

P-50638/DSASGT-1763-10/00

25 June, 2010

**CONTRACT N. ESA/ESRIN 22508/09/I-LG**

## **OPGW Software Validation Specification HMA FOLLOW ON TASK 4 – ORDER**

### SUMMARY

This document provides the Test Designs, Test Cases and Test Procedures for verifying and validating the OPGW Software.

**Prepared by:**Daniele MARCHIONNI

Certifies that the current document was prepared analyzing input documents and standards applicable to the project; cooperated with reference roles (OBS) reporting to a technical responsible.

(Project Manager)

**Verified by:**Daniele MARCHIONNI

Certifies: consistency and completeness of the content with respect to contract and company/divisional procedures and other outputs generated by previous phases; the correspondence to applicable standards; absence of orthographic and typographical errors.

(Project Manager)

**Technical Approval:**Daniele MARCHIONNI

Certifies validity of technical decisions and document coherence with both upper-level documentation and documents correlated to the expected project phase.

(Project Manager)

**Quality Approval:**Raffaele BARBATI

Certifies that: document complies with defined requirements; the process which generated it was performed as foreseen; quality records exist for the quality activities performed; any action generated was closed.

(SPA Manager)

**Authorized by:**Mario FONTI

Certifies that the document complies with directives and constraints set by Purchaser and that any possible commitment arisen by document is compatible with project constraints; authorizes usage and distribution.

(SW Engineering Manager)

**Distribution**

Pier Giorgio MARCHETTI – ESA

Yves COENE – Spacebel

Uwe VOGES – con terra

**Release and Edition Register**

Edition	Release	Date	Amendments description	Author
1	0	25/06/2010	First Issue	D. MARCHIONNI



# Table of Contents

1	Introduction .....	9
2	Applicable and Reference documents .....	10
3	Terms, definition and abbreviations terms .....	12
4	OPGW Software overview .....	14
4.1	Software static architecture.....	14
4.1.1	Components Overview.....	14
4.1.2	Relationship with other systems.....	15
4.1.3	Information Model Overview.....	16
4.2	Interfaces Context .....	16
5	OPGW Software validation specification task identification .....	20
5.1	Tasks & Criteria .....	20
5.2	Features to be tested .....	20
5.3	Features not to be tested .....	20
5.4	Test case Pass/Fail Criteria.....	20
5.5	Items that cannot be validated by test.....	21
6	OPGW Software validation testing specification design .....	22
6.1	TD_001: Installation & configuration.....	22
6.1.1	General .....	22
6.1.2	Test Items & Features to be tested.....	22
6.1.3	Approach refinements.....	22
6.2	TD_010: Ordering Service Verification .....	23
6.2.1	General .....	23
6.2.2	Test Items & Features to be tested.....	23
6.2.3	Approach refinements.....	23
6.3	TD_020: Joint Ordering & Programming Service Verification.....	24
6.3.1	General .....	24
6.3.2	Test Items & Features to be tested.....	24
6.3.3	Approach refinements.....	24
7	OPGW Software validation test case specification .....	25
7.1	TC_001_001: OPGW Installation, Configuration, Start-up .....	25
7.1.1	Test case identifier .....	25
7.1.2	Inputs specifications .....	25
7.1.3	Outputs specifications.....	26
7.1.4	Test pass - fail criteria .....	26
7.1.5	Environmental needs.....	26
7.1.6	Special procedural constraints .....	26
7.1.7	Interfaces dependencies.....	27
7.2	TC_010_001: Ordering Service - GetCapabilities .....	28
7.2.1	Test case identifier .....	28
7.2.2	Inputs and Outputs specifications .....	28
7.2.3	Test pass - fail criteria .....	28
7.2.4	Environmental needs.....	28
7.2.5	Special procedural constraints .....	28



