

*Heterogeneous Missions Accessibility - Testbed  
Sensor Planning Systems Interface*

***HMA-T SPS I/F***  
**SYSTEM TEST SPECIFICATION**

**Code :** HMAT-DMS-STS-001  
**Issue :** 1.0  
**Date :** 04/02/2009

	<b>Name</b>	<b>Function</b>	<b>Signature</b>
<b>Prepared by</b>	Reuben Wright (DMS) Ricardo Moyano (DMS) Philippe Mériqot (SI) Manuel Casado (DMS)	Project Engineer Project Manager Project Engineer Project Engineer	
<b>Reviewed by</b>	Ricardo Moyano (DMS) Alexandre Robin (SI)	Review Team	
<b>Approved by</b>	Ricardo Moyano (DMS)	Project Manager	
<b>Signatures and approvals on original</b>			

DEIMOS Space S.L.  
Ronda de Poniente, 19, Edificio Fiteni VI, 2-2ª  
28760 Tres Cantos (Madrid), SPAIN  
Tel.: +34 91 806 34 50 / Fax: +34 91 806 34 51  
E-mail: deimos@deimos-space.com

This page intentionally left blank

## Document Information

Contract Data	
Contract Number:	780-2/C09 "Heterogeneous Missions Accessibility - Testbed. Phase 2"
Contract Issuer:	SPACEBEL SA

Internal Distribution		
Name	Unit	Copies
Ricardo Moyano	DMS	1
Reuben Wright	DMS	1
Internal Confidentiality Level (DMS-COV-POL05)		
Unclassified <input type="checkbox"/>	Restricted <input checked="" type="checkbox"/>	Confidential <input type="checkbox"/>

External Distribution		
Name	Organisation	Copies
Yves Coene	SPACEBEL	1
Pier Giorgio Marchetti	ESA/ESRIN	1

Archiving	
Word Processor:	MS Word 2000
File Name:	HMA-T-DMS-STS-001-10-R.doc

## Document Status Log

Issue	Change description	Date	Approved
1.0	First version of the document	04/02/2009	

## Table of Contents

<b>1. INTRODUCTION</b>	<b>11</b>
<b>1.1. Purpose and Scope of the Document</b>	<b>11</b>
<b>1.2. Document structure</b>	<b>11</b>
<b>1.3. Evolution of the Plan</b>	<b>11</b>
<b>1.4. Acronyms and Abbreviations</b>	<b>11</b>
<b>1.5. Definitions</b>	<b>12</b>
<b>2. RELATED DOCUMENTS</b>	<b>13</b>
<b>2.1. Applicable Documents</b>	<b>13</b>
<b>2.2. Reference Documents</b>	<b>13</b>
<b>3. TEST SPECIFICATION OVERVIEW</b>	<b>14</b>
<b>3.1. Validation Approach</b>	<b>14</b>
<b>3.2. Test Case Specifications</b>	<b>14</b>
<b>3.3. Test Procedure Specifications</b>	<b>15</b>
<b>3.4. Test Environment</b>	<b>15</b>
<b>3.5. System Test Facilities</b>	<b>15</b>
<b>3.6. Test Organisation</b>	<b>16</b>
<b>4. SYSTEM TEST CASES</b>	<b>17</b>
<b>4.1. DEIMOS SPS Test Cases</b>	<b>17</b>
4.1.1. Test Case TC-0010: Installation	17
4.1.1.1. Purpose	17
4.1.1.2. Software Requirements Verified	17
4.1.1.3. Resources and Tools	17
4.1.1.4. Input Specification	17
4.1.1.5. Output Specification	17
4.1.1.6. Reference data	17
4.1.1.7. Pass/Fail Criteria	17
4.1.2. Test Case TC-0020: GetCapabilities response	18
4.1.2.1. Purpose	18
4.1.2.2. Software Requirements Verified	18
4.1.2.3. Resources and Tools	18
4.1.2.4. Input Specification	18
4.1.2.5. Output Specification	18

4.1.2.6. Reference data _____	18
4.1.2.7. Pass/Fail Criteria _____	19
4.1.3. Test Case TC-0030: DescribeTasking correct message response _____	20
4.1.3.1. Purpose _____	20
4.1.3.2. Software Requirements Verified _____	20
4.1.3.3. Resources and Tools _____	20
4.1.3.4. Input Specification _____	20
4.1.3.5. Output Specification _____	20
4.1.3.6. Reference data _____	20
4.1.3.7. Pass/Fail Criteria _____	20
4.1.4. Test Case TC-0040: DescribeTasking incorrect message response _____	20
4.1.4.1. Purpose _____	20
4.1.4.2. Software Requirements Verified _____	20
4.1.4.3. Resources and Tools _____	21
4.1.4.4. Input Specification _____	21
4.1.4.5. Output Specification _____	21
4.1.4.6. Reference data _____	21
4.1.4.7. Pass/Fail Criteria _____	21
4.1.5. Test Case TC-0050: GetFeasibility feasible request _____	21
4.1.5.1. Purpose _____	21
4.1.5.2. Software Requirements Verified _____	21
4.1.5.3. Resources and Tools _____	21
4.1.5.4. Input Specification _____	21
4.1.5.5. Output Specification _____	23
4.1.5.6. Reference data _____	23
4.1.5.7. Pass/Fail Criteria _____	25
4.1.6. Test Case TC-0060: GetFeasibility not feasible request _____	26
4.1.6.1. Purpose _____	26
4.1.6.2. Software Requirements Verified _____	26
4.1.6.3. Resources and Tools _____	26
4.1.6.4. Input Specification _____	26
4.1.6.5. Output Specification _____	27
4.1.6.6. Reference data _____	27
4.1.6.7. Pass/Fail Criteria _____	28
4.1.7. Test Case TC-0070: Submit correct response _____	28

4.1.7.1. Purpose	28
4.1.7.2. Software Requirements Verified	28
4.1.7.3. Resources and Tools	28
4.1.7.4. Input Specification	28
4.1.7.5. Output Specification	29
4.1.7.6. Reference data	29
4.1.7.7. Pass/Fail Criteria	31
4.1.8. Test Case TC-0080: GetStatus Known response	31
4.1.8.1. Purpose	31
4.1.8.2. Software Requirements Verified	32
4.1.8.3. Resources and Tools	32
4.1.8.4. Input Specification	32
4.1.8.5. Output Specification	32
4.1.8.6. Reference data	32
4.1.8.7. Pass/Fail Criteria	33
4.1.9. Test Case TC-0090: GetStatus Unknown response	33
4.1.9.1. Purpose	33
4.1.9.2. Software Requirements Verified	33
4.1.9.3. Resources and Tools	33
4.1.9.4. Input Specification	33
4.1.9.5. Output Specification	33
4.1.9.6. Reference data	34
4.1.9.7. Pass/Fail Criteria	34
4.1.10. Test Case TC-0100: DescribeResultAccess correct response	34
4.1.10.1. Purpose	34
4.1.10.2. Software Requirements Verified	34
4.1.10.3. Resources and Tools	34
4.1.10.4. Input Specification	34
4.1.10.5. Output Specification	35
4.1.10.6. Reference data	35
4.1.10.7. Pass/Fail Criteria	35
4.1.11. Test Case TC-0110: DescribeResultAccess unavailable data response	35
4.1.11.1. Purpose	35
4.1.11.2. Software Requirements Verified	35
4.1.11.3. Resource and Tools	35

4.1.11.4. Input Specification _____	35
4.1.11.5. Output Specification _____	36
4.1.11.6. Reference data _____	36
4.1.12. Test Case TC-0120: Performance on a Linux platform _____	36
4.1.12.1. Purpose _____	36
4.1.12.2. Software Requirements Verified _____	36
4.1.12.3. Resources and Tools _____	36
4.1.12.4. Input Specification _____	36
4.1.12.5. Output Specification _____	36
4.1.12.6. Pass/Fail Criteria _____	37
4.1.13. Test Case TC-0130: Performance on a Windows Server platform _____	37
4.1.13.1. Purpose _____	37
4.1.13.2. Software Requirements Verified _____	37
4.1.13.3. Resources and Tools _____	37
4.1.13.4. Input Specification _____	37
4.1.13.5. Output Specification _____	37
4.1.13.6. Pass/Fail Criteria _____	37
<b>4.2. SPOT SPS I/F Specific Test Cases _____</b>	<b>38</b>
4.2.1. Test Case TC-1010: SPS I/F functionality without notification _____	38
4.2.1.1. Purpose _____	38
4.2.1.2. Software Requirements Verified _____	38
4.2.1.3. Resources and Tools _____	38
4.2.1.4. Input Specification _____	38
4.2.1.5. Output Specification _____	38
4.2.1.6. Pass/Fail Criteria _____	39
4.2.2. Test Case TC-1020: SPS I/F functionality with notification _____	39
4.2.2.1. Purpose _____	39
4.2.2.2. Software Requirements Verified _____	39
4.2.2.3. Resources and Tools _____	39
4.2.2.4. Input Specification _____	39
4.2.2.5. Output Specification _____	39
4.2.2.6. Pass/Fail Criteria _____	39
<b>5. SYSTEM TEST PROCEDURES _____</b>	<b>40</b>
<b>5.1. DEIMOS SPS Test Procedures _____</b>	<b>40</b>
5.1.1. Test Procedure TP-0010: Installation _____	40



5.1.1.1. Objective	40
5.1.1.2. Pre-conditions	40
5.1.1.3. Procedure Steps	40
5.1.2. Test Procedure TP-0020: GetCapabilities response	41
5.1.2.1. Objective	41
5.1.2.2. Pre-conditions	41
5.1.2.3. Procedure Steps	41
5.1.3. Test Procedure TP-0030: DescribeTasking responses	42
5.1.3.1. Objective	42
5.1.3.2. Pre-conditions	42
5.1.3.3. Procedure Steps	42
5.1.4. Test Procedure TP-0040: GetFeasibility response	42
5.1.4.1. Objective	42
5.1.4.2. Pre-conditions	43
5.1.4.3. Procedure Steps	43
5.1.5. Test Procedure TP-0050: Submit / GetStatus response	43
5.1.5.1. Objective	43
5.1.5.2. Pre-conditions	44
5.1.5.3. Procedure Steps	44
5.1.6. Test Procedure TP-0060: DescribeResultAccess response	45
5.1.6.1. Objective	45
5.1.6.2. Pre-conditions	45
5.1.6.3. Procedure Steps	45
<b>5.2. SPOT SPS I/F Test Procedures</b>	<b>46</b>
5.2.1. Test Procedure TP-1010: SPS I/F functionality without notification	46
5.2.1.1. Objective	46
5.2.1.2. Pre-conditions	46
5.2.1.3. Procedure Steps	46
5.2.2. Test Procedure TP-1020: SPS I/F functionality with notification	48
5.2.2.1. Objective	48
5.2.2.2. Pre-conditions	48
5.2.2.3. Procedure Steps	48

## List of Tables

Table 1: Applicable documents .....	13
Table 2: Reference documents .....	13

## 1. INTRODUCTION

### 1.1. Purpose and Scope of the Document

This is the Software Test Specification (STS) for the HMA-T project related to the implementation of the SPS Profile for Earth Observation interface [OGC 07-018] applied to two different planning systems, developed by the two partners in the consortium and will be for the following planning systems:

1. SPOT IMAGE EO planning system
2. Earth Explorer Mission Software CFI

The plan presented here is intended to outline the testing to be done in support of the demonstration activities which constitute the part of the final presentation of the project. These interfaces are not intended to be operational systems.

