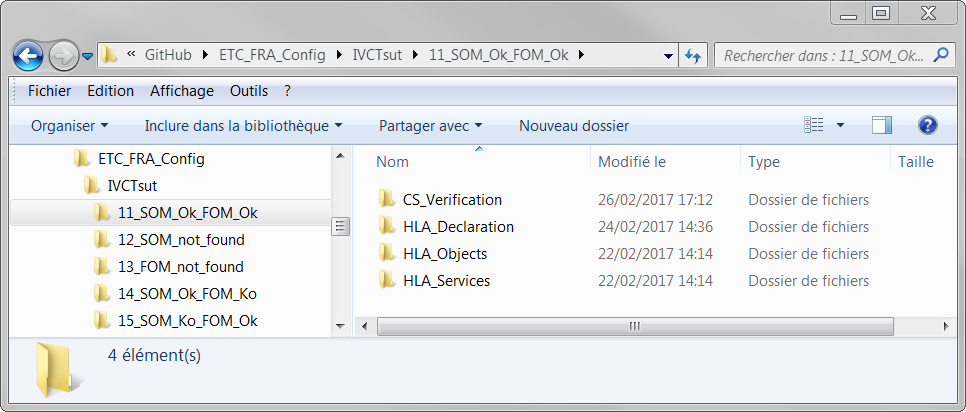


Figure 7: SuT configuration folder for different ETCs

The "IVCTsut" subdirectory contains all SuT configurations.

Each SuT subdirectory contains the "CS\_Verification" directory specific to the test case. Each "CS\_Verification" directory contains the SuT-specific configuration files (JSON parameter file and SOM and FOM files).

#### "TestSchedules" subdirectory

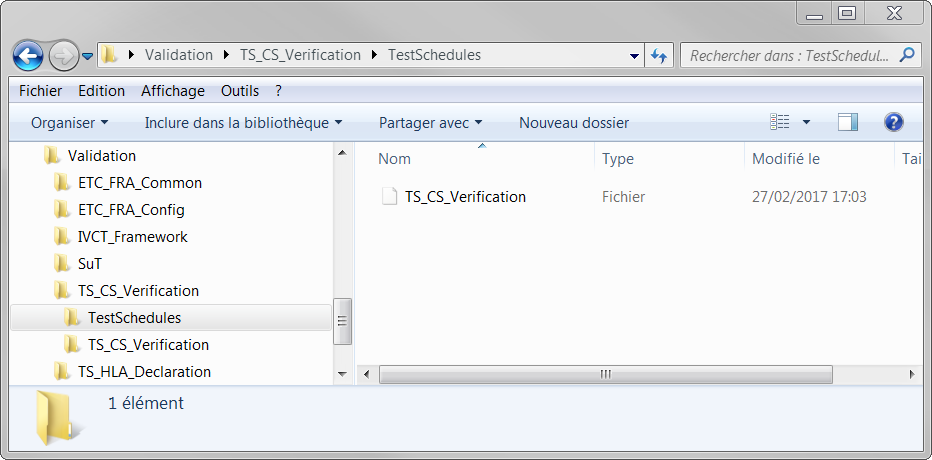


Figure 8: File trees of a Test Suite for the "CS Verification" ETC

The "TestSchedules" subdirectory contains a "TS\_CS\_Verification" file which list the test cases to be executed. Here the only test case to call is the one implemented by the "TC\_001\_Files\_Check" class (see §3.3.2.1).

### Execution

Once the test case is configured (see §3.3.4), the test case can be executed according to §2.3.3.

#### Nominal case (outside Eclipse)

An ETC FRA execution is started by launching "%IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat" script, which then allows to enter the test case execution commands (according to its configuration) in the command line tool.

For example:

**ssut 11\_SOM\_Ok\_FOM\_Ok**

**st TS\_CS\_Verification**

**sts TS\_CS\_Verification**

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | Reminder of the IVCT main commands:   * ssut (setSUT): Specifies the name of the system to be tested (matching a directory in the <sutDir> directory of the IVCTconfig.xml file) * st (setTestSuites): Indicates test suite to be executed (matching both a subdirectory contained in the directory specified by the "ssut" command and a name of a <name> tag of the IVCTtestsuites.xml file) * sts (startTestSchedule): Indicates the test schedule to be started (matching a file included in the test suite directory indicated by the "st" command) |

The execution result of the test case is logged in the "log" window, with the details of each different step performed:

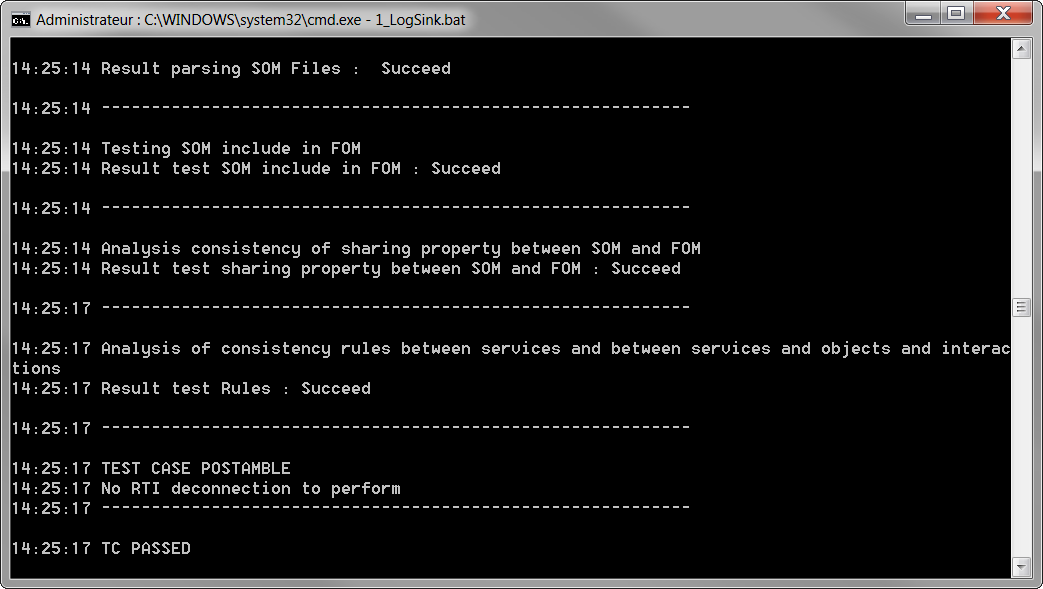


Figure 9: "Log" window

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | An example of execution log is given in appendix, §5.4. |

#### Development case (for Eclipse debugging)

Debugging of a test case under Eclipse is possible by creating a specific "Run configuration" that runs "de.fraunhofer.iosb.testrunner.JMSTestRunner" from the IVCT project named "TC".

The "Build Path" of the Eclipse project must contain the project of the test case to be executed ("TS\_CS\_Verification").

After launching the debugger on the "TC" executable, it is possible to activate the test case from the IVCT command interface: launching the "%IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat" script allows entering the execution commands of the test case according to its configuration.

For example:

**ssut 12\_SOM\_not\_found**

**st TS\_CS\_Verification**

**sts TS\_CS\_Verification**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | Any breakpoint set in the ETC "CS Verification" project sources will then stop the processing of this ETC and allow the user to analyze its step-by-step execution. |

### Results

The test case results are:

* Execution logs in the "log" window
* A ".txt" report file in a text format:
  + Created in the folder specified by the "resultDirectory" keyword of the JSON configuration file of the test case
  + Named "CS\_Verification\_report\_" followed by the execution date and time
  + With a content similar to the one provided by the FCTT NG:

-----------------------------------------------------------

Federate name :TestFederate

-----------------------------------------------------------

FOM Files :D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml

Result parsing FOM Files : Succeed

-----------------------------------------------------------

SOM Files :D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml

Result parsing SOM Files : Succeed

-----------------------------------------------------------

Result test SOM include in FOM : Succeed

-----------------------------------------------------------

Result test sharing property between SOM and FOM : Succeed

-----------------------------------------------------------

Result test Rules : Succeed

## Test Case « HLA Declaration Management »

### Role

This test case verifies that the federate only publishes and subscribes objects, interactions and attributes specified in its SOM.