- 7.2.6 Interfaces dependencies .....29
- 7.3 TC\_010\_010: Ordering Service - GetOptions .....30
  - 7.3.1 Test case identifier .....30
  - 7.3.2 Inputs and Outputs specifications .....30
  - 7.3.3 Test pass - fail criteria .....31
  - 7.3.4 Environmental needs.....31
  - 7.3.5 Special procedural constraints .....31
  - 7.3.6 Interfaces dependencies .....31
- 7.4 TC\_010\_020: Ordering Service - GetQuotation .....32
  - 7.4.1 Test case identifier .....32
  - 7.4.2 Inputs and Outputs specifications .....32
  - 7.4.3 Test pass - fail criteria .....32
  - 7.4.4 Environmental needs.....32
  - 7.4.5 Special procedural constraints .....32
  - 7.4.6 Interfaces dependencies .....32
- 7.5 TC\_010\_030: Ordering Service - Submit .....33
  - 7.5.1 Test case identifier .....33
  - 7.5.2 Inputs and Outputs specifications .....33
  - 7.5.3 Test pass - fail criteria .....36
  - 7.5.4 Environmental needs.....36
  - 7.5.5 Special procedural constraints .....36
  - 7.5.6 Interfaces dependencies .....36
- 7.6 TC\_010\_040: Ordering Service - GetStatus .....37
  - 7.6.1 Test case identifier .....37
  - 7.6.2 Inputs and Outputs specifications .....37
  - 7.6.3 Test pass - fail criteria .....38
  - 7.6.4 Environmental needs.....38
  - 7.6.5 Special procedural constraints .....38
  - 7.6.6 Interfaces dependencies .....38
- 7.7 TC\_010\_050: Ordering Service - SubmitResponse .....39
  - 7.7.1 Test case identifier .....39
  - 7.7.2 Inputs and Outputs specifications .....39
  - 7.7.3 Test pass - fail criteria .....39
  - 7.7.4 Environmental needs.....39
  - 7.7.5 Special procedural constraints .....39
  - 7.7.6 Interfaces dependencies .....40
- 7.8 TC\_010\_060: Ordering Service - Cancel.....41
  - 7.8.1 Test case identifier .....41
  - 7.8.2 Inputs and Outputs specifications .....41
  - 7.8.3 Test pass - fail criteria .....41
  - 7.8.4 Environmental needs.....41
  - 7.8.5 Special procedural constraints .....41
  - 7.8.6 Interfaces dependencies .....41
- 7.9 TC\_010\_070: Ordering Service - DescribeResultAccess .....42
  - 7.9.1 Test case identifier .....42



- 7.9.2 Inputs and Outputs specifications .....42
- 7.9.3 Test pass - fail criteria .....42
- 7.9.4 Environmental needs.....42
- 7.9.5 Special procedural constraints .....42
- 7.9.6 Interfaces dependencies .....42
- 7.10 TC\_020\_001: Ordering & Programming Service – Coverage Order.....43
  - 7.10.1 Test case identifier .....43
  - 7.10.2 Input and Output specifications .....43
  - 7.10.3 Test pass - fail criteria .....43
  - 7.10.4 Environmental needs.....43
  - 7.10.5 Special procedural requirements .....44
  - 7.10.6 Interfaces dependencies .....44
  - 7.10.7 Test script.....44
- 8 OPGW Software validation test procedures.....45
  - 8.1 TP\_001\_001\_001: OPGW Installation, Configuration, Start-up .....45
    - 8.1.1 Purpose .....45
    - 8.1.2 Procedure Steps .....45
    - 8.1.3 Test script.....45
  - 8.2 TP\_010\_001\_001: Ordering Service - GetCapabilities .....46
    - 8.2.1 Purpose .....46
    - 8.2.2 Procedure Steps .....46
    - 8.2.3 Test script.....46
  - 8.3 TP\_010\_010\_001: Ordering Service - GetOptions.....47
    - 8.3.1 Purpose .....47
    - 8.3.2 Procedure Steps .....47
    - 8.3.3 Test script.....48
  - 8.4 TP\_010\_020\_001: Ordering Service - GetQuotation .....49
    - 8.4.1 Purpose .....49
    - 8.4.2 Procedure Steps .....49
    - 8.4.3 Test script.....49
  - 8.5 TP\_010\_030\_001: Ordering Service – Submit .....50
    - 8.5.1 Purpose .....50
    - 8.5.2 Procedure Steps .....50
    - 8.5.3 Test script.....50
  - 8.6 TP\_010\_040\_001: Ordering Service - GetStatus .....51
    - 8.6.1 Purpose .....51
    - 8.6.2 Procedure Steps .....51
    - 8.6.3 Test script.....51
  - 8.7 TP\_010\_050\_001: Ordering Service - SubmitResponse .....52
    - 8.7.1 Purpose .....52
    - 8.7.2 Procedure Steps .....52
    - 8.7.3 Test script.....52
  - 8.8 TP\_010\_060\_001: Ordering Service - Cancel .....53
    - 8.8.1 Purpose .....53
    - 8.8.2 Procedure Steps .....53