The project is focussed on the mandatory operations of the application profile which are related to sending programming requests for EO products to support access to data from heterogeneous systems dealing with derived data products from satellite based measurements of the earth's surface and environment.

### 1.2. Document structure

Information provided in this document includes:

- Introduction to the document itself (the present section).
- List of reference and applicable documents (section 2).
- The description of how the Test Cases and Procedures are arranged (section 3).
- Test Case specifications, including for each test case, the objective, the list of software requirements covered and the test case and procedure (section 4).
- Test Procedure specifications, including for each test procedure, the objective, the test cases it covers and the list of steps that is required to accomplish the procedure (section 5).

### 1.3. Evolution of the Plan

This version of the document presents the preliminary plans following the analysis of software requirements and system architecture. It is to be updated only if necessary due to changed circumstances that require alterations to the approaches presented herein.

### 1.4. Acronyms and Abbreviations

The acronyms and abbreviations used in this document are the following ones:

Acronym	Description
ADD	Architectural Design Document

Acronym	Description
API	Application Programming Interface
COTS	Commercial Off-The-Shelf
DMS	DEIMOS Space
HW	Hardware
I/F	Interface
O/S	Operating System
PM	Progress Meeting Project Manager
PMP	Project Management Plan
RD	Reference Document
SOW	Statement Of Work
SRD	Software Requirements Document
ST	System Test
STS	Software Test Specification
SVTR	Software Validation Test Report
SW	Software
TBC	To Be Confirmed
TBD	To Be Defined / Decided
TC	Test Case
TER	Test Execution Record
TP	Test Procedure

## 1.5. Definitions

The definitions of the specific terms used in this document are the following ones:

- Acceptance testing:** Formal testing conducted to determine whether or not a system satisfies the acceptance criteria, previously defined by the customer.
- Validation:** Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled [ISO 9000:2000]

NOTE: The validation process (for software) is the process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product.

## 2. RELATED DOCUMENTS

### 2.1. Applicable Documents

The following table specifies the applicable documents that shall be complied with during project development.

*Table 1: Applicable documents*

Reference	Code	Title	Issue
[OGC 07-018]	OGC 07-018	OpenGIS Sensor Planning Service Application Profile for EO Sensors	2.0 draft 04/02/09
[OGC 09-000]	OGC 09-000	OpenGIS Sensor Planning Service Specification	2.0 06/01/09
[SOW]	SPB-HMA-T-SOW-002	Statement of Work For HMA Testbed (HMA-T) Phase 2	1.0
[SRD]	HMAT-DMS-SRD-001	HMA-T Sensor Planning Systems Interface Software Requirements Document	1.0

### 2.2. Reference Documents

The following table specifies the reference documents that shall be taken into account during project development.

*Table 2: Reference documents*

Reference	Code	Title	Issue
[PMP]	HMAT-SPMP-0002-SPB	Software Project Management Plan	1.3
[WS Addressing]	WS-Addressing	Web Services Addressing 1.0 - Core	1.0
[WS Notifications]	WS-BaseNotification	Web Services Base Notification OASIS Standard, 1 October 2006	1.3
[OGC 05-008]	OGC 05-008	OpenGIS Web Services Common Specification	1.0

## 3. TEST SPECIFICATION OVERVIEW

### 3.1. Validation Approach

Tests are designed to verify features or sets of features of the system. One feature may correspond to only one requirement or to a group of requirements.

The features that are to be tested correspond to the system requirements that are to be verified by test, according to the validation method of the requirements as identified in [SRD].

There will be two developed systems (DEIMOS and SPOT) and testing will be run for each system independently. The SPS Application profile [OGC 07-018] allows for two models of GetFeasibility processing – synchronous and asynchronous. The two developed systems implement these two different models, and so the tests related to the asynchronous method using a Notification Server are only to be carried out on one system. Otherwise all tests are carried out on both systems.

Unit and integration testing will be carried out during development as required, and formal System tests will be specified below in section 4 and carried out before formal presentations. The outputs will be recorded as Test Execution Records and made available for review.

As stated in the Software Requirements Document [SRD] various requirements are validated via Inspection or Review, rather than Testing. These are not covered in this document, and there does not need to be (for this project) a formal description of the process for their validation.

### 3.2. Test Case Specifications

Test cases must specify the details of the test approach for the set of the system requirements to be validated. Test cases shall be defined containing the following information:

- ❑ **Test case identifier:** Test cases shall be identified according the following convention:

TC-*nnnn*: <*test case name*>

where *nnnn* is a sequence number using four digits (e.g. 0010), and <*test case name*> is the name associated to the test case. The test case name will be unique for each test case, and will be made up of a few descriptive words.

- ❑ **Purpose:** Objectives of the test case in terms of features that are to be tested.
- ❑ **System requirement coverage:** Set of requirements that specify the functionality being tested. Note that the test cases often entail the execution of other functionalities besides the one that is the subject of the test, but the requirements associated with those other functionalities are not listed (i.e. they are not part of the aims of that particular test case). This is done deliberately to make it easier to define the pass/fail criteria for each test.
- ❑ **Resources and tools.** Needed resources and tools needed by the use cases. For example, they may be scripts or utilities to verify the test results.
- ❑ **Inputs:** Input data needed for the test case. Specifies the data sets and the values of the data sets representing the inputs to the test case.
- ❑ **Outputs:** Expected output data produced as a result of the test case execution.

- ❑ **Pass/Fail criteria:** Describes the criteria to be used to determine whether the test passes or fails, and consequently determines if the involved requirements are fulfilled or not.

### 3.3. Test Procedure Specifications

Test procedures describe the sequence of actions the operator or test engineer should carry out in order to properly execute the test.

A test procedure is constituted by:

- ❑ **Test procedure identifier:** Test procedures shall be identified according the following convention:

TP-*nnnn*: <*test procedure name*>

where *nnnn* is a sequence number using four digits (e.g. 0010), and <*test procedure name*> is the name associated to the test case. The test procedure name will be unique, and will be made up of a few descriptive words.

- ❑ **Objective:** Objectives of the test procedure, generally in terms of the test cases covered by the procedure. As mentioned earlier, a test procedure may cover one or more test cases, depending on the design of the procedure.
- ❑ **Pre-conditions:** List of conditions that must be met to ensure the correct behaviour of the test procedure.
- ❑ **Procedure steps:** Identifies all steps of the procedure and the corresponding expected results. Those steps and activities that are repeatedly performed for many tests can be placed in a separate test procedure which is invoked by all affected procedures.

### 3.4. Test Environment

This section identifies the elements that make up the environment needed to run the system tests defined in this document.

This environment is composed of the elements necessary to execute the tests:

1. The execution environment, which comprises the elements necessary to run system tests, i.e. hardware, operating system, software tools, etc.
2. The validation methods, which provide the means to validate the results of the tests.
3. The test data.

### 3.5. System Test Facilities

Spot Image SPS will be deployed in Spot Image facilities. TBD whether the DEIMOS SPS will be deployed on servers within DEIMOS facilities or at ESRIN. In any case they will both be accessible through the Internet by an operator using the SSE if a compatible client exists within the SSE. In the case of the DEIMOS SPS the testing will be specified through the use of the Eclipse IDE Web Services Explorer which allows detailed specification of input and examination of output messages.

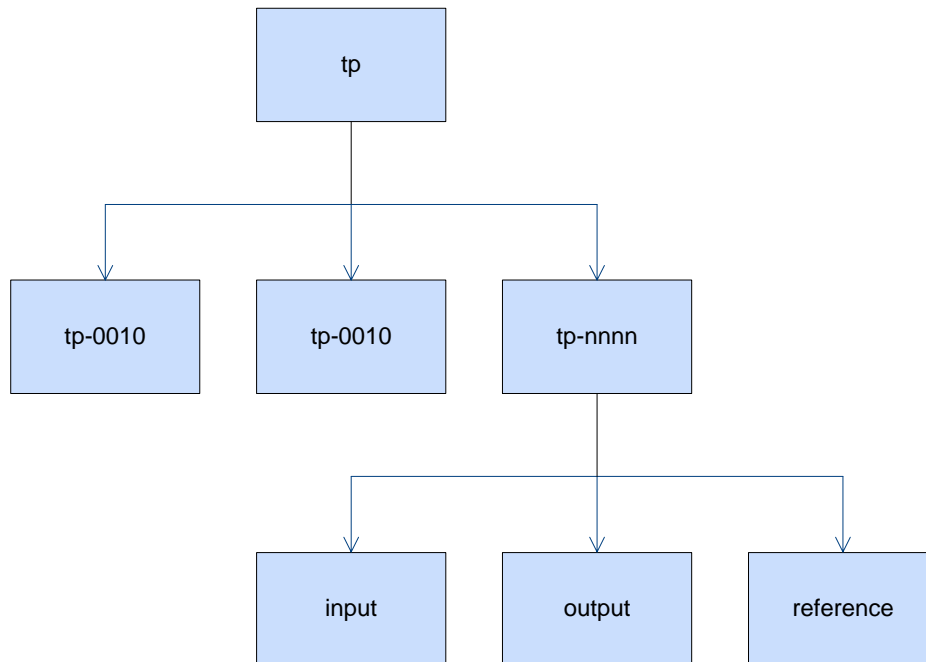
### 3.6. Test Organisation

Test procedures shall be organised under a dedicated directory named `tp`. An environment variable named `TEST_HOME` shall point to this directory. A directory per test procedure shall exist with all the necessary elements needed for its execution. Thus, it is possible to conduct the execution of the different test procedures and to control and manage the outputs that are obtained as a result of each run.