The interoperability requirements associated with this test case are as follows:

[IR-SOM-0003]: SuT shall publish all object class attributes defined as published in CS/SOM

[IR-SOM-0004]: SuT shall not publish any object class attribute that is not defined as published in CS/SOM

[IR-SOM-0005]: SuT shall publish all interaction classes defined as published is CS/SOM

[IR-SOM-0006]: SuT shall not publish any interaction class that is not defined as published is CS/SOM

[IR-SOM-0007]: SuT shall subscribe to all object class attributes defined as subscribed in CS/SOM

[IR-SOM-0008]: SuT shall not subscribe to any object class attribute that is not defined as subscribed in CS/SOM

[IR-SOM-0009]: SuT shall subscribe to all interaction classes defined as subscribed in CS/SOM

[IR-SOM-0010]: SuT shall not subscribe to any interaction class that is not defined as subscribed in CS/SOM

### Implementation

This test case is implemented in a Gradle project tree:

* The project is named "TS\_HLA\_Declaration"
* The source file for the test case is named "TC\_001\_Publish\_Subscribe\_Check.java"
* The source file for the test case parameter manager is named "HLA\_Declaration\_TcParam.java"
* The source file for the test case model is named "HLA\_Declaration\_BaseModel.java"

#### Test case main class

The test case is a class implementing the "IVCT\_AbstractTestCase" interface (see §2.3.1). The "tc\_hla\_declaration\TC\_001\_Publish\_Subscribe\_Check.java" source file implements the "TC\_001\_Publish\_Subscribe\_Check" class and its methods:

* "getIVCT\_BaseModel" to initialize the ETC by creating:
  + an instance of the "HLA\_Declaration\_BaseModel" model
  + an instance of the "HLA\_Declaration\_TcParam" parameters
  + a ""FCTTFilesCheck" business instance
* "logTestPurpose" to log the purpose of the test-case
* "preambleAction" to prepare the execution of the test-case:
  + Enable the "FCTTFilesCheck" instance to:
    - Load SOM and FOM files
    - Populate a data model (class "DataHLA" reused from the FCTT NG)
  + Connect to the RTI
* "performTest" to sequence the test-case steps:
  + Wait for the time needed to stimulate the federate while updating the data model
  + Check the coherence between the objects and interactions published and subscribed during the execution and the SOM file content
  + Generate a report file
* "postambleAction" to complete the test-case:
  + Disconnect from the RTI

The "DataHLA" class is used to construct a representation of the SOM and FOM files and then update the validity (or non-validity) of the actions performed by the test federate.

#### Data model management class

The data access is done in a class derived from "IVCT\_BaseModel" (see §2.3.1). The "tc\_lib\_hla\_declaration\HLA\_Declaration\_BaseModel.java" source file implements the "HLA\_Declaration\_BaseModel" class and overrides the following methods:

* "receiveInteraction" to process interactions receipt notifications
* "discoverObjectInstance" to process object creation notifications
* "removeObjectInstance" to process notifications of object removal
* "reflectAttributeValues" to process notifications about changing attribute values

This class initializes and updates, after each message receipt, an instance of "nato.ivct.etc.fr.fctt\_common.resultData.model.DataHLA".

#### Parameters management class

The test case parameters are located in a class implementing "IVCT\_TcParam" interface (see §2.3.1). The "tc\_lib\_hla\_declaration\HLA\_Declaration\_TcParam.java" file implements the "HLA\_Declaration\_TcParam" class and its methods for managing the parameters (defined inthe corresponding JSON configuration file, see §2.3.2):

* "getFederationName" (inherited from "IVCT\_TcParam") for the HLA federation name
* "getSutName" for the SuT name
* "getRtiAddress" for the RTI IP address
* "getRtiPort" for the RTI communication port
* "getTestDuration" for the waiting time of the test case before generation of the results (sec)
* "getFomFiles" for the list of federate FOM files to check
* "getSomFiles" for the list of federate SOM files to check
* "getResultDir" for the result generation folder

Example of "HLA Declaration Management" JSON parameter description file:

**{**

**"federationName" : "FEDERATION\_TEST"**

**"sutName" : "TestFederate"**

**"rtiAddress" : "127.0.0.1"**

**"rtiPort" : "8989"**

**"testDuration" : "180"**

**"resultDirectory" : "D:\TS\_HLA\_Declaration\Config1"**

**"fomFiles" : [**

**"fileName" : "D:\TS\_HLA\_Declaration\Config1\fomfile1.xml"**

**"fileName" : "D:\TS\_HLA\_Declaration\Config1\fomfile2.xml"**

**]**

**"somFiles" : [**

**"fileName" : "D:\TS\_HLA\_Declaration\Config1\somfile1.xml"**

**"fileName" : "D:\TS\_HLA\_Declaration\Config1\somfile2.xml"**

**]**

**}**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | FOM and SOM files lists are described here as JSON arrays. |

#### FCTT\_NG reused code

The approach is as follows:

* The "nato.ivct.etc.fr.tc\_lib\_hla\_declaration" test case package contains an implementation of the "IVCT\_TcParam" interface named "HLA\_Declaration\_TcParam" and a model class derived from "IVCT\_BaseModel" (see §2.3.1) named "HLA\_Declaration\_BaseModel"
* The class constructor "HLA\_Declaration\_BaseModel" creates an instance of "FCTTFilesCheck" reused from the "FCTT\_NG" source code which produces an instance of "DataHLA" when executed (memory representation of the FOM / SOM files and storage of every call made by the SuT)

The class "FCTTFilesCheck" is defined in the "ETC\_FRA\_COMMON" common component.

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | In order for the source code from FCTT\_NG to run, resource files of the FCTT\_NG have to be reused too. The resources of FCTT\_NG include XML and XSD files used for verification of FOM and SOM files. Some resources are situated in the project directory "src\main\resources", causing these files to be automatically integrated into the generated Java library. Other resources are situated in the project directory "bin\resources". |

### Compilation

Refer to §3.3.3 in all similar respects.

### Configuration

Refer to §3.3.4 in all similar respects.

### Execution

Refer to §3.3.5 in all similar respects.

### Results

The test case results are:

* Execution logs in the "log" window
* A ".txt":report file in a text format:
  + Created in the folder specified by the "resultDirectory" keyword of the JSON configuration file of the test case
  + Named "HLA\_Declaration\_certified\_data\_" and "HLA\_Declaration\_non\_certified\_data\_" followed by the execution date and time
  + With a content similar to the one provided by the FCTT NG:

###########################################################

Certification results "TestFederate"

Date : 2017\_04\_05\_15h24m21s

Results for the data and the interactions certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

Counter and send status Counter and receive status

Objets

HLAobjectRoot

BaseEntity

AccelerationVector 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

DeadReckoningAlgorithm 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

Orientation 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

WorldLocation 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

VelocityVector 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

Interactions

HLAinteractionRoot

WeaponFire

EventIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FireControlSolutionRange 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FireMissionIndex 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FiringLocation 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FiringObjectIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FuseType 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

InitialVelocityVector 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionObjectIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionType 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

QuantityFired 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

RateOfFire 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

TargetObjectIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

WarheadType 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

## Test Case « HLA Object Management »

### Role

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | This test case is functionally equivalent to the "Data" tab of the "FCTT\_NG". |

This test case verifies that the federate sends and receives only the objects and interactions specified in its SOM.

The interoperability requirements associated with this test case are as follows:

[IR-SOM-0011]: SuT shall register at least one object instance for each published object class

[IR-SOM-0012]: SuT shall discover object instances for all object classes with attributes defined as subscribed

[IR-SOM-0013]: SuT shall update attribute values for each published object class attribute

[IR-SOM-0014]: SuT shall reflect attribute values for each subscribed object class attribute

[IR-SOM-0015]: SuT shall send at least one interaction for each published interaction class

[IR-SOM-0016]: SuT shall receive interactions for each subscribed interaction class

### Implementation

This test case is implemented in a Gradle project tree:

* The project is named "TS\_HLA\_Object"
* The source file for the test case is named "TC\_001\_Object\_Interaction\_Check.java"
* The source file for the test case parameter manager is named "HLA\_Object\_TcParam.java"
* The source file for the test case model is named "HLA\_Object\_BaseModel.java"

#### Test case main class

The test case is a class implementing the "IVCT\_AbstractTestCase" interface (see §2.3.1). The "tc\_hla\_object\TC\_001\_Object\_Interaction\_Check.java" source file implements the "TC\_001\_Object\_Interaction\_Check" class and its methods:

* "getIVCT\_BaseModel" to initialize the ETC by creating:
  + an instance of the "HLA\_Object\_BaseModel" model
  + an instance of the "HLA\_Object\_TcParam" parameters
  + a "FCTTFilesCheck" business instance
* "logTestPurpose" to log the purpose of the test-case
* "preambleAction" to prepare the execution of the test-case:
  + Enable the "FCTTFilesCheck" instance to:
    - Load SOM and FOM files
    - Populate a data model (class "DataHLA" reused from the FCTT NG)
  + Connect to the RTI
* "performTest" to sequence the test-case steps:
  + Wait for the time needed to stimulate the federate while updating the data model
  + Check the coherence between the objects attributes and interactions sent and received during the execution and the SOM file content
  + Generate a report file
* "postambleAction" to complete the test-case:
  + Disconnect from the RTI

The "DataHLA" class is used to construct a representation of the SOM and FOM files and then update the validity (or non-validity) of the actions performed by the test federate

#### Data model management class

The data access is done in a class derived from "IVCT\_BaseModel" (see §2.3.1). The "tc\_lib\_hla\_object\HLA\_Object\_BaseModel.java" source file implements the "HLA\_Object\_BaseModel" class and overrides the following methods:

* "receiveInteraction" to process interactions receipt notifications
* "discoverObjectInstance" to process object creation notifications
* "removeObjectInstance" to process notifications of object removal
* "reflectAttributeValues" to process notifications about changing attribute values

This class initializes and updates, after each message receipt, an instance of "nato.ivct.etc.fr.fctt\_common.resultData.model.DataHLA".