- 8.8.3 Test script.....53
- 8.9 TP\_010\_070\_001: Ordering Service - DescribeResultAccess .....54
  - 8.9.1 Purpose .....54
  - 8.9.2 Procedure Steps .....54
  - 8.9.3 Test script.....54
- 8.10 TP\_020\_001\_001: Ordering & Programming Service – Coverage Order .....55
  - 8.10.1 Purpose .....55
  - 8.10.2 Procedure Steps .....55
  - 8.10.3 Test script.....56
- 9 OPGW Software validation analysis, inspection, review of design .....57
- 10 OPGW Validation test platform requirements.....59
  - 10.1 Test Platform .....59
    - 10.1.1 Network Architecture .....59
    - 10.1.2 Test tool .....59
- 11 Requirements verification matrix.....60
  - 11.1 Requirements vs. Test cases traceability matrix.....60
  - 11.2 Test cases vs. Requirements traceability matrix.....70



## Indexes of Figures

Figure 4-1: OPGW Component Diagram.....	15
Figure 4-2: OPGW Context Diagram. ....	17



## Indexes of Tables

Table 2-1. Applicable Documents .....	10
Table 2-2: Reference Documents .....	10
Table 3-1: Acronyms and definitions.....	13
Table 7-1: OPGW collections configuration.....	26
Table 7-2: Users ingestion specification.....	26
Table 7-3: TC_010_001: Ordering Service - GetCapabilities input and output specifications.....	28
Table 7-4: TC_010_010: Ordering Service - GetOptions input and output specifications. ....	30
Table 7-5: TC_010_020: Ordering Service - GetQuotation.....	32
Table 7-6: TC_010_030: Ordering Service - Submit input and output specifications. ....	36
Table 7-7: TC_010_040: Ordering Service - GetStatus input and output specifications.....	37
Table 7-8: TC_010_060: Ordering Service - Cancel input and output specifications.....	41
Table 7-9: TC_010_070: Ordering Service - DescribeResultAccess input and output specifications.....	42
Table 7-10: TC_020_001 - Ordering & Programming Service – Coverage Order input and output specifications .....	43
Table 9-1: Inspection procedures.....	58
Table 11-1: Requirements vs. Test Cases traceability matrix. ....	69
Table 11-2: Test Cases vs. Requirements traceability matrix.....	79





## 1 Introduction

This document defines the Software Validation Specification of the ESA G/S Ordering & Programming Gateway – OPGW system that is the Order Server implementation required in HMA Follow On Task 4 – Order.

This document provides the definition of the Test Design, Test Case and Test Procedures identified for verifying and validating the OPGW software against the Software Requirements defined in [AD-07].

This document has been prepared according to the ESA ECSS-E-ST-40C standard, following the SVS DRD.

This document will be updated during the HMA Follow-on project for taking into account the updates on HMA Ordering ICD that will be performed in the OGC SWG.

Document content:

- the chapter 1 lists the information that can be found in this document;
- the chapter 2 provides the list of applicable and reference documents;
- the chapter 3 provides the terms and abbreviations used in this document;
- the chapter 4 provides the overview of the OPGW system to be validated;
- the chapter 5 provides the identified tasks for carrying out the verification activity;
- the chapter 6 provides the list of identified Test Designs;
- the chapter 7 provides the list of identified Test Cases;
- the chapter 8 provides the list of identified Test Procedures;
- the chapter 9 provides, for each items where it can be justified that a test is not possible, another validation method based on analysis, inspection, review of design;
- the chapter 10 provides list the validation requirements related to the validation test platform to be used;
- the chapter 11 provides the traceability matrix of test cases w.r.t. requirements.

## 2 Applicable and Reference documents

The following table provide the list of applicable documents:

Id.	Title	Reference	Issue	Date
[AD-01]	EARTHNET ONLINE XML FRONT-END INTERFACE CONTROL DOCUMENT	EOLI-XML-006-ICD	2.8	21 Jan 2008
[AD-02]	OGC™ Catalogue Services Specification 2.0 Extension Package for ebRIM (ISO/TS 15000-3) Application Profile: Earth Observation Products	OGC 06-131r6	0.2.4	07 May 2008
[AD-03]	Application schema for Earth Observation products	OGC 06-080r4	0.9.3	21 Jul 2008
[AD-04]	ECSS – Space engineering – Software	ECSS-E-ST-40C		06 Mar 2009
[AD-05]	Ordering Services for Earth Observation Products	OGC 06-141r2	0.9.5	02 Jul 2010
[AD-06]	EARTHNET ONLINE XML FRONT-END: ORDER AND ON-LINE ACCESS EXTENSION INTERFACE CONTROL DOCUMENT	EOLI/Order-XML-ICD	3.4	07 Apr 2008
[AD-07]	OPGW Software Requirements Specification Document	P50638/DSASGT-0082-10/00	1.1	25 Jun 2010
[AD-08]	Proposal for HMA Follow On Task 4 – Order	EF000D135/DSASGT-0501-09	1.0	13 Mar 2009
[AD-09]	Multi Mission User Information Services MMOHS/UMS XML ICD, UMS Import/Export XML ICD	UMS-MMOHS-XML-ICD	1.2	31 May 2006
[AD-10]	M2AS MMOHS IMPORT/EXPORT XML ICD	OSME-USMP-SEDA-IS-08-2059	1.1	21 Nov 2008
[AD-11]	User Management Interfaces for Earth Observation Services	OGC 07-118r4	0.0.6	29 Jan 2010
[AD-12]	SOFTWARE DEVELOPMENT PLAN FOR HMA FOLLOW ON TASK 4 – ORDER	P50638/DSASGT-2995-09/00	1.0	20 Nov 2009
[AD-13]	SOFTWARE PRODUCT ASSURANCE PLAN FOR HMA FOLLOW ON TASK 4 – ORDER	P-P50638/DSAQUD-3046-09/00	1.0	20 Nov 2009

**Table 2-1. Applicable Documents**

The following table provide the list of reference documents:

Id.	Title	Reference	Issue	Date
[RD-01]	HSQLDB Web Site	<a href="http://www.hsqldb.org/">http://www.hsqldb.org/</a>		
[RD-02]	EOLI-SA Configuration File Interface Control Document	VEGA.EOLI-SA.ICD.042	1.17	16 Apr 2008
[RD-03]	OPGW Software Requirements Specification Document	OSME-USMP-SEDA-RS-08-1855	1.2	20 Mar 2009
[RD-04]	Apache XML Security	<a href="http://santuario.apache.org/Java/index.html">http://santuario.apache.org/Java/index.html</a>		
[RD-05]	OPGW SDD	P50638/DSASGT-0083-10/00	1.1	25 Jun 2010
[RD-06]	Prototype Operations Concept	HMA-PR-SPB-EN-0001	1.0	12 Aug 2006

**Table 2-2: Reference Documents**



### 3 Terms, definition and abbreviations terms

Acronym	Definition
API	Application Programmer's Interface
AR	Acceptance Review
ASCII	American Standard for Code Information Interchange
ASN.1	Abstract Syntax Notation One
BNF	Backus-Naur Form
CDR	Critical Design Review
CM	Contributing Mission
COTS	Commercial off-the-shelf
DAIL	Data Access Integration Layer
DAP	Data Access Portfolio
DB	Database
EO-A	Enhanced On-line Access
EOLI	Earthnet On-line Interactive
EolISA	Earthnet On-line Interactive and Stand-Alone Client: main user interface to the catalogue and ordering on-line services
ESA	European Space Agency
GMES	Global Monitoring for Environment and Security
GSC	GMES Space Component
GSDR	Ground Segment Design Review
GSOV	Ground Segment Operation Validation.
HMA	Heterogeneous Mission Accessibility
ICD	Interface Control Document
IDS	Inventory Data Server
M2EOS	Multi Mission Earth Observation Services
M2AS	Multi Mission authorization Service
M2BS	Multi Mission Browse Server
M2CS	Multi Mission Catalogue Server
MMFI	Multi Mission Facility Infrastructure
MMMC	Multi Mission Master Catalogue
MMOHS	Multi-Mission Order Handling System
MTA	Medium Term Archive
MUIS	Multi-Mission User information Services
N/A	Not Applicable
NRT	Near Real Time
OMT	Object Modelling Technique
OPGW	ESA G/S Ordering and Programming Gate-Way
OR	Operational qualification Review
OSME	Operational Support, Maintenance and Evolution Contract
PBS	Product Browse System
PDR	Preliminary Design Review
PDS	Payload Data Segment
SEDA	Serco / Elsig Datamat Consortium
SOAP	Simple Object Access Protocol
SWG	Standard Working Group
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TCP	Transmission Control Protocol

Acronym	Definition
UML	Unified Modelling Language
UMS	User Management System
USMP	User Services & Mission Planning
WSM	ORACLE Web Service Manager

**Table 3-1: Acronyms and definitions**

## 4 OPGW Software overview

This chapter provides a general overview of the system specifying:

- The context in which the system is operating
- The background of the project and relationships with other projects
- An high level view of OPGW design.