The directory name for each system test shall be named as `tp-nnnn`, being `nnnn` the number of the test procedure (e.g. 0020). Under this directory, data are arranged in a set of sub-directories:

- **input/** including the set of files that will represent the input data for the associated test procedure.
- **output/** where the tests output, i.e., information logged as a result of the execution of the test is placed.
- **reference/** where reference data is kept for the verification activities of the test procedure.

The figure below summarises the directory structure applicable for the HMA-T test development.



*Figure 3-1: Test procedures directory structure*



## 4. SYSTEM TEST CASES

These system test cases will verify that the system requirements have been implemented in the software. They will generally involve the sending of specific test SOAP messages to the SPS I/F and monitoring the responses. The set of messages to use is to be determined alongside the development of a scenario for the demonstration of the I/F.

The section has been organised according to the different kinds of test cases:

- DEIMOS STS specific test cases – test cases applicable only to DEIMOS SPS. The identifier of these test cases shall start with the “TC-0” label.
- SPOT STS specific test cases – test cases applicable only to SPOT SPS. The identifier of these test cases shall start with the “TC-1” label.

### 4.1. DEIMOS SPS Test Cases

#### 4.1.1. Test Case TC-0010: Installation

##### 4.1.1.1. Purpose

This test case checks that SPS interface system that accesses to DEIMOS SPS is correctly installed.

##### 4.1.1.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-INS-0010

##### 4.1.1.3. Resources and Tools

- Tools to uncompress the delivered package
- Compilers / tools to generate executables

##### 4.1.1.4. Input Specification

Not applicable.

##### 4.1.1.5. Output Specification

- The SPS interface is correctly installed in the target platform.

##### 4.1.1.6. Reference data

Not applicable.

##### 4.1.1.7. Pass/Fail Criteria

- DEIMOS SPS is correctly installed and is able to be executed

## 4.1.2. Test Case TC-0020: GetCapabilities response

### 4.1.2.1. Purpose

This test case checks that the response to a GetCapabilities request is correctly given.

### 4.1.2.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

### 4.1.2.3. Resources and Tools

- Client simulator able to connect to SPS I/F using HTTP GET and display XML responses

### 4.1.2.4. Input Specification

- `http://hostname:port/HMA-SPS/GetCapabilitiesServlet?service=SPS&request=GetCapabilities` where "hostname:port" the Internet Protocol hostname or numeric address and the port number used for the interaction.

### 4.1.2.5. Output Specification

- Synchronous returned XML message corresponding to the GetCapabilities response as specified in section 10.5 of [OGC 07-018].

### 4.1.2.6. Reference data

- GetCapabilitiesResponse.xml file for DEIMOS SPS. The information contained in the file is static and therefore can be known in advance. The only part that will change is that within the `xlink:href` attributes the value "localhost:8080" will have been replaced by the administrator with the IP address and port for their server. In keeping with the scope of this project the OperationsMetadata will list: DescribeTasking, GetFeasibility, Submit, DescribeResultAccess and GetStatus as being supported, but not DescribeSensor, Cancel, and Update. The expected document is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/sps/2.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0.0">
<ows:ServiceIdentification>
  <ows:Title>HMA-T Sensor Planning Service I/F</ows:Title>
  <ows:Abstract>HMA-T sps developed by Deimos-Space</ows:Abstract>
  <ows:ServiceType>SPS</ows:ServiceType>
  <ows:ServiceTypeVersion>2.0</ows:ServiceTypeVersion>
  <ows:Fees>none</ows:Fees>

  <ows:AccessConstraints>none</ows:AccessConstraints>
</ows:ServiceIdentification>
<ows:OperationMetadata>
  <ows:Operation name="DescribeTasking">
    <ows:DCP>
      <ows:HTTP>
```

```

        <ows:Get          xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
        <ows:Post       xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
        </ows:HTTP>

    </ows:DCP>
</ows:Operation>
<ows:Operation name="GetFeasibility">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get          xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            <ows:Post       xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            </ows:HTTP>
        </ows:DCP>

    </ows:Operation>
<ows:Operation name="Submit">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get          xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            <ows:Post       xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            </ows:HTTP>
        </ows:DCP>
    </ows:Operation>

<ows:Operation name="GetStatus">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get          xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            <ows:Post       xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            </ows:HTTP>
        </ows:DCP>
    </ows:Operation>
<ows:Operation name="DescribeResultAccess">

    <ows:DCP>
        <ows:HTTP>
            <ows:Get          xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            <ows:Post       xlink:href="http://localhost:8080/HMA-T/services/HMA-SPS"
xlink:type="simple"/>
            </ows:HTTP>
        </ows:DCP>
    </ows:Operation>
</ows:OperationMetadata>
<Capabilities>

```

#### 4.1.2.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].

- The returned message is equal to the reference file corresponding to the requested SPS (DEIMOS or SPOT).

### ***4.1.3. Test Case TC-0030: DescribeTasking correct message response***

#### **4.1.3.1. Purpose**

This test case checks that the response to a correct DescribeTasking request is correctly given.

#### **4.1.3.2. Software Requirements Verified**

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### **4.1.3.3. Resources and Tools**

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### **4.1.3.4. Input Specification**

- CorrectDescribeTasking.txt:** DescribeTasking request (SOAP message) with the following contents:

```
<sps:DescribeTasking                      service="SPS"                      version="2.0.0"  
xmlns:sps="http://www.opengis.net/sps/2.0">  
  <sps:sensorID>urn:esa:int:Sentinel-1:C-SAR</sps:sensorID>  
</sps:DescribeTasking>
```

#### **4.1.3.5. Output Specification**

- Synchronous returned XML file compliant to the DescribeTasking response.

#### **4.1.3.6. Reference data**

- ReferenceDescribeTaskingResponse.xml: expected XML file corresponding to the response to the DescribeTasking request.

#### **4.1.3.7. Pass/Fail Criteria**

- The returned message conforms to [OGC 07-018].
- The contents of the returned XML file matches the reference file from section 4.1.3.6.

### ***4.1.4. Test Case TC-0040: DescribeTasking incorrect message response***

#### **4.1.4.1. Purpose**

This test case checks that the response to an incorrect DescribeTasking request is an error message indicating that anomalies were detected in the input request.

#### **4.1.4.2. Software Requirements Verified**

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### 4.1.4.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.4.4. Input Specification

- IncorrectDescribeTasking.txt:** DescribeTasking request (SOAP message) which is the same as for the previous case but two mandatory fields (service and version) are omitted:

```
<sps:DescribeTasking xmlns:sps="http://www.opengis.net/sps/2.0">  
  <sps:sensorID>urn:esa:int:Sentinel-1:C-SAR</sps:sensorID>  
</sps:DescribeTasking>
```

#### 4.1.4.5. Output Specification

- Response indicating that there is a SOAP fault. On the SPS server there will be an error message indicating that values of the “SERVICES” and “VERSION” parameters are missing.

#### 4.1.4.6. Reference data

Not applicable

#### 4.1.4.7. Pass/Fail Criteria

- There is a SOAP fault message returned. Also an error is logged indicating that the values of the “SERVICES” and “VERSION” parameters are missing (only to be checked if have access to server).

### 4.1.5. *Test Case TC-0050: GetFeasibility feasible request*

#### 4.1.5.1. Purpose

This test case checks that the response to a feasible GetFeasibility request to DEIMOS SPS is correctly given.

#### 4.1.5.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010

SR-FUN-0020

SR-FUN-0030

SR-FUN-0210

#### 4.1.5.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.5.4. Input Specification

- GetFeasibility.txt:** GetFeasibility request (SOAP message) with the following contents:

```
<sps:GetFeasibility service="SPS" version="2.0.0"  
xsi:schemaLocation="http://www.opengis.net/sps/2.0  
http://schemas.opengis.net/sps/2.0.0/sps.xsd"  
xmlns:eop="http://www.esa.int/eop" xmlns:sps="http://www.opengis.net/sps/2.0"  
xmlns:swe="http://www.opengis.net/swe/2.0"  
xmlns:gml="http://www.opengis.net/gml/3.2"  
xmlns:xlink="http://www.w3.org/1999/xlink"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```

<sps:sensorID>urn:ESA:sensors:Sentinel-1:C-SAR</sps:sensorID>
<sps:taskingParameters>
  <sps:ParameterData>
    <sps:encoding>
      <swe:XMLEncoding namespace="http://www.esa.int/eop"/>
    </sps:encoding>
    <sps:values>
      <eop:CoverageProgrammingRequest>
        <eop:RegionOfInterest>
          <eop:Polygon>
            <eop:Exterior elementCount="5">
              <eop:Point>
                <eop:Lat>60.0</eop:Lat>
                <eop:Lon>60.0</eop:Lon>
              </eop:Point>
              <eop:Point>
                <eop:Lat>61.0</eop:Lat>
                <eop:Lon>60.0</eop:Lon>
              </eop:Point>
              <eop:Point>
                <eop:Lat>61.0</eop:Lat>
                <eop:Lon>61.0</eop:Lon>
              </eop:Point>
              <eop:Point>
                <eop:Lat>60.0</eop:Lat>
                <eop:Lon>61.0</eop:Lon>
              </eop:Point>
              <eop:Point>
                <eop:Lat>60.0</eop:Lat>
                <eop:Lon>60.0</eop:Lon>
              </eop:Point>
            </eop:Exterior>
          </eop:Polygon>
        </eop:RegionOfInterest>
        <eop:TimeOfInterest>
          <eop:SurveyPeriod>
            <eop:min>2008-01-01T00:00:00Z</eop:min>
            <eop:max>2008-02-03T00:00:00Z</eop:max>
          </eop:SurveyPeriod>
        </eop:TimeOfInterest>
        <eop:AcquisitionType>
          <eop:MonoscopicAcquisition>
            <eop:IncidenceAngle>
              <eop:Elevation>
                <eop:min>20.0</eop:min>
                <eop:max>40.0</eop:max>
              </eop:Elevation>
            </eop:IncidenceAngle>
            <eop:AcquisitionParametersSAR>
              <eop:InstrumentMode>EW</eop:InstrumentMode>
              <eop:PolarizationMode>HH/HV</eop:PolarizationMode>
            </eop:AcquisitionParametersSAR>
          </eop:MonoscopicAcquisition>
        </eop:AcquisitionType>
      </eop:CoverageProgrammingRequest>
    </sps:values>
  </sps:ParameterData>

```