#### Parameters management class

The test case parameters are located in a class implementing the "IVCT\_TcParam" interface (see §2.3.1). The "tc\_lib\_hla\_object\HLA\_Object\_TcParam.java" file implements the "HLA\_Object\_TcParam" class and its methods for managing the parameters (defined inthe corresponding JSON configuration file, see §2.3.2):

* "getFederationName" (inherited from "IVCT\_TcParam") for the HLA federation name
* "getSutName" for the SuT name
* "getRtiAddress" for the RTI IP address
* "getRtiPort" for the RTI communication port
* "getTestDuration" for the waiting time of the test case before generation of the results (sec)
* "getFomFiles" for the list of federate FOM files to check
* "getSomFiles" for the list of federate SOM files to check
* "getResultDir" for the result generation folder

Refer to §3.4.2.3 for an example JSON file.

#### FCTT\_NG reused code

Refer to §3.4.2.4, classes called "HLA\_Declaration\*" becoming "HLA\_Object\*".

### Compilation

Refer to §3.3.3 in all similar respects.

### Configuration

Refer to §3.3.4 in all similar respects.

### Execution

Refer to §3.3.5 in all similar respects.

### Results

The test case results are:

* Execution logs in the "log" window
* A ".txt" report files in the text format:
  + Created in the folder specified by the "resultDirectory" keyword of the JSON configuration file of the test case
  + Named "HLA\_Object\_certified\_data\_" and "HLA\_Object\_non\_certified\_data\_" followed by the execution date and time
  + With a content similar to the one provided by the FCTT NG:

###########################################################

Certification results "TestFederate"

Date : 2017\_04\_05\_15h30m41s

Results for the data and the interactions certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

Counter and send status Counter and receive status

Objets

HLAobjectRoot

BaseEntity

AccelerationVector 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

DeadReckoningAlgorithm 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

Orientation 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

WorldLocation 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

VelocityVector 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

Interactions

HLAinteractionRoot

WeaponFire

EventIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FireControlSolutionRange 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FireMissionIndex 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FiringLocation 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FiringObjectIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FuseType 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

InitialVelocityVector 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionObjectIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionType 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

QuantityFired 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

RateOfFire 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

TargetObjectIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

## Test Case « HLA Services Verification »

### Role

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | This test case is functionally equivalent to the "Services" tab of the "FCTT\_NG". |

This test case verifies that the federate uses only the services specified in its SOM.

The interoperability requirements associated with this test case are as follows:

[IR-SOM-0019]: SuT shall implement/use all HLA services as described as implemented/used in CS/SOM

[IR-SOM-0020]: SuT shall not implement/use any HLA service that is not described as implemented/used in CS/SOM

### Implementation

This test case is implemented in a Gradle project tree:

* The project is named "TS\_HLA\_Services"
* The source file for the test case is named "TC\_001\_Services\_Check.java"
* The source file for the test case parameter manager is named "HLA\_Services\_TcParam.java"
* The source file for the test case model is named "HLA\_Services\_BaseModel.java"

#### Test case main class

The test case is a class implementing the "IVCT\_AbstractTestCase" interface (see §2.3.1). The "tc\_hla\_services\TC\_001\_Services\_Check.java" source file implements the "TC\_001\_Services\_Check" class and its methods:

* "getIVCT\_BaseModel" to initialize the ETC by creating:
  + an instance of the "HLA\_Services\_BaseModel" model
  + an instance of the "HLA\_Services\_TcParam" parameters
  + a "FCTTFilesCheck" business instance
* "logTestPurpose" to log the purpose of the test-case
* "preambleAction" to prepare the execution of the test-case:
  + Enable the "FCTTFilesCheck" instance to:
    - Load SOM and FOM files
    - Populate a data model (class "ServiceHLA" reused from the FCTT NG)
  + Connect to the RTI
* "performTest" to sequence the test-case steps:
  + Wait for the time needed to stimulate the federate while updating the data model
  + Check the coherence between the services used during the execution and the SOM file content
  + Generate a report file
* "postambleAction" to complete the test-case:
  + Disconnect from the RTI

The "ServiceHLA" class is used to construct a representation of the SOM and FOM files and then update the validity (or non-validity) of the actions performed by the test federate

#### Data model management class

The data access is done in a class derived from "IVCT\_BaseModel" (see §2.3.1). The source "tc\_lib\_hla\_services\HLA\_Services\_BaseModel.java" file implements the "HLA\_Services\_BaseModel" class and overrides the following methods:

* "receiveInteraction" to process interactions receipt notifications
* "discoverObjectInstance" to process object creation notifications
* "removeObjectInstance" to process notifications of object removal
* "reflectAttributeValues" to process notifications about changing attribute values

This class initializes and updates after each message receipt, an instance of "nato.ivct.etc.fr.fctt\_common.resultServices.model.ServiceHLA".

#### Parameters management class

The test case parameters are located in a class implementing the "IVCT\_TcParam" interface (see §2.3.1). The "tc\_lib\_hla\_object\HLA\_Services\_TcParam.java" file implements the "HLA\_Services\_TcParam" class and its methods for managing the parameters (defined inthe corresponding JSON configuration file, see §2.3.2):

* "getFederationName" (inherited from "IVCT\_TcParam") for the HLA federation name
* "getSutName" for the SuT name
* "getRtiAddress" for the RTI IP address
* "getRtiPort" for the RTI communication port
* "getTestDuration" for the waiting time of the test case before generation of the results (sec)
* "getFomFiles" for the list of federate FOM files to check
* "getSomFiles" for the list of federate SOM files to check
* "getResultDir" for the result generation folder

Refer to §3.4.2.3 for an example JSON file.

#### FCTT\_NG reused code

Refer to §3.4.2.4, classes called "HLA\_Declaration\*" becoming "HLA\_Services\*".

### Compilation

Refer to §3.3.3 in all similar respects.

### Configuration

Refer to §3.3.4 in all similar respects.

### Execution

Refer to §3.3.5 in all similar respects.

### Results

The test case results are:

* Execution logs in the "log" window
* A ".txt" report files in the text format ".txt":
  + Created in the folder specified by the "resultDirectory" keyword of the JSON configuration file of the test case
  + Named "HLA\_Services\_certified\_services\_" and "HLA\_Services\_non\_certified\_services\_" followed by the execution date and time
  + With a content similar to the one provided by the FCTT NG:

###########################################################

Certification results "TestFederate"

Date : 2017\_04\_05\_15h34m17s

Results for the services certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

Counter and services status

Services HLA 1516-2010

Federation management

Connect 2 R : ExpectedSeen (D : ExpectedNotSeen)

Disconnect 2 R : ExpectedSeen (D : ExpectedNotSeen)

Connection Lost 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Create Federation Execution 2 R : ExpectedSeen (D : ExpectedNotSeen)

Destroy Federation Execution 2 R : ExpectedSeen (D : ExpectedNotSeen)

List Federation Executions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Report Federation Executions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Join Federation Execution 2 R : ExpectedSeen (D : ExpectedNotSeen)

Resign Federation Execution 2 R : ExpectedSeen (D : ExpectedNotSeen)

Register Federation Synchronization Point 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Synchronization Point Registration 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Announce Synchronization Point 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Synchronization Point Achieved 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Synchronized 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Federation Save 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Initiate Federate Save 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federate Save Begun 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federate Save Complete 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Saved 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Abort Federation Save 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Federation Save Status 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Save Status Response 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Federation Restore 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Federation Restoration Request 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Restore Begun 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Initiate Federate Restore 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federate Restore Complete 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Restored 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Abort Federation Restore 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Federation Restore Status 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Restore Status Response 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Declaration management

Publish Object Class Attributes 2 R : ExpectedSeen (D : ExpectedNotSeen)

Unpublish Object Class Attributes 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Publish Interaction Class 2 R : ExpectedSeen (D : ExpectedNotSeen)

Unpublish Interaction Class 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Object Class Attributes 2 R : ExpectedSeen (D : ExpectedNotSeen)

Unsubscribe Object Class Attributes 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Interaction Class 2 R : ExpectedSeen (D : ExpectedNotSeen)

Unsubscribe Interaction Class 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Start Registration For Object Class 2 R : ExpectedSeen (D : ExpectedNotSeen)

Stop Registration For Object Class 1 R : ExpectedSeen (D : ExpectedNotSeen)

Turn Interactions On 2 R : ExpectedSeen (D : ExpectedNotSeen)

Turn Interactions Off 1 R : ExpectedSeen (D : ExpectedNotSeen)