### 4.1 Software static architecture

This section reports the main components of the system, the identified relationships, the different statuses in which the system is operating and the model of the main information items handled by it.

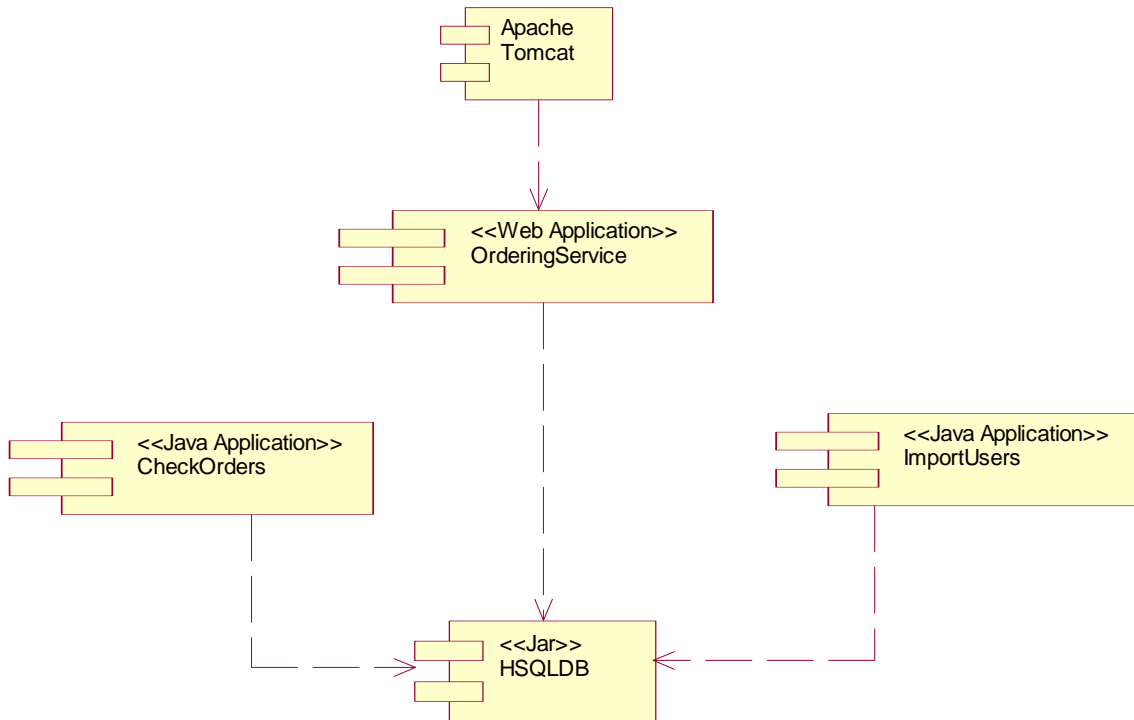
#### 4.1.1 Components Overview

The Ordering & Programming Gateway - OPGW system is in charge of providing a reference implementation of HMA Ordering ICD [AD-05] in order to allow the clients to exercise their implementation of that protocol.

Due to the nature of the HMA Ordering specification, OPGW has to deal with:

- user identity management including encryption and digital signature: in fact the ordering concept is tightly linked with the user profile, user accounting and delivery and then these information must be managed with the suitable level of confidentiality. The way this information is transferred between client and server is specified in HMA User Identity Management ICD [AD-11].
- Asynchronous web services operations: the ordering process is a long lasting activity and then the operations of that specification are mainly asynchronous i.e. the client triggers the task, but the corresponding job is completed after some time (e.g. planning of the new data to acquire, actual sensing of the data, processing and delivery).
- Protocol translation: the HMA ordering service is actually implemented by reformatting the request in the protocol suitable for the legacy system OPGW is connected to i.e. EOLI XML Ordering protocol [AD-06].

The following component diagram (UML notations) shows the main software components included in OPGW:



**Figure 4-1: OPGW Component Diagram.**

Because OPGW implements the Ordering Service ICD, which is a Web Service, then it is mainly developed as a Java Web Application deployed in Tomcat. In particular, OPGW includes the following components:

- Tomcat, which is the basic infrastructure for the implementation of the HMA Ordering Web Service
- Ordering Service, which is the java web application, developed customizing the HMA Skeleton, in charge of implementing the operations of the HMA Order Service [AD-05].
- HSQLDB, which is a light weight RDBMS written in java, that provides the storage for OPGW data i.e.: the received and submitted orders, the minimum user profile.
- ImportUsers, which is a command line Java application in charge of loading the minimal user profile information necessary for submitting an order to the EOLI XML Order server connected to OPGW.
- CheckOrders, which is a command line Java application, in charge of sending asynchronous notification to the client.