```
</sps:taskingParameters>
</sps:GetFeasibility>
```

#### 4.1.5.5. Output Specification

- ❑ Synchronous returned XML file compliant to the GetFeasibility response.

#### 4.1.5.6. Reference data

- ❑ ReferenceGetFeasibilityResponse-Feasible.xml: expected XML file corresponding to the response to the GetFeasibility request. Note that the taskID is generated by the server dynamically, so will not be the same as below.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <GetFeasibilityResponse xmlns:sps="http://www.opengis.net/sps/2.0">
      <sps:result>
        <sps:FeasibilityReport>
          <sps:title>CFI based Feasibility Study</sps:title>
          <sps:sensorID>urn:ESA:sensors:Sentinel-1:C-SAR</sps:sensorID>
          <sps:taskID> http://www.deimos-space.com/sps/Tue Jan 27 10:49:21
GMT 2009</sps:taskID>
          <sps:updateTime>2008-01-01T18:52:31Z</sps:updateTime>
          <sps:statusCode>FEASIBLE</sps:statusCode>
          <sps:extendedData>
            <sps:ParameterData>
              <sps:encoding>
                <swe:XMLEncoding xmlns:swe="http://www.opengis.net/swe/2.0"
namespace="http://www.esa.int/eop"/>
              </sps:encoding>
              <sps:values>
                <eop:FeasibilityStudy xmlns:eop="http://www.esa.int/eop">
                  <eop:EstimatedCost>0.0</eop:EstimatedCost>
                  <eop:EstimatedSegments elementCount="2">
                    <eop:Segment>
                      <eop:ID>SEG_001</eop:ID>
                      <eop:Platform>Terrasar</eop:Platform>
                      <eop:InstrumentMode>EW</eop:InstrumentMode>
                      <eop:StartTime>2008-01-12T13:30:39Z</eop:StartTime>
                      <eop:StopTime>2008-01-12T13:24:27Z</eop:StopTime>
                      <eop:IncidenceAngle>
                        <eop:Elevation>0.0</eop:Elevation>
```

```
</eop:IncidenceAngle>
<eop:Footprint>
  <eop:Exterior elementCount="5">
    <eop:Point>
      <eop:Lat>59.52722831939547</eop:Lat>
      <eop:Lon>59.81855775763444</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>61.20993750780577</eop:Lat>
      <eop:Lon>60.02503384403735</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>58.97578809427712</eop:Lat>
      <eop:Lon>60.873238562795436</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>60.712529508069984</eop:Lat>
      <eop:Lon>61.083924196515554</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>59.52722831939547</eop:Lat>
      <eop:Lon>59.81855775763444</eop:Lon>
    </eop:Point>
  </eop:Exterior>
</eop:Footprint>
</eop:Segment>
<eop:Segment>
  <eop:ID>SEG_002</eop:ID>
  <eop:Platform>Terrasar</eop:Platform>
  <eop:InstrumentMode>EW</eop:InstrumentMode>
  <eop:StartTime>2008-01-21T13:27:36Z</eop:StartTime>
  <eop:StopTime>2008-01-21T13:29:39Z</eop:StopTime>
  <eop:IncidenceAngle>
    <eop:Elevation>0.0</eop:Elevation>
  </eop:IncidenceAngle>
  <eop:Footprint>
    <eop:Exterior elementCount="5">
      <eop:Point>
```



```
<eop:Lat>61.0002253213824</eop:Lat>
  <eop:Lon>60.33485622433769</eop:Lon>
</eop:Point>
<eop:Point>
  <eop:Lat>62.70890498617043</eop:Lat>
  <eop:Lon>60.54334073641997</eop:Lon>
</eop:Point>
<eop:Point>
  <eop:Lat>60.64649251388478</eop:Lat>
  <eop:Lon>61.00000597169676</eop:Lon>
</eop:Point>
<eop:Point>
  <eop:Lat>62.38998816492376</eop:Lat>
  <eop:Lon>61.211225806118954</eop:Lon>
</eop:Point>
<eop:Point>
  <eop:Lat>61.0002253213824</eop:Lat>
  <eop:Lon>60.33485622433769</eop:Lon>
</eop:Point>
</eop:Exterior>
</eop:Footprint>
</eop:Segment>
</eop:EstimatedSegments>
</eop:FeasibilityStudy>
</sps:values>
</sps:ParameterData>
</sps:extendedData>
</sps:FeasibilityReport>
</sps:result>
</GetFeasibilityResponse>
</soapenv:Body>
</soapenv:Envelope>
```

#### 4.1.5.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].
- The contents of the returned XML file matches the reference file from section 4.1.5.6.

## 4.1.6. Test Case TC-0060: GetFeasibility not feasible request

### 4.1.6.1. Purpose

This test case checks that the response to a non-feasible GetFeasibility request to DEIMOS SPS is correctly given.

### 4.1.6.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0030                      SR-FUN-0210

### 4.1.6.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

### 4.1.6.4. Input Specification

- GetFeasibility-notFeasible.txt:** GetFeasibility request (SOAP message) with the following contents:

```
<sps:GetFeasibility service="SPS" version="2.0.0"
xsi:schemaLocation="http://www.opengis.net/sps/2.0
http://schemas.opengis.net/sps/2.0.0/sps.xsd"
xmlns:eop="http://www.esa.int/eop" xmlns:sps="http://www.opengis.net/sps/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <sps:sensorID>urn:ESA:sensors:Sentinel-1:C-SAR</sps:sensorID>
  <sps:taskingParameters>
    <sps:ParameterData>
      <sps:encoding>
        <swe:XMLEncoding namespace="http://www.esa.int/eop"/>
      </sps:encoding>
      <sps:values>
        <eop:CoverageProgrammingRequest>
          <eop:RegionOfInterest>
            <eop:Polygon>
              <eop:Exterior elementCount="5">
                <eop:Point>
                  <eop:Lat>60.0</eop:Lat>
                  <eop:Lon>60.0</eop:Lon>
                </eop:Point>
                <eop:Point>
                  <eop:Lat>61.0</eop:Lat>
                  <eop:Lon>60.0</eop:Lon>
                </eop:Point>
                <eop:Point>
                  <eop:Lat>61.0</eop:Lat>
                  <eop:Lon>61.0</eop:Lon>
                </eop:Point>
                <eop:Point>
                  <eop:Lat>60.0</eop:Lat>
                  <eop:Lon>61.0</eop:Lon>
                </eop:Point>
              </eop:Exterior>
            </eop:Polygon>
          </eop:RegionOfInterest>
        </eop:CoverageProgrammingRequest>
      </sps:values>
    </sps:ParameterData>
  </sps:taskingParameters>
</sps:GetFeasibility>
```

```

        <eop:Point>
            <eop:Lat>60.0</eop:Lat>
            <eop:Lon>60.0</eop:Lon>
        </eop:Point>
    </eop:Exterior>
</eop:Polygon>
</eop:RegionOfInterest>
<eop:TimeOfInterest>
    <eop:SurveyPeriod>
        <eop:min>2007-01-01T00:00:00Z</eop:min>
        <eop:max>2007-02-03T00:00:00Z</eop:max>
    </eop:SurveyPeriod>
</eop:TimeOfInterest>
<eop:AcquisitionType>
    <eop:MonoscopicAcquisition>
        <eop:IncidenceAngle>
            <eop:Elevation>
                <eop:min>20.0</eop:min>
                <eop:max>40.0</eop:max>
            </eop:Elevation>
        </eop:IncidenceAngle>
        <eop:AcquisitionParametersSAR>
            <eop:InstrumentMode>EW</eop:InstrumentMode>
            <eop:PolarizationMode>HH/HV</eop:PolarizationMode>
        </eop:AcquisitionParametersSAR>
    </eop:MonoscopicAcquisition>
</eop:AcquisitionType>
</eop:CoverageProgrammingRequest>
</sps:values>
</sps:ParameterData>
</sps:taskingParameters>
</sps:GetFeasibility>

```

#### 4.1.6.5. Output Specification

- Synchronous returned XML file compliant to the GetFeasibility response.