Object management

Reserve Object Instance Name 2 R : ExpectedSeen (D : ExpectedNotSeen)

Object Instance Name Reserved 2 R : ExpectedSeen (D : ExpectedNotSeen)

Release Object Instance Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Reserve Multiple Object Instance Names 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Multiple Object Instance Names Reserved 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Release Multiple Object Instance Names 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Register Object Instance 2 R : ExpectedSeen (D : ExpectedNotSeen)

Discover Object Instance 2 R : ExpectedSeen (D : ExpectedNotSeen)

Update Attribute Values 30 R : ExpectedSeen (D : ExpectedNotSeen)

Reflect Attribute Values 26 R : ExpectedSeen (D : ExpectedNotSeen)

Send Interaction 6 R : ExpectedSeen (D : ExpectedNotSeen)

Receive Interaction 5 R : ExpectedSeen (D : ExpectedNotSeen)

Delete Object Instance 2 R : ExpectedSeen (D : ExpectedNotSeen)

Remove Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Local Delete Object Instance 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attributes In Scope 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attributes Out Of Scope 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Value Update 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Provide Attribute Value Update 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Turn Updates On For Object Instance 2 R : ExpectedSeen (D : ExpectedNotSeen)

Turn Updates Off For Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Request Attribute Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Attribute Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Attribute Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Report Attribute Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Interaction Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Interaction Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Interaction Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Report Interaction Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Ownership management

Unconditional Attribute Ownership Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Negotiated Attribute Ownership Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Ownership Assumption 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Divestiture Confirmation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Acquisition Notification 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Acquisition 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Acquisition If Available 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Unavailable 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Ownership Release 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Release Denied 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Divestiture If Wanted 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Cancel Negotiated Attribute Ownership Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Cancel Attribute Ownership Acquisition 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Attribute Ownership Acquisition Cancellation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Attribute Ownership 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Inform Attribute Ownership 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Is Attribute Owned By Federate 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time management

Enable Time Regulation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Regulation Enabled 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Time Regulation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Time Constrained 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Constrained Enabled 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Time Constrained 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Advance Request 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Advance Request Available 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Next Message Request 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Next Message Request Available 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Flush Queue 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Advance Grant 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Asynchronous Delivery 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Asynchronous Delivery 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query GALT 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Logical Time 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query LITS 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Modify Lookahead 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Lookahead 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Retract 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Retraction 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Change Attribute Order Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Change Interaction Order Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Data distribution management

Create Region 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Commit Region Modifications 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Delete Region 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Register Object Instance With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Associate Regions For Updates 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Unassociate Regions For Updates 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Object Class Attributes With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Unsubscribe Object Class Attributes With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Interaction Class With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Unsubscribe Interaction Class With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Send Interaction With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Value Update With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Support Services

Get Automatic Resign Directive 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Set Automatic Resign Directive 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Federate Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Federate Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Object Class Handle 2 R : ExpectedSeen (D : ExpectedNotSeen)

Get Object Class Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Known Object Class Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Object Instance Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Object Instance Name 2 R : ExpectedSeen (D : ExpectedNotSeen)

Get Attribute Handle 10 R : ExpectedSeen (D : ExpectedNotSeen)

Get Attribute Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Update Rate Value 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Update Rate Value For Attribute 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Interaction Class Handle 2 R : ExpectedSeen (D : ExpectedNotSeen)

Get Interaction Class Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Parameter Handle 26 R : ExpectedSeen (D : ExpectedNotSeen)

Get Parameter Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Order Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Order Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Transportation Type Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Transportation Type Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Available Dimensions For Class Attribute 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Available Dimensions For Interaction Class 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Upper Bound 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Handle Set 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Range Bounds 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Set Range Bounds 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Normalize Federate Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Normalize Service Group 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Object Class Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Object Class Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Attribute Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Attribute Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Attribute Scope Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Attribute Scope Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Interaction Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Interaction Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Evoke Callback 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Evoke Multiple Callbacks 46 R : ExpectedSeen (D : ExpectedNotSeen)

Enable Callbacks 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Callbacks 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

# Hardware resources

The resources required to execute the ETC FRA are the same as those of FCTT NG.

The main resource is the memory, necessary to store the large amount of data generated by the presence of an ETC FRA in the HLA federation. As a reminder, the use of the Report Service Invocation (RSI) mechanism of HLA MOM / MIM used to collect the information implies that every call made by a federate of the federation (whether SuT or not) leads to the receipt of information by an ETC FRA. A filtering is set up to store only the useful data in memory, however, all the information received must be temporarily stored.

As a result, the minimum hardware requirements for the execution of ETC FRA are detailed in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Computer type | Processor type | RAM | Disk space | System |
| Laptop | Intel i5 2,5 GHz | 4 GB minimum, 8 GB recommended | About 350 MB | Windows 7 Professional |

Table 1: Minimal hardware configuration for ETC FRA

Regarding the disk space, the space consumed by the application itself is about 250 MB. A fixed size of 100 MB is added to take account of the space required to store the files of the federate to be certified (FOM, SOM), the configuration file, the result and log files.

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The execution of ETC FRA has not been tested under Windows 8 or later. It does not guarantee its execution on these operating systems. |

# Appendices

## SCHEMATRON rules file

<schema xmlns=*"http://purl.oclc.org/dsdl/schematron"*>

<ns prefix=*"dp"* uri=*"http://standards.ieee.org/IEEE1516-2010"* />

<title>SOM validation</title>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Definition des modèles de règles pour les services\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<pattern abstract=*"true"* id=*"vrai"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed='true'"*>$p1 = <value-of select=*"$p1/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed='false' or $p2/@isUsed='true'"*>$p1 => $p2

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique2"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed='false' or $p2/@isUsed='true' or $p3/@isUsed='true'"*>$p1 => $p2 or $p3

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2/@isUsed"*/> or <value-of select=*"$p3/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique3"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed='false' or $p2/@isUsed='true' or $p3/@isUsed='true' or $p4/@isUsed='true'"*>$p1 => $p2 or $p3 or $p4

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2/@isUsed"*/> or <value-of select=*"$p3/@isUsed"*/> or <value-of select=*"$p4/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique4"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed='false' or $p2/@isUsed='true' or $p3/@isUsed='true' or $p4/@isUsed='true' or $p5/@isUsed='true'"*>$p1 => $p2 or $p3 or $p4 or $p5

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2/@isUsed"*/> or <value-of select=*"$p3/@isUsed"*/> or <value-of select=*"$p4/@isUsed"*/> or <value-of select=*"$p5/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique5"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed='false' or $p2/@isUsed='true' or $p3/@isUsed='true' or $p4/@isUsed='true' or $p5/@isUsed='true' or $p6/@isUsed='true'"*>$p1 => $p2 or $p3 or $p4 or $p5 or $p6

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2/@isUsed"*/> or <value-of select=*"$p3/@isUsed"*/> or <value-of select=*"$p4/@isUsed"*/> or <value-of select=*"$p5/@isUsed"*/> or <value-of select=*"$p6/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"egal"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"$p1/@isUsed=$p2/@isUsed"*>$p1 = $p2

<value-of select=*"$p1/@isUsed"*/> = <value-of select=*"$p2/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"egalEt1"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"($p1/@isUsed = $p2/@isUsed) and ($p3/@isUsed = $p4/@isUsed)"*>($p1=$p2) and ($p3=$p4)

<value-of select=*"$p1/@isUsed"*/> = <value-of select=*"$p2/@isUsed"*/> and <value-of select=*"$p3/@isUsed"*/> = <value-of select=*"$p4/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"egalEt2"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"($p1/@isUsed = $p2/@isUsed) and ($p3/@isUsed = $p4/@isUsed) and ($p5/@isUsed = $p6/@isUsed)"*>($p1=$p2) and ($p3=$p4) and ($p5=$p6)

<value-of select=*"$p1/@isUsed"*/> = <value-of select=*"$p2/@isUsed"*/> and <value-of select=*"$p3/@isUsed"*/> = <value-of select=*"$p4/@isUsed"*/> and <value-of select=*"$p5/@isUsed"*/> = <value-of select=*"$p6/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"reg1"*>

<rule context=*"objectModel/serviceUtilization"*>

<assert test=*"($p1/@isUsed = 'false' ) or ($p2/@isUsed = 'true' or (($p3/@isUsed = 'true' or $p4/@isUsed ='true') and ($p5/@isUsed = 'true' or $p6/@isUsed = 'true' or $p7/@isUsed = 'true' or $p8/@isUsed ='true')))"*>$p1 => $p2 or (($p3 or $p4) and ($p5 or $p6 or $p7 or $p8))

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2/@isUsed"*/> or ( (<value-of select=*"$p3/@isUsed"*/> or <value-of select=*"$p4/@isUsed"*/>) and (<value-of select=*"$p5/@isUsed"*/> or <value-of select=*"$p6/@isUsed"*/> or <value-of select=*"$p7/@isUsed"*/> or <value-of select=*"$p8/@isUsed"*/>) )

</assert>

</rule>

</pattern>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Test dépendances entre les services \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<pattern is-a=*"vrai"* id=*"vrai1"*>