#### 4.1.2 Relationship with other systems

OPGW has several relationships with other systems:

- **SSE**

The SSE (Service Support Environment) is an open, interoperable system based on widely accepted standards from W3C, OASIS, WSI and OGC. It implements a Service Oriented Architecture (SOA) facilitating access to and deployment of services and combining services using workflow technology. Services can be Ground Segment related modules such as catalogues, ordering but also external services for oil spill monitoring, fire risk, algae bloom, coordinate transformation, classification etc.

In the frame of HMA Follow-on project, SSE will be the main client of OPGW and a dedicated activity will be performed for integrating OPGW between the list of SSE connected services.

- **External OGC Catalogue**

The product ordering process is performed after that the client has identified the EO Product of interest, then an EBRIM Catalogue has to be properly configured and loaded in order to return to users products that OPGW is able to order. Then OPGW and this catalogue shall share the configured collections and the loaded EO Product metadata.

- **M2EOS**

OPGW is not an ordering system (i.e. a system that does processing, formatting and delivery of products), but it is a gateway that translates the input request in a request that is suitable for triggering an actual ordering system.

So, OPGW will be connected to M2EOS system (the new system that will replace the MUIS, which is the system currently in operation that provides access to ESA users) via the EOLI XML Order protocol [AD-06], which is the currently operationally used protocol for ordering products.

### 4.1.3 Information Model Overview

In order to implement the HMA Order protocol, OPGW has to store the following information items:

- **ESA GS Users minimal profile information:** in fact to allow the EOLI XML Order server (M2EOS) accepting the order request, OPGW shall provide correct credentials of an actual ESA GS user, then it has to map the user identifier specified in the input request on an ESA GS user.
- **Orders:** because OPGW has to support asynchronous operations dealing with the status of submitted orders, and the HMA Order protocol [AD-05] is based on a stateless protocol (HTTP), then these orders must be stored in a persistent storage: for this purpose a small order database is included in OPGW.
- **Order Options:** for each EO Product / collection different order options are possible then this almost static information has to be held into OPGW for being returned to the client when it discovers the available ordering options.

### 4.2 Interfaces Context

The following picture reports the context in which the OPGW will operate.