#### 4.1.6.6. Reference data

- ReferenceGetFeasibilityResponse-notFeasible.xml: expected XML file corresponding to the response to the GetFeasibility request. Note that the taskID is generated by the server dynamically, so will not be the same as below.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <GetFeasibilityResponse xmlns:sps="http://www.opengis.net/sps/2.0">
      <sps:result>
        <sps:FeasibilityReport>
          <sps:title>CFI based Feasibility Study</sps:title>
          <sps:sensorID>urn:ESA:sensors:Sentinel-1:C-SAR</sps:sensorID>
          <sps:taskID>
http://www.deimos-space.com/sps/Tue Jan 27 10:49:21 GMT 2009
</sps:taskID>

```

```

    <sps:updateTime>2009-02-04T15:05:26.234Z</sps:updateTime>
    <sps:statusCode>REJECTED</sps:statusCode>
  </sps:FeasibilityReport>
</sps:result>
</GetFeasibilityResponse>
</soapenv:Body>
</soapenv:Envelope>

```

#### 4.1.6.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].
- The contents of the returned XML file matches the reference file from section 4.1.6.6.

### 4.1.7. *Test Case TC-0070: Submit correct response*

#### 4.1.7.1. Purpose

This test case checks that the response to a Submit request is correctly given.

#### 4.1.7.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### 4.1.7.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.7.4. Input Specification

- Submit.txt:** Submit request (SOAP message) with the following contents:

```

<sps:Submit service="SPS" version="2.0.0"
xsi:schemaLocation="http://www.opengis.net/sps/2.0
http://schemas.opengis.net/sps/2.0.0/sps.xsd"
xmlns:eop="http://www.esa.int/eop" xmlns:sps="http://www.opengis.net/sps/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <sps:sensorID>urn:ESA:sensors:Sentinel-1:C-SAR</sps:sensorID>
  <sps:taskingParameters>
    <sps:ParameterData>
      <sps:encoding>
        <swe:XMLEncoding namespace="http://www.esa.int/eop"/>
      </sps:encoding>
      <sps:values>
        <eop:CoverageProgrammingRequest>
          <eop:RegionOfInterest>
            <eop:Polygon>
              <eop:Exterior elementCount="5">
                <eop:Point>

```

```

        <eop:Lat>60.0</eop:Lat>
        <eop:Lon>60.0</eop:Lon>
    </eop:Point>
    <eop:Point>
        <eop:Lat>61.0</eop:Lat>
        <eop:Lon>60.0</eop:Lon>
    </eop:Point>
    <eop:Point>
        <eop:Lat>61.0</eop:Lat>
        <eop:Lon>61.0</eop:Lon>
    </eop:Point>
    <eop:Point>
        <eop:Lat>60.0</eop:Lat>
        <eop:Lon>61.0</eop:Lon>
    </eop:Point>
    <eop:Point>
        <eop:Lat>60.0</eop:Lat>
        <eop:Lon>60.0</eop:Lon>
    </eop:Point>
</eop:Exterior>
</eop:Polygon>
</eop:RegionOfInterest>
<eop:TimeOfInterest>
    <eop:SurveyPeriod>
        <eop:min>2008-01-01T00:00:00Z</eop:min>
        <eop:max>2008-02-03T00:00:00Z</eop:max>
    </eop:SurveyPeriod>
</eop:TimeOfInterest>
<eop:AcquisitionType>
    <eop:MonoscopicAcquisition>
        <eop:IncidenceAngle>
            <eop:Elevation>
                <eop:min>20.0</eop:min>
                <eop:max>40.0</eop:max>
            </eop:Elevation>
        </eop:IncidenceAngle>
        <eop:AcquisitionParametersSAR>
            <eop:InstrumentMode>EW</eop:InstrumentMode>
            <eop:PolarizationMode>HH/HV</eop:PolarizationMode>
        </eop:AcquisitionParametersSAR>
    </eop:MonoscopicAcquisition>
</eop:AcquisitionType>
</eop:CoverageProgrammingRequest>
</sps:values>
</sps:ParameterData>
</sps:taskingParameters>
</sps:Submit>

```

#### 4.1.7.5. Output Specification

- XML file corresponding to a Submit request response.

#### 4.1.7.6. Reference data

- ReferenceSubmitResponse.xml: expected XML file corresponding to the response to the Submit request.

```

<soapenv:Envelope>
  <soapenv:Body>
    <SubmitResponse>
      <sps:result>
        <sps:StatusReport>
          <sps:title>CFI based Feasibility Study</sps:title>
          <sps:sensorID>urn:ESA:sensors:Sentinel-1:C-SAR</sps:sensorID>
          <sps:taskID>
http://www.deimos-space.com/sps/Tue Jan 27 11:17:26 GMT 2009
          </sps:taskID>
          <sps:updateTime>2008-01-01T18:52:31Z</sps:updateTime>
          <sps:statusCode>FEASIBLE</sps:statusCode>
          <sps:extendedData>
            <sps:ParameterData>
              <sps:encoding>
                <swe:XMLEncoding namespace="http://www.esa.int/eop"/>
              </sps:encoding>
              <sps:values>
                <eop:FeasibilityStudy>
                  <eop:EstimatedCost>0.0</eop:EstimatedCost>
                  <eop:EstimatedSegments elementCount="2">
                    <eop:Segment>
                      <eop:ID>SEG_001</eop:ID>
                      <eop:Platform>Terrasar</eop:Platform>
                      <eop:InstrumentMode>EW</eop:InstrumentMode>
                      <eop:StartTime>2008-01-12T13:30:39Z</eop:StartTime>
                      <eop:StopTime>2008-01-12T13:24:27Z</eop:StopTime>
                      <eop:IncidenceAngle>
                        <eop:Elevation>0.0</eop:Elevation>
                      </eop:IncidenceAngle>
                      <eop:Footprint>
                        <eop:Exterior elementCount="5">
                          <eop:Point>
                            <eop:Lat>59.52722831939547</eop:Lat>
                            <eop:Lon>59.81855775763444</eop:Lon>
                          </eop:Point>
                          <eop:Point>
                            <eop:Lat>61.20993750780577</eop:Lat>
                            <eop:Lon>60.02503384403735</eop:Lon>
                          </eop:Point>
                          <eop:Point>
                            <eop:Lat>58.97578809427712</eop:Lat>
                            <eop:Lon>60.873238562795436</eop:Lon>
                          </eop:Point>
                          <eop:Point>
                            <eop:Lat>60.712529508069984</eop:Lat>
                            <eop:Lon>61.083924196515554</eop:Lon>
                          </eop:Point>
                          <eop:Point>
                            <eop:Lat>59.52722831939547</eop:Lat>
                            <eop:Lon>59.81855775763444</eop:Lon>
                          </eop:Point>
                        </eop:Exterior>
                      </eop:Footprint>
                    </eop:Segment>
                    <eop:Segment>
                      <eop:ID>SEG_002</eop:ID>

```

```

<eop:Platform>Terrasar</eop:Platform>
<eop:InstrumentMode>EW</eop:InstrumentMode>
<eop:StartTime>2008-01-21T13:27:36Z</eop:StartTime>
<eop:StopTime>2008-01-21T13:29:39Z</eop:StopTime>
<eop:IncidenceAngle>
  <eop:Elevation>0.0</eop:Elevation>
</eop:IncidenceAngle>
<eop:Footprint>
  <eop:Exterior elementCount="5">
    <eop:Point>
      <eop:Lat>61.0002253213824</eop:Lat>
      <eop:Lon>60.33485622433769</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>62.70890498617043</eop:Lat>
      <eop:Lon>60.54334073641997</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>60.64649251388478</eop:Lat>
      <eop:Lon>61.00000597169676</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>62.38998816492376</eop:Lat>
      <eop:Lon>61.211225806118954</eop:Lon>
    </eop:Point>
    <eop:Point>
      <eop:Lat>61.0002253213824</eop:Lat>
      <eop:Lon>60.33485622433769</eop:Lon>
    </eop:Point>
  </eop:Exterior>
</eop:Footprint>
</eop:Segment>
</eop:EstimatedSegments>
</eop:FeasibilityStudy>
</sps:values>
</sps:ParameterData>
</sps:extendedData>
</sps>StatusReport>
</sps:result>
</SubmitResponse>
</soapenv:Body>
</soapenv:Envelope>

```

#### 4.1.7.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].
- The contents of the returned XML file matches the reference file from section 4.1.7.6.

### 4.1.8. *Test Case TC-0080: GetStatus Known response*

#### 4.1.8.1. Purpose

This test case checks that the response to a GetStatus request is correctly given.

#### 4.1.8.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### 4.1.8.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.8.4. Input Specification

- GetStatus.txt:** GetStatus request (SOAP message) with the following contents:

```
<sps:GetStatus service="SPS" version="2.0.0"
xmlns:esa="http://www.esa.int/sps" xmlns:sps="http://www.opengis.net/sps/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<sps:taskID>http://www.deimos-space.com/sps/Wed Jan 21 11:17:46 GMT
2009</sps:taskID>
</sps:GetStatus>
```

*taskID* being a string value with the identifier returned by a previously invoked Submit request.

#### 4.1.8.5. Output Specification

- XML file corresponding to a GetStatus request response. The file content varies depending on the status returned.

#### 4.1.8.6. Reference data

- ReferenceGetStatusResponse.xml: expected XML file corresponding to the response to the GetStatus request.

```
<soapenv:Envelope>
  <soapenv:Body>
    <GetStatusResponse>
      <sps:result>
        <sps:StatusReport>
          <sps:title/>
          <sps:abstract/>
          <sps:sensorID/>
          <sps:taskID>
http://www.deimos-space.com/sps/Tue Jan 27 11:17:26 GMT 2009
</sps:taskID>
          <sps:updateTime>2009-01-27T00:00:00Z</sps:updateTime>
          <sps:statusCode>finished</sps:statusCode>
          <sps:estimatedToC>2009-01-27T11:29:11.748Z</sps:estimatedToC>
        </sps:StatusReport>
      </sps:result>
    </GetStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>
```



#### 4.1.8.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].
- The returned message corresponds to a GetStatus request response and the status is one of the following:
  - “Unknown”
  - “Inoperation”
  - “Finished”
  - “Not yet started”
  - “Cancelled”
  - “Delayed”

### 4.1.9. *Test Case TC-0090: GetStatus Unknown response*

#### 4.1.9.1. Purpose

This test case checks that the response to a GetStatus request with a TaskId, which is not recorded into the Database, is correctly given.

#### 4.1.9.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### 4.1.9.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.9.4. Input Specification

- GetStatus.txt:** GetStatus request (SOAP message) where the TaskId does not match with any of the tasks inside the database – ie a previous Submit request response. In particular for this example the previous TaskId (from test TC-0080) can be modified in order to get this file:

```
<sps:GetStatus service="SPS" version="2.0.0"
xmlns:esa="http://www.esa.int/sps" xmlns:sps="http://www.opengis.net/sps/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<sps:taskID>http://www.deimos-space.com/sps/Wed Jan 21 11:17:46 GMT
2009</sps:taskID>
</sps:GetStatus>
```

#### 4.1.9.5. Output Specification

- XML file corresponding to a GetStatus request response. The file content varies depending on the status returned.

#### 4.1.9.6. Reference data

- ReferenceGetStatusResponse-unknown.xml: expected XML file corresponding to the response to the GetStatus request.