<param name=*"p1"* value=*"joinFederationExecution"* />

</pattern>

<pattern is-a=*"vrai"* id=*"vrai2"*>

<param name=*"p1"* value=*"resignFederationExecution"* />

</pattern>

<pattern is-a=*"vrai"* id=*"vrai3"*>

<param name=*"p1"* value=*"connect"* />

</pattern>

<pattern is-a=*"vrai"* id=*"vrai4"*>

<param name=*"p1"* value=*"disconnect"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp1"*>

<param name=*"p1"* value=*"requestFederationSave"* />

<param name=*"p2"* value=*"initiateFederateSave"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp2"*>

<param name=*"p1"* value=*"requestFederationRestore"* />

<param name=*"p2"* value=*"federationRestoreBegun"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp3"*>

<param name=*"p1"* value=*"startRegistrationForObjectClass"* />

<param name=*"p2"* value=*"publishObjectClassAttributes"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp4"*>

<param name=*"p1"* value=*"stopRegistrationForObjectClass"* />

<param name=*"p2"* value=*"publishObjectClassAttributes"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp5"*>

<param name=*"p1"* value=*"turnInteractionsOn"* />

<param name=*"p2"* value=*"publishInteractionClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp6"*>

<param name=*"p1"* value=*"turnInteractionsOff"* />

<param name=*"p2"* value=*"publishInteractionClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp7"*>

<param name=*"p1"* value=*"registerObjectInstance"* />

<param name=*"p2"* value=*"publishObjectClassAttributes"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp8"*>

<param name=*"p1"* value=*"sendInteraction"* />

<param name=*"p2"* value=*"publishInteractionClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp9"*>

<param name=*"p1"* value=*"changeInteractionTransportationType"* />

<param name=*"p2"* value=*"publishInteractionClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp10"*>

<param name=*"p1"* value=*"requestAttributeOwnershipAssumption"* />

<param name=*"p2"* value=*"publishObjectClassAttributes"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp11"*>

<param name=*"p1"* value=*"attributeOwnershipAcquisitionNotification"* />

<param name=*"p2"* value=*"publishObjectClassAttributes"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp12"*>

<param name=*"p1"* value=*"attributeOwnershipAcquisition"* />

<param name=*"p2"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp13"*>

<param name=*"p1"* value=*"attributeOwnershipAcquisitionIfAvailable"* />

<param name=*"p2"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp14"*>

<param name=*"p1"* value=*"cancelNegotiatedAttributeOwnershipDivestiture"* />

<param name=*"p2"* value=*"negotiatedAttributeOwnershipDivestiture"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp15"*>

<param name=*"p1"* value=*"cancelAttributeOwnershipAcquisition"* />

<param name=*"p2"* value=*"attributeOwnershipAcquisition"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp16"*>

<param name=*"p1"* value=*"disableTimeRegulation"* />

<param name=*"p2"* value=*"timeRegulationEnabled"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp17"*>

<param name=*"p1"* value=*"disableTimeConstrained"* />

<param name=*"p2"* value=*"timeConstrainedEnabled"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp18"*>

<param name=*"p1"* value=*"timeAdvanceRequest"* />

<param name=*"p2"* value=*"timeAdvanceGrant"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp19"*>

<param name=*"p1"* value=*"timeAdvanceRequestAvailable"* />

<param name=*"p2"* value=*"timeAdvanceGrant"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp20"*>

<param name=*"p1"* value=*"nextMessageRequest"* />

<param name=*"p2"* value=*"timeAdvanceGrant"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp21"*>

<param name=*"p1"* value=*"nextMessageRequestAvailable"* />

<param name=*"p2"* value=*"timeAdvanceGrant"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp22"*>

<param name=*"p1"* value=*"flushQueueRequest"* />

<param name=*"p2"* value=*"timeAdvanceGrant"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp23"*>

<param name=*"p1"* value=*"disableAsynchronousDelivery"* />

<param name=*"p2"* value=*"enableAsynchronousDelivery"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp24"*>

<param name=*"p1"* value=*"modifyLookahead"* />

<param name=*"p2"* value=*"enableTimeRegulation"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp25"*>

<param name=*"p1"* value=*"queryLookahead"* />

<param name=*"p2"* value=*"enableTimeRegulation"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp26"*>

<param name=*"p1"* value=*"retract"* />

<param name=*"p2"* value=*"enableTimeRegulation"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp27"*>

<param name=*"p1"* value=*"changeInteractionOrderType"* />

<param name=*"p2"* value=*"publishInteractionClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp28"*>

<param name=*"p1"* value=*"commitRegionModifications"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp29"*>

<param name=*"p1"* value=*"deleteRegion"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp30"*>

<param name=*"p1"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p2"* value=*"publishObjectClassAttributes"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp31"*>

<param name=*"p1"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp32"*>

<param name=*"p1"* value=*"associateRegionsForUpdates"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp33"*>

<param name=*"p1"* value=*"unassociateRegionsForUpdates"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp34"*>

<param name=*"p1"* value=*"subscribeObjectClassAttributesWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp35"*>

<param name=*"p1"* value=*"unsubscribeObjectClassAttributesWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp36"*>

<param name=*"p1"* value=*"subscribeInteractionClassWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp37"*>

<param name=*"p1"* value=*"unsubscribeInteractionClassWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp38"*>

<param name=*"p1"* value=*"sendInteractionWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp39"*>

<param name=*"p1"* value=*"sendInteractionWithRegions"* />

<param name=*"p2"* value=*"publishInteractionClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp40"*>

<param name=*"p1"* value=*"requestAttributeValueUpdateWithRegions"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp41"*>

<param name=*"p1"* value=*"enableObjectClassRelevanceAdvisorySwitch"* />

<param name=*"p2"* value=*"startRegistrationForObjectClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp42"*>

<param name=*"p1"* value=*"enableObjectClassRelevanceAdvisorySwitch"* />

<param name=*"p2"* value=*"stopRegistrationForObjectClass"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp43"*>

<param name=*"p1"* value=*"enableAttributeRelevanceAdvisorySwitch"* />

<param name=*"p2"* value=*"turnUpdatesOnForObjectInstance"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp44"*>

<param name=*"p1"* value=*"enableAttributeRelevanceAdvisorySwitch"* />

<param name=*"p2"* value=*"turnUpdatesOffForObjectInstance"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp45"*>

<param name=*"p1"* value=*"enableAttributeScopeAdvisorySwitch"* />

<param name=*"p2"* value=*"attributesInScope"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp46"*>

<param name=*"p1"* value=*"enableAttributeScopeAdvisorySwitch"* />

<param name=*"p2"* value=*"attributesOutOfScope"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp47"*>

<param name=*"p1"* value=*"enableInteractionRelevanceAdvisorySwitch"* />

<param name=*"p2"* value=*"turnInteractionsOn"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp48"*>

<param name=*"p1"* value=*"enableInteractionRelevanceAdvisorySwitch"* />

<param name=*"p2"* value=*"turnInteractionsOff"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp49"*>

<param name=*"p1"* value=*"getDimensionHandleSet"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp50"*>

<param name=*"p1"* value=*"getRangeBounds"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp51"*>

<param name=*"p1"* value=*"setRangeBounds"* />

<param name=*"p2"* value=*"createRegion"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp52"*>

<param name=*"p1"* value=*"abortFederationSave"* />

<param name=*"p2"* value=*"initiateFederateSave"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp53"*>

<param name=*"p1"* value=*"abortFederationSave"* />

<param name=*"p2"* value=*"federationSaved"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp54"*>

<param name=*"p1"* value=*"abortFederationRestore"* />

<param name=*"p2"* value=*"initiateFederateRestore"* />

</pattern>

<pattern is-a=*"implique"* id=*"imp55"*>

<param name=*"p1"* value=*"abortFederationRestore"* />

<param name=*"p2"* value=*"federationRestored"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg1"*>

<param name=*"p1"* value=*"registerFederationSynchronizationPoint"* />

<param name=*"p2"* value=*"confirmSynchronizationPointRegistration"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg2"*>

<param name=*"p1"* value=*"queryFederationSaveStatus"* />

<param name=*"p2"* value=*"federationSaveStatusResponse"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg3"*>

<param name=*"p1"* value=*"queryFederationRestoreStatus"* />

<param name=*"p2"* value=*"federationRestoreStatusResponse"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg4"*>

<param name=*"p1"* value=*"startRegistrationForObjectClass"* />

<param name=*"p2"* value=*"stopRegistrationForObjectClass"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg5"*>

<param name=*"p1"* value=*"turnInteractionsOn"* />

<param name=*"p2"* value=*"turnInteractionsOff"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg6"*>

<param name=*"p1"* value=*"reserveObjectInstanceName"* />

<param name=*"p2"* value=*"objectInstanceNameReserved"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg7"*>

<param name=*"p1"* value=*"attributesInScope"* />

<param name=*"p2"* value=*"attributesOutOfScope"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg8"*>