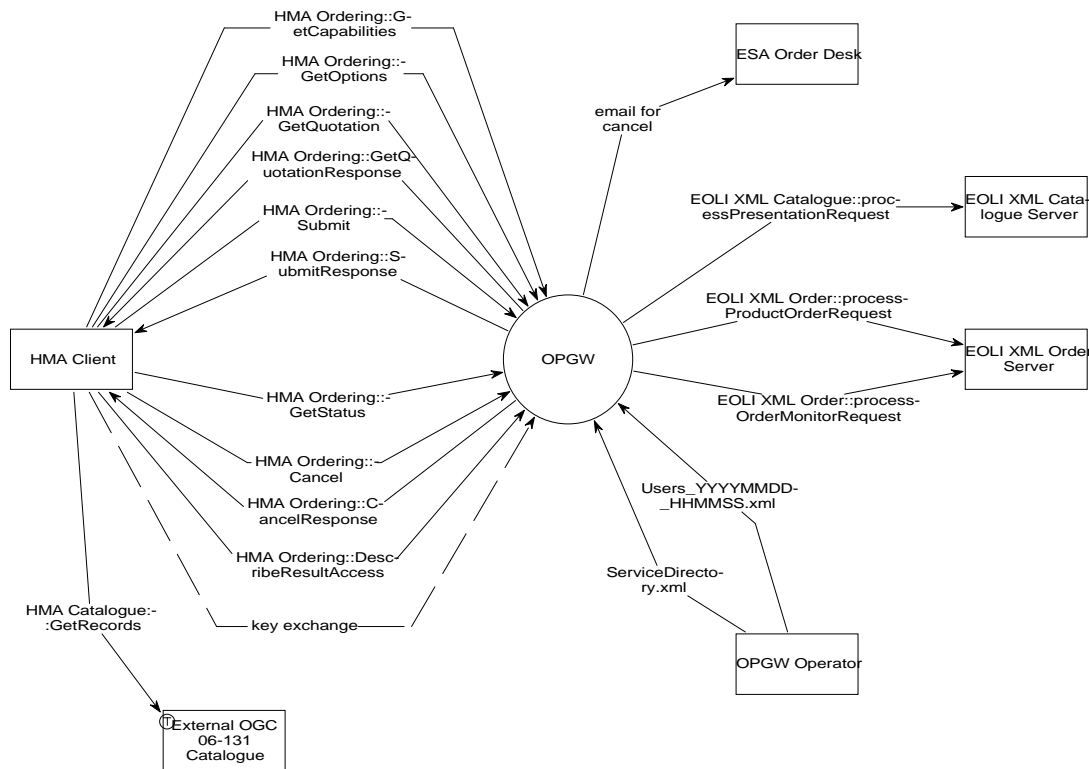


Figure 4-2: OPGW Context Diagram.

As highlighted in the previous figure, OPGW interacts with the following entities:

- **HMA Client**

It is a client in charge of sending SOAP requests compliant with OGC 06-141. The expected HMA Clients are:

- The TEAM engine, which will be used for validating the updated interfaces;
- SSE, which will be connected to OPGW and added to the list of available Service Providers.

The following are the exchanged interfaces:

- OGC 06-131, for ordering interfaces;
- OGC 07-118, for the encoding of user management information (it is the protocol currently used from the DAIL).

- **EOLI XML Catalogue Server**

It is a place holder for a Web Service implementing the EOLI XML Catalogue interface for ESA collections. At the moment, this interface is provided by the MUIS system (via DSM + IDS), but MUIS is going to be replaced by M2EOS during 2010 and then OPGW will be connected to this system (actually M2AS and then M2CS sub-system).

- **EOLI XML Order Server**

It is a place holder for a Web Service implementing the EOLI XML Ordering interface for ESA collections. At the moment, this interface is provided by MUIS system (via DSM + OFS and then MMOHS), but MUIS is going to be replaced by M2EOS during 2010 and then OPGW will be connected to this system (actually M2AS and then MMOHS).

- **OPGW Operator**

It is the entity in charge of operating the OPGW. It has to provide the following configuration items:

- Configuration files for the basic working of OPGW;
- Capabilities XML files for the implemented ordering service;
- List of configured users, with profile, i.e. the users that allowed issuing orders via OPGW.

- **External OGC 06-131 Catalogue**

This is the OGC Catalogue storing the EO Product metadata to be queried and ordered from the HMA Client.

The ordering of products is performed after having discovered the products of interest via a catalogue search. The catalogue is not needed for the working of OPGW: in fact it is able to process the orders on its own; the catalogue is needed for supporting the user on selecting the product of interest so OPGW has an indirect relationship with the External OGC Catalogue. The external catalogue is queried by the HMA Client for discovering the list of products available for ordering; then the HMA Client will send a product order to OPGW specifying the product identifiers returned by OGC Catalogue. Then OPGW and OGC Catalogue shall understand the same product identifiers.

This catalogue must be kept synchronized with OPGW in order to make sure that the EO products returned by this catalogue have the same identifiers understood from OPGW and the same EO Product metadata is available also in the connected EOLI XML Catalogue Server.

- **ESA Order Desk**

The ESA order desk is notified by e-mail in case of cancellation of an HMA Order.

ESA User Services do not support the on-line cancellation of orders, but this function is supported via interaction with the Order Desk. In order to simulate the on-line cancellation of orders, OPGW will send an e-mail to a configured address asking for the cancellation of the specified order.

The interfaces exchanged between the components and entities showed in the diagram are:

- **OPGW vs. HMA Client:**

- HMA Ordering ICD [AD-05]: it specifies the HMA interfaces for ordering products from catalogue.
- DAIL UM ICD [AD-11]: it is not explicitly mentioned in above diagram, but it is implicitly included in all DAIL interfaces;

- **OPGW vs. EOLI XML Catalogue:**

- EOLI XML Catalogue [AD-01]: it allows querying the EO products catalogue for getting mainly the visible products necessary for compiling an order.

- **OPGW vs. EOLI XML Order Server:**

- EOLI XML Ordering [AD-06]: it allows issuing EO product orders.
- Users\_YYYYMMDD\_HHMMSS.xml, it carries on the user profile information needed for preparing EOLI XML Orders to send.

- **OPGW vs. OPGW Operator:**

- Monitor & control interfaces for operating, configuring OPGW.

- **HMA Client vs. External OGC Catalogue**
  - HMA Catalogue ICD [AD-02], for getting the list of available EO products.
  