```

<soapenv:Envelope>
  <soapenv:Body>
    <GetStatusResponse>
      <sps:result>
        <sps:StatusReport>
          <sps:title>null</sps:title>
          <sps:abstract>TaskId      not      found      into      the
database</sps:abstract>
          <sps:sensorID>null</sps:sensorID>
          <sps:taskID>
            http://www.deimos-space.com/sps/Wed   Jan   21
11:17:46 GMT 2009
          </sps:taskID>
          <sps:updateTime>2009-01-
29T11:26:25.802Z</sps:updateTime>
          <sps:statusCode>Unknown</sps:statusCode>
          <sps:estimatedToC>2009-01-
29T11:26:25.802Z</sps:estimatedToC>
        </sps:StatusReport>
      </sps:result>
    </GetStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

#### 4.1.9.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].
- The contents of the returned XML file matches the reference file from section 4.1.9.6.

### **4.1.10. Test Case TC-0100: DescribeResultAccess correct response**

#### 4.1.10.1. Purpose

This test case checks that the response to a DescribeResultAccess request is correctly given.

#### 4.1.10.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### 4.1.10.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.10.4. Input Specification

- DescribeResultAccess request (SOAP message) with the following contents:

```

<sps:DescribeResultAccess service="SPS" version="2.0.0"
xmlns:esa="http://www.esa.int/sps" xmlns:sps="http://www.opengis.net/sps/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0"

```

```
xmlns:gml="http://www.opengis.net/gml/3.2"  
xmlns:xlink="http://www.w3.org/1999/xlink"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <sps:ID>http://www.deimos-space.com/sps/Wed Jan 21 11:17:46 G 2009</sps:ID>  
</sps:DescribeResultAccess>
```

#### 4.1.10.5. Output Specification

- XML file corresponding to a DescribeResultAccess request response. The file content varies depending on the status returned.

#### 4.1.10.6. Reference data

- DescribeResultAccess-correct.xml: expected XML file corresponding to the response to the DescribeResultAccess request.

```
<soapenv:Envelope>  
  <soapenv:Body>  
    <DescribeResultAccessResponse>  
      <ows:ReferenceGroup>  
        <ows:Title>Testing Value</ows:Title>  
        <ows:Abstract/>  
        <ows:Identifier>OGC</ows:Identifier>  
      </ows:ReferenceGroup>  
    </DescribeResultAccessResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

#### 4.1.10.7. Pass/Fail Criteria

- The returned message conforms to [OGC 07-018].
- The returned message contains correct information from the requested SPS.

### ***4.1.11. Test Case TC-0110: DescribeResultAccess unavailable data response***

#### 4.1.11.1. Purpose

This test case checks that the response to a DescribeResultAccess request where the data related with the TaskId parameter is not available.

#### 4.1.11.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010                      SR-FUN-0020                      SR-FUN-0210

#### 4.1.11.3. Resource and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F.

#### 4.1.11.4. Input Specification

- DescribeResultAccess request (SOAP message) with the following contents:

```
<sps:DescribeResultAccess service="SPS" version="2.0.0"
xmlns:esa="http://www.esa.int/sps" xmlns:sps="http://www.opengis.net/sps/2.0"
xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <sps:ID>http://www.deimos-space.com/sps/Wed Jan </sps:ID>
</sps:DescribeResultAccess>
```

#### 4.1.11.5. Output Specification

- XML file corresponding to a DescribeResultAccess request response. The file content varies depending on the status returned.

#### 4.1.11.6. Reference data

- DescribeResultAccess-unavailable.xml: expected XML file corresponding to the response to the DescribeResultAccess request.

```
<soapenv:Envelope>
  <soapenv:Body>
    <DescribeResultAccessResponse>
      <sps:DataNotAvailable>
        <sps:reasonCode>data currently unavailable</sps:reasonCode>
        <sps:description>The information was not found into the data
base</sps:description>
      </sps:DataNotAvailable>
    </DescribeResultAccessResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

### 4.1.12. *Test Case TC-0120: Performance on a Linux platform*

#### 4.1.12.1. Purpose

This test case checks the performance of the DEIMOS SPS system deployed on a Linux server.

#### 4.1.12.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0020                      SR-PER-0010                      SR-PER-0020                      SR-RES-0010

#### 4.1.12.3. Resources and Tools

- Same as TC-0010.

#### 4.1.12.4. Input Specification

- Same as TC-0020 to TC-0110 (for performance measurement).

#### 4.1.12.5. Output Specification

- Same as TC-0010, and TC-0020 to TC-0110 (for performance measurement).

#### 4.1.12.6. Pass/Fail Criteria

- PFC of TC-0010, and TC-0020 to TC-0110 (for performance measurement).
- The SPS is deployed on a Linux server.
- The response from the SPS took less than 60 seconds.

### **4.1.13. Test Case TC-0130: Performance on a Windows Server platform**

#### 4.1.13.1. Purpose

This test case checks the performance of the SPOT SPS system deployed on a Windows Server 2003.

#### 4.1.13.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-PER-0010                      SR-FUN-0020                      SR-PER-0020                      SR-RES-0020

#### 4.1.13.3. Resources and Tools

- Same as TC-0010.

#### 4.1.13.4. Input Specification

- Same as of TC-0010.

#### 4.1.13.5. Output Specification

- Same as TC-0010.

#### 4.1.13.6. Pass/Fail Criteria

- PFC of TC-0010
- The SPS is deployed on a Windows 2003 server.
- The response from the SPS took less than 60 seconds.

## 4.2. SPOT SPS I/F Specific Test Cases

### 4.2.1. Test Case TC-1010: SPS I/F functionality without notification

#### 4.2.1.1. Purpose

This test case checks an end-to-end function without using the notification service. Thus the operations to be part of the test are all those included in the interface:

- GetCapabilities
- DescribeTasking
- GetFeasibility
- Submit
- GetSatus
- DescribeResultAccess

#### 4.2.1.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010	SR-FUN-0020	SR-FUN-0210	SR-PER-0010
SR-PER-0020	SR-INT-0010	SR-RES-0020	

#### 4.2.1.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F as well as a web client used to visualize requests/responses in plain XML.

#### 4.2.1.4. Input Specification

- GetCapabilities: none
- DescribeTasking: one of the sensors listed in the capabilities
- GetFeasibility: tasking parameters described in the DescribeTasking response
- Submit: same as GetFeasibility
- GetStatus: task identifier returned in the Submit response
- DescribeResultAccess: task identifier returned in the Submit response

#### 4.2.1.5. Output Specification

In all cases the response is returned in a XML file.

- GetCapabilities: capabilities containing the list of sensors that the user can task using the service
- DescribeTasking: the list and the description of the tasking parameters for a given sensor
- GetFeasibility: the list and the description, including footprint, of the scenes that may be acquired if the user sends a Submit request with same tasking parameters
- Submit: a task identifier

- GetStatus: a progress report, containing the status of the task and the list of scenes that have been acquired
- DescribeResultAccess: the URL of Web Services that serve the acquired images

#### 4.2.1.6. Pass/Fail Criteria

- The responses conform to [OGC 07-018].

### **4.2.2. Test Case TC-1020: SPS I/F functionality with notification**

#### 4.2.2.1. Purpose

This test case checks an end-to-end function using the notification service. Thus the operations to be part of the test are all those included in the interface:

- All operations tested in Test Case TC-1010: SPS I/F functionality without notification
- Subscribe
- Notify

#### 4.2.2.2. Software Requirements Verified

The following requirements from [SRD] are covered by this test case:

SR-FUN-0010	SR-FUN-0020	SR-FUN-0040	SR-FUN-0210
SR-FUN-0230	SR-FUN-0310	SR-FUN-0320	SR-PER-0010
SR-PER-0020	SR-INT-0010	SR-RES-0020	

#### 4.2.2.3. Resources and Tools

- Client simulator able to send and receive the SOAP request (HTTP POST) to SPS I/F as well as a web client used to visualize requests/responses in plain XML.

#### 4.2.2.4. Input Specification

- Same as 4.2.1.4
- Subscribe: task identifier returned in the Submit response + one (or more) event(s) listed in the capabilities
- Notify: none

#### 4.2.2.5. Output Specification

- Same as 4.2.1.5
- Subscribe: subscription identifier
- Notify: identifier of the subscription, identifier of the event that occurred, date of the event, task identifier, status of the task, sensorID, EstimatedTimeOfCompletion.

#### 4.2.2.6. Pass/Fail Criteria

- The responses conform to [OGC 07-018].

## 5. SYSTEM TEST PROCEDURES

References are made below to “Client simulator” which connects to the SPS. Tests involve this client sending messages to the SPS Interface and receiving return messages. These will be checked for conformance to [SRD] and (for SPS Application profile messages) [OGC 07-018].

For DEIMOS this client may be a COTS SOAP tester such as the Eclipse Web Service Explorer, which is a standard part of the Web Tools Platform. This has a simple forms interface for creating and sending messages and displaying responses. Details can be found at: <http://www.eclipse.org/webtools/jst/components/ws/1.0/tutorials/WebServiceExplorer/WebServiceExplorer.html>. For Spot Image the client will be the SpaceTime Toolkit and possibly a web browser in order to visualize requests/responses in plain XML.

Alternatively a third-party, perhaps Spacebel, may have a suitable client system with a form based interface (dynamically created after analysing the DescribeTasking response). If this is available at testing time then it is another possibility for the “Client simulator”.

The Client Simulator will have to be set up to make its requests to the correct SPS. This is done via setting a service endpoint reference, as defined by the WS-Addressing standard [WS Addressing]. In practice this should be as easy as giving the correct base URL (eg in the WSDL file if using the Eclipse Web Service Explorer) for either the DEIMOS or SPOT SPS. The tests have to be carried out against both systems, and in one case the results compared. This comparison can be done offline.

The simulation of a client and the creation of testing SOAP messages will of course also be of assistance in providing a demonstration of the developed SPS interfaces.

As for the test cases, this section has been organised according to the different kinds of test procedures:

- Common test procedures – applicable to both SPS
- DEIMOS STS specific test procedures – applicable only to DEIMOS SPS
- SPOT STS specific test procedures – applicable only to SPOT SPS

### 5.1. DEIMOS SPS Test Procedures

#### 5.1.1. Test Procedure TP-0010: Installation

##### 5.1.1.1. Objective

This procedure covers the execution of the following test cases:

- Test Case TC-0010: Installation

##### 5.1.1.2. Pre-conditions

- The Operational platform (operational machine plus the target operating system) is available.
- A WAR file named HMA-SPS.war containing the web application to be deployed.