<param name=*"p1"* value=*"turnUpdatesOnForObjectInstance"* />

<param name=*"p2"* value=*"turnUpdatesOffForObjectInstance"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg9"*>

<param name=*"p1"* value=*"attributeOwnershipAcquisitionIfAvailable"* />

<param name=*"p2"* value=*"attributeOwnershipUnavailable"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg10"*>

<param name=*"p1"* value=*"cancelAttributeOwnershipAcquisition"* />

<param name=*"p2"* value=*"confirmAttributeOwnershipAcquisitionCancellation"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg11"*>

<param name=*"p1"* value=*"queryAttributeOwnership"* />

<param name=*"p2"* value=*"informAttributeOwnership"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg12"*>

<param name=*"p1"* value=*"enableTimeRegulation"* />

<param name=*"p2"* value=*"timeRegulationEnabled"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg13"*>

<param name=*"p1"* value=*"enableTimeConstrained"* />

<param name=*"p2"* value=*"timeConstrainedEnabled"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg14"*>

<param name=*"p1"* value=*"listFederationExecutions"* />

<param name=*"p2"* value=*"reportFederationExecutions"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg15"*>

<param name=*"p1"* value=*"reserveMultipleObjectInstanceNames"* />

<param name=*"p2"* value=*"multipleObjectInstanceNamesReserved"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg16"*>

<param name=*"p1"* value=*"requestAttributeTransportationTypeChange"* />

<param name=*"p2"* value=*"confirmAttributeTransportationTypeChange"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg17"*>

<param name=*"p1"* value=*"queryAttributeTransportationType"* />

<param name=*"p2"* value=*"reportAttributeTransportationType"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg18"*>

<param name=*"p1"* value=*"requestInteractionTransportationTypeChange"* />

<param name=*"p2"* value=*"confirmInteractionTransportationTypeChange"* />

</pattern>

<pattern is-a=*"egal"* id=*"eg19"*>

<param name=*"p1"* value=*"queryInteractionTransportationType"* />

<param name=*"p2"* value=*"reportInteractionTransportationType"* />

</pattern>

<pattern is-a=*"implique2"* id=*"imp2\_1"*>

<param name=*"p1"* value=*"discoverObjectInstance"* />

<param name=*"p2"* value=*"subscribeObjectClassAttributes"* />

<param name=*"p3"* value=*"subscribeObjectClassAttributesWithRegions"* />

</pattern>

<pattern is-a=*"implique2"* id=*"imp2\_2"*>

<param name=*"p1"* value=*"receiveInteraction"* />

<param name=*"p2"* value=*"subscribeInteractionClass"* />

<param name=*"p3"* value=*"subscribeObjectClassAttributesWithRegions"* />

</pattern>

<pattern is-a=*"implique2"* id=*"imp2\_3"*>

<param name=*"p1"* value=*"requestAttributeOwnershipAssumption"* />

<param name=*"p2"* value=*"attributeOwnershipAcquisitionIfAvailable"* />

<param name=*"p3"* value=*"attributeOwnershipAcquisition"* />

</pattern>

<pattern is-a=*"implique2"* id=*"imp2\_4"*>

<param name=*"p1"* value=*"attributeOwnershipAcquisitionNotification"* />

<param name=*"p2"* value=*"attributeOwnershipAcquisition"* />

<param name=*"p3"* value=*"attributeOwnershipAcquisitionIfAvailable"* />

</pattern>

<pattern is-a="*implique2"* id=*"imp2\_5"*>

<param name=*"p1"* value=*"releaseObjectInstanceName"* />

<param name=*"p2"* value=*"objectInstanceNameReserved"* />

<param name=*"p3"* value=*"multipleObjectInstanceNamesReserved"* />

</pattern>

<pattern is-a="*implique2"* id="*imp2\_6"*>

<param name=*"p1"* value=*"releaseMultipleObjectInstanceNames"* />

<param name=*"p2"* value=*"objectInstanceNameReserved"* />

<param name=*"p3"* value=*"multipleObjectInstanceNamesReserved"* />

</pattern>

<pattern is-a=*"implique2"* id=*"imp2\_7"*>

<param name=*"p1"* value=*"enableCallbacks"* />

<param name=*"p2"* value=*"evokeCallback"* />

<param name=*"p3"* value=*"evokeMultipleCallbacks"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_1"*>

<param name=*"p1"* value=*"deleteObjectInstance"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_3"*>

<param name=*"p1"* value=*"provideAttributeValueUpdate"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_4"*>

<param name=*"p1"* value=*"turnUpdatesOnForObjectInstance"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_5"*>

<param name=*"p1"* value=*"turnUpdatesOffForObjectInstance"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_6"*>

<param name=*"p1"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_7"*>

<param name=*"p1"* value=*"negotiatedAttributeOwnershipDivestiture"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_8"*>

<param name=*"p1"* value=*"requestAttributeOwnershipRelease"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_9"*>

<param name=*"p1"* value=*"requestAttributeOwnershipRelease"* />

<param name=*"p2"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p3"* value=*"negotiatedAttributeOwnershipDivestiture"* />

<param name=*"p4"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_10"*>

<param name=*"p1"* value=*"attributeOwnershipDivestitureIfWanted"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_11"*>

<param name=*"p1"* value=*"queryAttributeOwnership"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_12"*>

<param name=*"p1"* value=*"isAttributeOwnedByFederate"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_13"*>

<param name=*"p1"* value=*"requestRetraction"* />

<param name=*"p2"* value=*"reflectAttributeValues"* />

<param name=*"p3"* value=*"receiveInteraction"* />

<param name=*"p4"* value=*"removeObjectInstance"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_14"*>

<param name=*"p1"* value=*"associateRegionsForUpdates"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_15"*>

<param name=*"p1"* value=*"unassociateRegionsForUpdates"* />

<param name=*"p2"* value=*"registerObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p4"* value=*"attributeOwnershipAcquisitionNotification"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_16"*>

<param name=*"p1"* value=*"getObjectInstanceHandle"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_17"*>

<param name=*"p1"* value=*"getObjectInstanceName"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

</pattern>

<pattern is-a=*"implique3"* id=*"imp3\_18"*>

<param name=*"p1"* value=*"getKnownObjectClassHandle"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

</pattern>

<pattern is-a=*"implique4"* id=*"imp4\_1"*>

<param name=*"p1"* value=*"retract"* />

<param name=*"p2"* value=*"updateAttributeValues"* />

<param name=*"p3"* value=*"sendInteraction"* />

<param name=*"p4"* value=*"sendInteractionWithRegions"* />

<param name=*"p5"* value=*"deleteObjectInstance"* />

</pattern>

<pattern is-a=*"implique4"* id=*"imp4\_2"*>

<param name=*"p1"* value=*"requestAttributeOwnershipRelease"* />

<param name=*"p2"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p3"* value=*"negotiatedAttributeOwnershipDivestiture"* />

<param name=*"p4"* value=*"attributeOwnershipDivestitureIfWanted"* />

<param name=*"p5"* value=*"attributeOwnershipReleaseDenied"* />

</iso:pattern>

<pattern is-a=*"implique5"* id=*"imp5\_1"*>

<param name=*"p1"* value=*"timeAdvanceGrant"* />

<param name=*"p2"* value=*"timeAdvanceRequest"* />

<param name=*"p3"* value=*"timeAdvanceRequestAvailable"* />

<param name=*"p4"* value=*"nextMessageRequest"* />

<param name=*"p5"* value=*"nextMessageRequestAvailable"* />

<param name=*"p6"* value=*"flushQueueRequest"* />

</pattern>

<pattern is-a=*"egalEt1"* id=*"egalEt1\_1"*>

<param name=*"p1"* value=*"announceSynchronizationPoint"* />

<param name=*"p2"* value=*"synchronizationPointAchieved"* />

<param name=*"p3"* value=*"synchronizationPointAchieved"* />

<param name=*"p4"* value=*"federationSynchronized"* />

</pattern>

<pattern is-a=*"egalEt1"* id=*"egalEt1\_2"*>

<param name=*"p1"* value=*"negotiatedAttributeOwnershipDivestiture"* />

<param name=*"p2"* value=*"requestDivestitureConfirmation"* />

<param name=*"p3"* value=*"requestDivestitureConfirmation"* />

<param name=*"p4"* value=*"confirmDivestiture"* />

</pattern>

<pattern is-a=*"egalEt2"* id=*"egalEt2\_1"*>

<param name=*"p1"* value=*"initiateFederateSave"* />

<param name=*"p2"* value=*"federateSaveBegun"* />

<param name=*"p3"* value=*"federateSaveBegun"* />

<param name=*"p4"* value=*"federateSaveComplete"* />

<param name=*"p5"* value=*"federateSaveComplete"* />

<param name=*"p6"* value=*"federationSaved"* />

</pattern>

<pattern is-a=*"egalEt2"* id=*"egalEt2\_2"*>

<param name=*"p1"* value=*"federationRestoreBegun"* />

<param name=*"p2"* value=*"initiateFederateRestore"* />

<param name=*"p3"* value=*"initiateFederateRestore"* />

<param name=*"p4"* value=*"federateRestoreComplete"* />

<param name=*"p5"* value=*"federateRestoreComplete"* />

<param name=*"p6"* value=*"federationRestored"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_1"*>