##### 5.1.1.3. Procedure Steps

1. Stop Apache Tomcat if running



2. Copy the WAR file HMA-SPS.war to \$CATALINA\_HOME/webapps. The environment variable \$CATALINA\_HOME indicates the path where Apache Tomcat is installed, e.g. /opt/apache-tomcat-6.0.18
3. Update \$CATALINA\_HOME/bin/setenv.sh (create if it does not exist) with the following information

```
#!/bin/sh
export LD_LIBRARY_PATH=$CATALINA_HOME/webapps/HMA-SPS/WEB-INF/lib/CFIs/lib
export ROP_PATH=$CATALINA_HOME/webapps/HMA-SPS/WEB-INF/lib/CFIs/ropFiles
```

4. Start Apache Tomcat
5. A new folder with the name HMA-SPS is created in \$CATALINA\_HOME/webapps as result of the deployment of the HMA-SPS.war file.
6. Shutdown Apache Tomcat
7. Update the permissions of the shared libraries libCFIWrapper.so and libCFIWrapperJNI.so contained in \$CATALINA\_HOME/webapps/HMA-SPS/WEB-INF/lib/CFIs/lib

```
cd $CATALINA_HOME/webapps/HMA-SPS/WEB-INF/lib/CFIs/lib
chmod +x *.so
```

8. Start Apache Tomcat

## 5.1.2. Test Procedure TP-0020: GetCapabilities response

### 5.1.2.1. Objective

This procedure covers the execution of the following test cases:

- Test Case TC-0020: GetCapabilities response

### 5.1.2.2. Pre-conditions

- DEIMOS SPS is deployed as web service
- Client simulator able to connect to the web service using HTTP GET and display XML responses

### 5.1.2.3. Procedure Steps

1. Open the Web browser and enter the following URL that requests a GetCapabilities operation on the requested SPS listed in the input specification of test case 0010:  
`http://hostname:port/HMA-SPS/GetCapabilitiesServlet?service=SPS&request=GetCapabilities` where "hostname:port" the Internet Protocol hostname or numeric address and the port number used for the interaction.
2. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a GetCapabilities response. For this, compare the returned message/file with the reference file stored in the reference directory (tp/tp-0020/reference/GetCapabilitiesResponse.xml)

### **5.1.3. Test Procedure TP-0030: DescribeTasking responses**

#### **5.1.3.1. Objective**

This procedure covers the execution of the following test cases:

- Test Case TC-0030: DescribeTasking correct message response
- Test Case TC-0040: DescribeTasking incorrect message response

#### **5.1.3.2. Pre-conditions**

- SPS is deployed as web service
- Eclipse client simulator is running and is able to connect to the HMA-SPS web service using HTTP POST for SOAP messages and is able to display SOAP responses

#### **5.1.3.3. Procedure Steps**

1. Right-click on the HMA-SPS.wsdl file of the HMA-T project at the left side of the Eclipse platform, and select the “Web Services→Test with Web Services Explorer” option. Check that the Web Services Explorer appears on the right and central frame of the tool and set the endpoint if necessary – click the HMA-SPSSOAP link on the main frame and select “Add” next to “Endpoints”. It should be the same URL as received in the GetCapabilities response. Click “Go” to add it.
2. On the Navigator area click on the “DescribeTasking” operation under the HMA-SPSSOAP item (or alternatively click on the HMA-SPSSOAP link on the main frame and select the “DescribeTasking” operation).
3. Check that a form appears in the main frame devoted to specify the parameters of the WSDL operation (DescribeTasking in this case). Click on the “source” link at the top right side of the window.
4. Go down to the “<soapenv:Body>” field of the XML source code and click on the “Browse” button.
5. Select the “tp/tp-0030/input/CorrectDescribeTasking.txt” file and press the “Load” link. Check that the contents of the input file matches the Input Specification of TC-0030.
6. Move to the bottom of the main frame and click on the “Go” button.
7. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a DescribeTasking response. For this, compare the returned message/file with the reference file stored tp/tp-0030/reference/ DescribeTaskingResponse.xml.
8. Repeat steps 4 to 6 but in step 5 use the  
“tp/tp-0030/input/IncorrectDescribeTasking.txt” input file instead.
9. Check that a fault appears in the “<soapenv:Body>” of the response. Optionally, if the server machine is accessible then the console of the SPS Tomcat installation for the SPS can be checked to see that it identifies that the values for the “SERVICES” and “VERSION” parameters are missing.

### **5.1.4. Test Procedure TP-0040: GetFeasibility response**

#### **5.1.4.1. Objective**

This procedure covers the execution of the following test cases:

- Test Case TC-0050: GetFeasibility feasible reques
- Test Case TC-0060: GetFeasibility not feasible request

#### 5.1.4.2. Pre-conditions

- SPS is deployed as web service
- Eclipse client simulator is running and is able to connect to the HMA-SPS web service using HTTP POST for SOAP messages and is able to display XML responses

#### 5.1.4.3. Procedure Steps

1. Right-click on the HMA-SPS.wsdl file of the HMA-T project at the left side of the Eclipse platform, and select the “Web Services→Test with Web Services Explorer” option. Check that the Web Services Explorer appears on the right and central frame of the tool and set the endpoint if necessary – click the HMA-SPSSOAP link on the main frame and select “Add” next to “Endpoints”. It should be the same as received in the GetCapabilities response. Click “Go” to add it.
2. On the Navigator area click on the “GetFeasibility” operation under the HMA-SPSSOAP item (or alternatively click on the HMA-SPSSOAP link on the main frame and select the “GetFeasibility” operation).
3. Check that a form appears in the main frame devoted to specify the parameters of the WSDL operation (GetFeasibility in this case). Click on the “source” link at the top right side of the window.
4. Go down to the “<soapenv:Body>” field of the XML source code and click on the “Browse” button.
5. Select the “hma-t/tp/tp-0040/input/GetFeasibility-Feasible.txt” file and press the “Load” link. Check that the contents of the input file matches the Input Specification of TC-0050.
6. Move to the bottom of the main frame and click on the “Go” button.
7. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a GetFeasibility response. For this, compare the returned message/file with the file stored in the reference directory (tp/tp-0040/reference/ReferenceGetFeasibility-Feasible.xml).
8. Select the “hma-t/tp/tp-0040/input/GetFeasibility-NotFeasible.txt” file and press the “Load” link. Check that the contents of the input file matches the Input Specification of TC-0060.
9. Move to the bottom of the main frame and click on the “Go” button.
10. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a GetFeasibility response. For this, compare the returned message/file with the file stored in the reference directory (tp/tp-0040/reference/ReferenceGetFeasibility-NotFeasible.xml).

### **5.1.5. Test Procedure TP-0050: Submit / GetStatus response**

#### 5.1.5.1. Objective

This procedure covers the execution of the following test cases:

- Test Case TC-0070: Submit correct response

- The contents of the returned XML file matches the reference file from section 4.1.7.6.
- Test Case TC-0080: GetStatus Known response
- Test Case TC-0090: GetStatus Unknown response

#### 5.1.5.2. Pre-conditions

- SPS is deployed as web service
- Eclipse client simulator is running and is able to connect to the HMA-SPS web service using HTTP POST for SOAP messages and is able to display SOAP responses

#### 5.1.5.3. Procedure Steps

1. Right-click on the HMA-SPS.wsdl file of the HMA-T project at the left side of the Eclipse platform, and select the “Web Services→Test with Web Services Explorer” option. Check that the Web Services Explorer appears on the right and central frame of the tool and set the endpoint if necessary – click the HMA-SPSSOAP link on the main frame and select “Add” next to “Endpoints”. It should be the same as received in the GetCapabilities response. Click “Go” to add it.
2. On the Navigator area click on the “Submit” operation under the HMA-SPSSOAP item (or alternatively click on the HMA-SPSSOAP link on the main frame and select the “Submit” operation).
3. Check that a form appears in the main frame devoted to specify the parameters of the WSDL operation (Submit in this case). Click on the “source” link at the top right side of the window.
4. Go down to the “<soapenv:Body>” field of the XML source code and click on the “Browse” button.
5. Select the “hma-t/tp/tp-0050/input/Submit.txt” file and press the “Load” link. Check that the contents of the input file matches the Input Specification of TC-0080.
6. Access to the bottom of the main frame and click on the “Go” button.
7. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a Submit response. For this, compare the returned message/file with the file stored in the reference directory (tp/tp-0050/reference/ReferenceSubmitResponse.xml).
8. Access the Navigator area and click on the “GetStatus” operation. Enter the feasibility identifier used for the Submit operation in the “TaskId” field on the main area and click on the “Go” button.
9. Check that the response is an XML file conformant to [OGC 07-018] and contains the expected data for a GetStatus response. Check that the returned status is set to “Finished”.
10. Modify the “TaskId” field used in the previous GetStatus Input operation. The aim of this modification is get a “TaskId” which does not match any of the database entries. Modifying the date inside the “TaskId” field to one in the future is the recommended method.
11. Check that the response is an XML file conformant to [OGC 07-018] and contains the expected data for a GetStatus response with returned status set to “Unknown”. For this, compare the returned message/file with the file stored in the reference directory (tp/tp-0050/reference/ReferenceSubmitResponse-unkown.xml).

### **5.1.6. Test Procedure TP-0060: DescribeResultAccess response**

#### **5.1.6.1. Objective**

This procedure covers the execution of the following test cases:

#### **5.1.6.2. Pass/Fail Criteria**

- The returned message conforms to [OGC 07-018].
- The contents of the returned XML file matches the reference file from section 4.1.9.6.
- Test Case TC-0100: DescribeResultAccess correct response
- Test Case TC-0110: DescribeResultAccess unavailable data response

#### **5.1.6.3. Pre-conditions**

- SPS is deployed as web service
- Eclipse client simulator is running and is able to connect to the HMA-SPS web service using HTTP POST for SOAP messages and is able to display SOAP responses

#### **5.1.6.4. Procedure Steps**

1. Right-click on the HMA-SPS.wsdl file of the HMA-T project at the left side of the Eclipse platform, and select the “Web Services→Test with Web Services Explorer” option. Check that the Web Services Explorer appears on the right and central frame of the tool and set the endpoint if necessary – click the HMA-SPSSOAP link on the main frame and select “Add” next to “Endpoints”. It should be the same as received in the GetCapabilities response. Click “Go” to add it.
2. On the Navigator area click on the “DescribeResultAccess” operation under the HMA-SPSSOAP item (or alternatively click on the HMA-SPSSOAP link on the main frame and select the “DescribeResultAccess” operation).
3. Check that a form appears in the main frame devoted to specify the parameters of the WSDL operation (DescribeResultAccess in this case). Click on the “source” link at the top right side of the window.
4. Go down to the “<soapenv:Body>” field of the XML source code and click on the “Browse” button.
5. Select the “hma-t/tp/tp-0060/input/DescribeResultAccess-Correct.txt” file and press the “Load” link. Check that the contents of the input file matches the Input Specification of TC-0100.
6. Move to the bottom of the main frame and click on the “Go” button.
7. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a DescribeResultAccess response where data is available. For this, compare the returned message/file with the reference file stored tp/tp-0060/reference/DescribeResultAccess-correct.xml.
8. Modify the “TaskId” field used in the previous GetStatus Input operation. The aim of this modification is get a “TaskId” which does not match any of the database entries. Modifying the date inside the “TaskId” field to one in the future is the recommended method. Check that the contents of the input file matches the Input Specification of TC-0110.
9. Move to the bottom of the main frame and click on the “Go” button.

10. Check that the message returned is an XML document conformant to [OGC 07-018] and contains the expected data for a DescribeResultAccess response where the data is unavailable. For this, compare the returned message/file with the reference file stored tp/tp-0060/reference/ DescribeResultAccess-unavailable.xml.

## 5.2. SPOT SPS I/F Test Procedures

### 5.2.1. Test Procedure TP-1010: SPS I/F functionality without notification

#### 5.2.1.1. Objective

This procedure covers the execution of the following test cases:

- Test Case TC-1010: SPS I/F functionality without notification

#### 5.2.1.2. Pre-conditions

- SPOT SPS is deployed as web service

#### 5.2.1.3. Procedure Steps

The procedure steps are shown in the sequence diagram shown below:

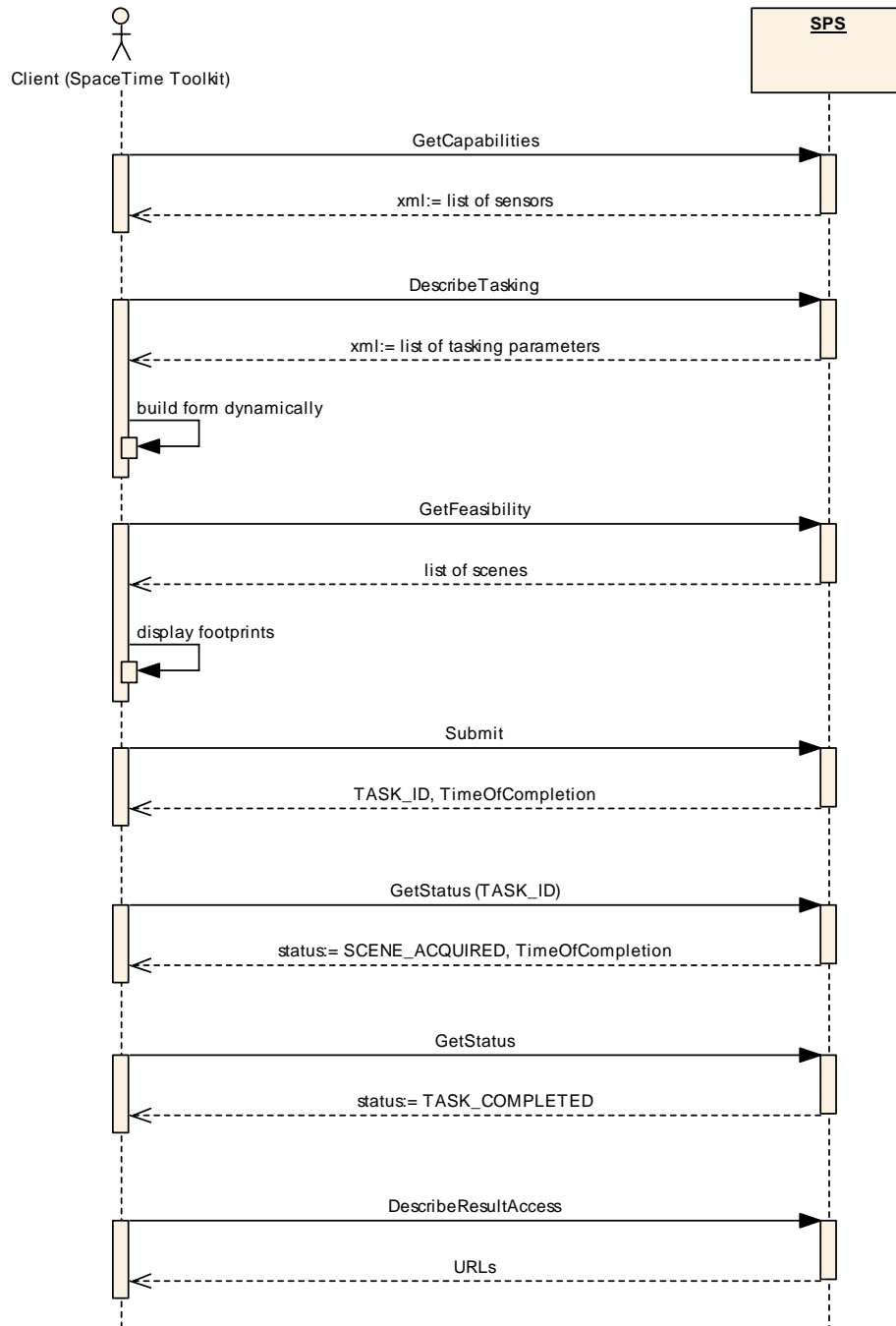


Figure 5-1: Test procedure for SPOT SPS I/F operations using the notification service

## 5.2.2. Test Procedure TP-1020: SPS I/F functionality with notification

### 5.2.2.1. Objective

This procedure covers the execution of the following test cases:

- ❑ Test Case TC-1020: SPS I/F functionality with notification

### 5.2.2.2. Pre-conditions

- ❑ SPOT SPS is deployed as web service

### 5.2.2.3. Procedure Steps

The procedure steps are shown in the sequence diagram shown below:

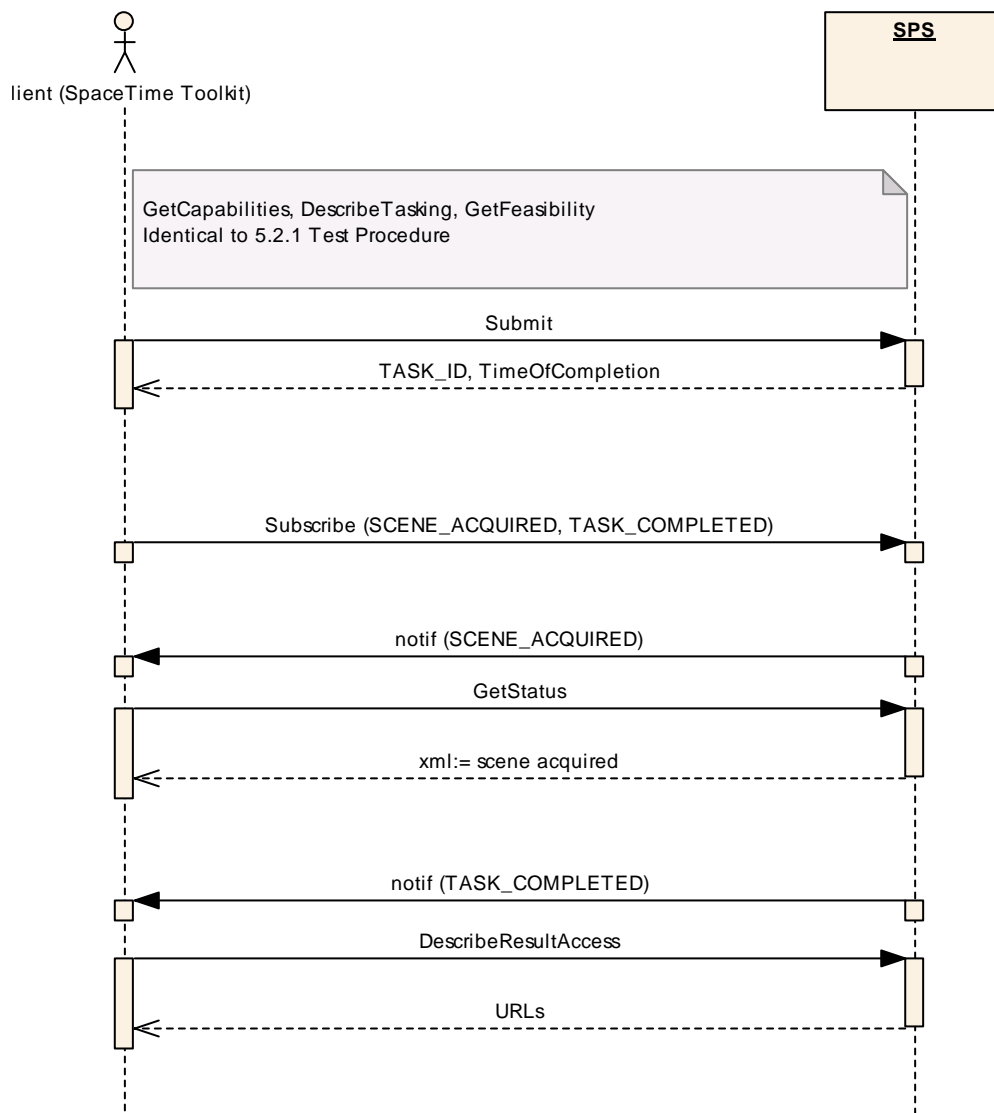


Figure 5-2: Test procedure for SPOT SPS I/F operations using the notification service