<param name=*"p1"* value=*"reflectAttributeValues"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_2"*>

<param name=*"p1"* value=*"removeobjectinstance"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_3"*>

<param name=*"p1"* value=*"localDeleteObjectInstance"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_4"*>

<param name=*"p1"* value=*"attributesInScope"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_5"*>

<param name=*"p1"* value=*"attributesOutOfScope"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_6"*>

<param name=*"p1"* value=*"requestAttributeOwnershipAssumption"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<pattern is-a=*"reg1"* id=*"reg1\_7"*>

<param name=*"p1"* value=*"attributeOwnershipAcquisitionNotification"* />

<param name=*"p2"* value=*"discoverObjectInstance"* />

<param name=*"p3"* value=*"registerObjectInstance"* />

<param name=*"p4"* value=*"registerObjectInstanceWithRegions"* />

<param name=*"p5"* value=*"unpublishObjectClassAttributes"* />

<param name=*"p6"* value=*"confirmDivestiture"* />

<param name=*"p7"* value=*"unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p8"* value=*"attributeOwnershipDivestitureIfWanted"* />

</pattern>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*Modeles des règles pour les tests entre les objets/interactions et les services\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<pattern abstract=*"true"* id=*"egal\_servobj"*>

<rule context=*"objectModel/objects"*>

<assert test=*"$p1/@isUsed=$p2"*>$p1 = $p2

<value-of select=*"$p1/@isUsed"*/> = <value-of select=*"$p2"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"egal\_servobj\_2"*>

<rule context=*"objectModel/objects"*>

<assert test=*"$p3 = $p1/@isUsed or $p2/@isUsed"*>$p3 = $p1 or $p2

<value-of select=*"$p3"*/> = <value-of select=*"$p1/@isUsed"*/> or <value-of select=*"$p2/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"egal\_servobj\_3"*>

<rule context=*"objectModel/objects"*>

<assert test=*"$p4 = $p1/@isUsed or $p2/@isUsed or $p3/@isUsed"*>$p4 = $p1 or $p2 or $p3

<value-of select=*"$p4"*/> = <value-of select=*"$p1/@isUsed"*/> or <value-of select=*"$p2/@isUsed"*/> or <value-of select=*"$p3/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique\_servobj"*>

<rule context=*"objectModel/objects"*>

<assert test=*"$p3 = 'false' or ($p1/@isUsed='true' or $p2/@isUsed='true')"*>$p3 => $p1 or $p2

<value-of select=*"$p3"*/> => <value-of select=*"$p1/@isUsed"*/> or <value-of select=*"$p2/@isUsed"*/>

</assert>

</rule>

</pattern>

<pattern abstract=*"true"* id=*"implique\_servobj\_1"*>

<rule context=*"objectModel/objects"*>

<assert test=*"$p1/@isUsed = 'false' or $p2='true'"*>$p1 => $p2

<value-of select=*"$p1/@isUsed"*/> => <value-of select=*"$p2"*/>

</assert>

</rule>

</pattern>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Test des règles sur les objets/interactions et les services\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<pattern is-a=*"egal\_servobj"* id=*"egal\_servobj1"*>

<param name=*"p1"* value=*"../serviceUtilization/publishObjectClassAttributes"* />

<param name=*"p2"* value=*"((count(//objectClass[sharing='PublishSubscribe']) + count(//objectClass[sharing='Publish'])) > 0)"* />

</pattern>

<pattern is-a=*"egal\_servobj"* id=*"egal\_servobj2"*>

<param name=*"p1"* value=*"../serviceUtilization/publishInteractionClass"* />

<param name=*"p2"* value=*"((count(//interactionClass[sharing='PublishSubscribe']) + count(//interactionClass[sharing='Publish']))> 0)"* />

</pattern>

<pattern is-a=*"egal\_servobj"* id=*"egal\_servobj3"*>

<param name=*"p1"* value=*"../serviceUtilization/attributeOwnershipAcquisitionNotification"* />

<param name=*"p2"* value=*"((count(//attribute[ownership='DivestAcquire']) + count(//attribute[ownership='Acquire']))> 0)"* />

</pattern>

<pattern is-a=*"egal\_servobj"* id=*"egal\_servobj4"*>

<param name=*"p1"* value=*"../serviceUtilization/updateAttributeValues"* />

<param name=*"p2"* value=*"((count(//attribute[sharing='PublishSubscribe']) + count(//attribute[sharing='Publish']))> 0)"* />

</pattern>

<pattern is-a=*"egal\_servobj"* id=*"egal\_servobj5"*>

<param name=*"p1"* value=*"../serviceUtilization/reflectAttributeValues"* />

<param name=*"p2"* value=*"((count(//attribute[sharing='PublishSubscribe']) + count(//attribute[sharing='Subscribe']))> 0)"* />

</pattern>

<pattern is-a=*"egal\_servobj\_2"* id=*"egal\_servobj6"*>

<param name=*"p1"* value=*"../serviceUtilization/subscribeObjectClassAttributes"* />

<param name=*"p2"* value=*"../serviceUtilization/subscribeObjectClassAttributesWithRegions"* />

<param name=*"p3"* value=*"((count(//objectClass[sharing='PublishSubscribe']) + count(//objectClass[sharing='Subscribe']))> 0)"* />

</pattern>

<pattern is-a=*"egal\_servobj\_3"* id=*"egal\_servobj7"*>

<param name=*"p1"* value=*"../serviceUtilization/confirmDivestiture"* />

<param name=*"p2"* value=*"../serviceUtilization/unconditionalAttributeOwnershipDivestiture"* />

<param name=*"p3"* value=*"../serviceUtilization/attributeOwnershipDivestitureIfWanted"* />

<param name=*"p4"* value=*"((count(//interactionClass[ownership='DivestAcquire']) + count(//interactionClass[ownership='Divest']))> 0)"* />

</pattern>

<pattern is-a=*"implique\_servobj"* id=*"implique\_servobj1"*>

<param name=*"p1"* value=*"../serviceUtilization/subscribeInteractionClass"* />

<param name=*"p2"* value=*"../serviceUtilization/subscribeInteractionClassWithRegions"* />

<param name=*"p3"* value=*"((count(//interactionClass[sharing='PublishSubscribe']) + count(//interactionClass[sharing='Subscribe']))> 0)"* />

</pattern>

<pattern is-a=*"implique\_servobj\_1"* id=*"implique\_servobj11"*>

<param name=*"p1"* value=*"../serviceUtilization/receiveInteraction"* />

<param name=*"p2"* value=*"((count(//interactionClass[sharing='PublishSubscribe']) + count(//interactionClass[sharing='Subscribe']))> 0)"* />

</pattern>

</schema>

## « ETC\_FRA\_COMMON.gradle » Gradle generation file

**apply plugin: 'application'**

**mainClassName = 'nato.ivct.etc.fr.fctt\_common'**

**dependencies {**

**compile libraries.slf4j\_api**

**compile fileTree(dir: 'libs', include: '\*.jar')**

**compile files('C:/Program Files/prti1516e/lib/prti1516e.jar')**

**compile files('libs/com.springsource.org.dom4j-1.6.1.jar')**

**compile files('libs/controlsfx-8.40.10.jar')**

**compile files('libs/fr.itcs.sme.architecture.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.base.api\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.base.core\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.base.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.conception.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.configuration.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.deployment.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.specification.mm\_3.0.0.201206041019.jar')**

**compile files('libs/jdom-2.0.6.jar')**

**compile files('libs/log4j-1.2-api-2.4.1.jar')**

**compile files('libs/log4j-api-2.4.1.jar')**

**compile files('libs/log4j-core-2.4.1.jar')**

**compile files('libs/org.eclipse.core.runtime-3.1.0.jar')**

**compile files('libs/org.eclipse.emf.common\_2.5.0.v200906151043.jar')**

**compile files('libs/org.eclipse.emf.ecore.xmi\_2.5.0.v200906151043.jar')**

**compile files('libs/org.eclipse.emf.ecore\_2.5.0.v200906151043.jar')**

**compile files('libs/org.eclipse.emf.edit\_2.5.0.v200906151043.jar')**

**compile files('libs/probatron.jar')**

**compile files('libs/xercesImpl.jar')**

**runtime libraries.slf4j\_jcl\_over\_slf4j**

**runtime libraries.slf4j\_jul\_to\_slf4j**

**runtime libraries.slf4j\_log4j\_over\_slf4j**

**runtime fileTree(dir: 'libs', include: '\*.jar')**

**runtime files('C:/Program Files/prti1516e/lib/prti1516e.jar')**

**runtime files('libs/com.springsource.org.dom4j-1.6.1.jar')**

**runtime files('libs/controlsfx-8.40.10.jar')**

**runtime files('libs/fr.itcs.sme.architecture.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.base.api\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.base.core\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.base.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.conception.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.configuration.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.deployment.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.specification.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/jdom-2.0.6.jar')**

**runtime files('libs/log4j-1.2-api-2.4.1.jar')**

**runtime files('libs/log4j-api-2.4.1.jar')**

**runtime files('libs/log4j-core-2.4.1.jar')**

**runtime files('libs/org.eclipse.core.runtime-3.1.0.jar')**

**runtime files('libs/org.eclipse.emf.common\_2.5.0.v200906151043.jar')**

**runtime files('libs/org.eclipse.emf.ecore.xmi\_2.5.0.v200906151043.jar')**

**runtime files('libs/org.eclipse.emf.ecore\_2.5.0.v200906151043.jar')**

**runtime files('libs/org.eclipse.emf.edit\_2.5.0.v200906151043.jar')**

**runtime files('libs/probatron.jar')**

**runtime files('libs/xercesImpl.jar')**

## « TS\_CS\_VERIFICATION.gradle » Gradle generation file

**apply plugin: 'application'**

**mainClassName = 'de.fraunhofer.iosb.testrunner.JMSTestRunner'**

**dependencies {**

**compile group: 'msg134-ivct-framework', name: 'IEEE1516e', version: ivctVersion**

**compile group: 'msg134-ivct-framework', name: 'TC', version: ivctVersion**

**compile group: 'msg134-ivct-framework', name: 'MessagingHelpers', version: ivctVersion**

**compile group: 'msg134-common', name: 'ETC\_FRA\_Common', version: ETC\_FRA\_CommonVersion**

**compile libraries.slf4j\_api**

**compile fileTree(dir: 'libs', include: '\*.jar')**

**compile files('libs/com.springsource.org.dom4j-1.6.1.jar')**

**compile files('libs/controlsfx-8.40.10.jar')**

**compile files('libs/fr.itcs.sme.architecture.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.base.api\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.base.core\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.base.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.conception.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.configuration.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.deployment.mm\_3.0.0.201206041019.jar')**

**compile files('libs/fr.itcs.sme.specification.mm\_3.0.0.201206041019.jar')**

**compile files('libs/jdom-2.0.6.jar')**

**compile files('libs/log4j-1.2-api-2.4.1.jar')**

**compile files('libs/log4j-api-2.4.1.jar')**

**compile files('libs/log4j-core-2.4.1.jar')**

**compile files('libs/org.eclipse.core.runtime-3.1.0.jar')**

**compile files('libs/org.eclipse.emf.common\_2.5.0.v200906151043.jar')**

**compile files('libs/org.eclipse.emf.ecore.xmi\_2.5.0.v200906151043.jar')**

**compile files('libs/org.eclipse.emf.ecore\_2.5.0.v200906151043.jar')**

**compile files('libs/org.eclipse.emf.edit\_2.5.0.v200906151043.jar')**

**compile files('libs/probatron.jar')**

**compile files('libs/xercesImpl.jar')**

**runtime group: 'msg134-common', name: 'ETC\_FRA\_Common', version: ETC\_FRA\_CommonVersion**

**runtime libraries.actimemqClient**

**runtime libraries.logback\_classic**

**runtime libraries.logback\_core**

**runtime libraries.slf4j\_jcl\_over\_slf4j**

**runtime libraries.slf4j\_jul\_to\_slf4j**

**runtime libraries.slf4j\_log4j\_over\_slf4j**

**runtime fileTree(dir: 'libs', include: '\*.jar')**

**runtime files('libs/com.springsource.org.dom4j-1.6.1.jar')**

**runtime files('libs/controlsfx-8.40.10.jar')**

**runtime files('libs/fr.itcs.sme.architecture.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.base.api\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.base.core\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.base.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.conception.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.configuration.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.deployment.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/fr.itcs.sme.specification.mm\_3.0.0.201206041019.jar')**

**runtime files('libs/jdom-2.0.6.jar')**

**runtime files('libs/log4j-1.2-api-2.4.1.jar')**

**runtime files('libs/log4j-api-2.4.1.jar')**

**runtime files('libs/log4j-core-2.4.1.jar')**

**runtime files('libs/org.eclipse.core.runtime-3.1.0.jar')**

**runtime files('libs/org.eclipse.emf.common\_2.5.0.v200906151043.jar')**

**runtime files('libs/org.eclipse.emf.ecore.xmi\_2.5.0.v200906151043.jar')**

**runtime files('libs/org.eclipse.emf.ecore\_2.5.0.v200906151043.jar')**

**runtime files('libs/org.eclipse.emf.edit\_2.5.0.v200906151043.jar')**

**runtime files('libs/probatron.jar')**

**runtime files('libs/xercesImpl.jar')**

**}**

## ETC CS\_Verification execution logs

### Nominal case

**14:25:13 -----------------------------------------------------------**

**Test purpose**

**Check federate FOM/SOM files existence**

**Check federate FOM/SOM files parsing**

**Check federate FOM/SOM files sharing options**

**Check federate FOM/SOM files rules conformance**

**14:25:13 TEST CASE PREAMBLE**

**14:25:13 No RTI connection to perform**

**14:25:13 -----------------------------------------------------------**

**14:25:13 TEST CASE BODY**

**14:25:13 -----------------------------------------------------------**

**14:25:13 Federate name :TestFederate**

**14:25:13 -----------------------------------------------------------**

**14:25:14 Testing FOM Files**

**14:25:14 FOM Files :**

**14:25:14 D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml**

**14:25:14**

**14:25:14 Result parsing FOM Files : Succeed**

**14:25:14 -----------------------------------------------------------**

**14:25:14 Testing SOM Files**

**14:25:14 SOM Files :**

**14:25:14 D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml**

**14:25:14**

**14:25:14 Result parsing SOM Files : Succeed**

**14:25:14 -----------------------------------------------------------**

**14:25:14 Testing SOM include in FOM**

**14:25:14 Result test SOM include in FOM : Succeed**

**14:25:14 -----------------------------------------------------------**

**14:25:14 Analysis consistency of sharing property between SOM and FOM**

**14:25:14 Result test sharing property between SOM and FOM : Succeed**

**14:25:17 -----------------------------------------------------------**

**14:25:17 Analysis of consistency rules between services and between services and objects and interac**

**tions**

**14:25:17 Result test Rules : Succeed**

**14:25:17 -----------------------------------------------------------**

**14:25:17 TEST CASE POSTAMBLE**

**14:25:17 No RTI deconnection to perform**

**14:25:17 -----------------------------------------------------------**

**14:25:17 TC PASSED**

### Failed case

The following error is that of the SOM is not included in the FOM:

**14:36:48 -----------------------------------------------------------**

**Test purpose**

**Check federate FOM/SOM files existence**

**Check federate FOM/SOM files parsing**

**Check federate FOM/SOM files sharing options**

**Check federate FOM/SOM files rules conformance**

**14:36:48 TEST CASE PREAMBLE**

**14:36:48 No RTI connection to perform**

**14:36:48 -----------------------------------------------------------**

**14:36:48 TEST CASE BODY**

**14:36:48 -----------------------------------------------------------**

**14:36:48 Federate name :TestFederate**

**14:36:48 -----------------------------------------------------------**

**14:36:48 Testing FOM Files**

**14:36:48 FOM Files :**

**14:36:48 D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\16\_SOM\_not\_in\_FOM\TS\_CS\_Verification\MAKsimple1516e.xml**

**14:36:48**

**14:36:48 Result parsing FOM Files : Succeed**

**14:36:48 -----------------------------------------------------------**

**14:36:48 Testing SOM Files**

**14:36:48 SOM Files :**

**14:36:48 D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\16\_SOM\_not\_in\_FOM\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml**

**14:36:48**

**14:36:48 Result parsing SOM Files : Succeed**

**14:36:48 -----------------------------------------------------------**

**14:36:48 Testing SOM include in FOM**

**14:36:48 Result test SOM include in FOM : Failed**

**14:36:48 SOM is not included in FOM**

**14:36:48 List of interaction not included in FOM:**

**14:36:48 WeaponFire2**

**14:36:48**

**14:36:48 -----------------------------------------------------------**

**14:36:48 Analysis consistency of sharing property between SOM and FOM**

**14:36:48 Result test sharing property between SOM and FOM : Succeed**

**14:36:51 -----------------------------------------------------------**

**14:36:51 Analysis of consistency rules between services and between services and objects and interac**

**tions**

**14:36:51 Result test Rules : Succeed**

**14:36:51 TC FAILED Invalid FOM/SOM files**

1. A Test Suite is a set of Test Cases. For French Test Cases, a Test Suite includes only one Test Case. [↑](#footnote-ref-1)
2. Restricted access, requires a user account [↑](#footnote-ref-2)