- **OPGW vs. ESA Order Desk**
  - E-mail for cancelling an order.

## 5 OPGW Software validation specification task identification

This section reports the testing approach for validating the OPGW system.

### 5.1 Tasks & Criteria

Testing tasks include the execution of test procedures (including the test result analysis) and writing of test reports and problem reports.

### 5.2 Features to be tested

This document is in charge of verifying all requirements listed in [AD-07].

### 5.3 Features not to be tested

All requirements and features not reported in [AD-07] are out of the scope of this document.

### 5.4 Test case Pass/Fail Criteria

The item pass/fail criterion depends on the verification method for the item. The verification method is specified in the requirement verification matrix provided in chapter §11. In fact for each software requirement the test cases in charge of verifying it and related verification methods are specified. Possible verification methods [AD-07] are:

#### Test verification method [T]

The method shall be referred as "Test" when requirements have to be verified by measuring product performance and function under various simulated environments. These measurements may require the use of special equipment, instrumentation and simulation techniques. Established principles and procedures shall be used to determine conformance to requirements.

The analysis of data derived from test shall be considered an integral part of the test. When relevant, Test also includes the demonstration of qualitative operational performance and requirements. The performance, as demonstrated, shall be observed and recorded.

#### Inspection verification method [I]

The method shall be referred as "Inspection" when verification is achieved by visual determination of physical characteristics (such as construction features, hardware conformance to document drawing or workmanship requirements).

#### Analysis verification method [A]

The method shall be referred as "Analysis" when verification is achieved by performing theoretical or empirical evaluation by accepted techniques. The analytical techniques shall be selected amongst systematic, statistical and qualitative design analysis, modelling and computational simulation. Verification by similarity is considered part of Analysis. It shall be applied if it can be shown that the article under verification is similar to another article that has already been verified to equivalent or more stringent requirements. The verification activity consists of the assessment and review of prior test data, hardware configuration and applications.

**Review of Design verification method [D]** The method shall be referred as "Review of design" when verification is achieved by validation of records or by evidence of validated design documents or when approved design reports, technical descriptions, engineering drawings unambiguously show that the requirement is met.

If the method is 'Inspection', the justification is analysed. If the arguments are deemed insufficient, they will be updated. If it is decided that the requirement cannot be verified by Inspection, a new test using the 'Test' method will be created.

For the 'Test' method, the expected result of each test is given in the test procedure description, for any step. In case of test failure, the test is suspended while the cause of the fault is determined.

### **5.5 Items that cannot be validated by test**

There are a number of requirements that cannot be verified by test, but manual inspection is needed for verifying their achievement, e.g.:

- Design and implementation requirements specifying the re-use of software developed in previous projects or the usage of COTS;
- The development process carried out according to ECSS standards
- Etc.

In section §9 for each of them the corresponding approach is specified.

## 6 OPGW Software validation testing specification design

This section provides the Test Designs aimed to validate the OPGW software.

For the identification of Test Designs the following naming convention is followed:

- TD\_<TD\_nnn> <Title>:
  - <TD\_nnn>: 3 digits number starting from 1 (and having generally a step by 10)
  - <Title>: title of the test design

### 6.1 TD\_001: Installation & configuration

#### 6.1.1 General

This test design deals with the building of the software, installation on run time environment, configuration and then start-up of the system.

#### 6.1.2 Test Items & Features to be tested

The aim of this test case is to verify correctness of:

- OPGW Installation Document that specifies:
  - procedures for building and deploying the system;
  - procedures the user / operator shall follow for configuring and using the system;
- OPGW delivery kit that stores all configuration items (source files, scripts, configuration files) needed to build and run the system.

#### 6.1.3 Approach refinements

The aim of this test design is:

- to set-up the system to be tested through the next test designs
- to configure the appropriate parameters e.g.: order options, tasking parameters, collections etc. in order to perform the next test designs.

The following test cases have been identified:

- TC\_001\_001 - OPGW installation, configuration, start-up

## 6.2 TD\_010: Ordering Service Verification

### 6.2.1 General

This test design deals with the verification of the correct handling of HMA Ordering ICD.

### 6.2.2 Test Items & Features to be tested

The aim of this test case is:

- to verify the correct management of the supported HMA Ordering ICD interfaces:
  - GetCapabilities
  - GetOptions
  - Submit
  - GetStatus
- To verify the correct handling of unsupported HMA Ordering ICD interfaces:
  - GetQuotation
  - Cancel
  - DescribeResultAccess
- To verify the correct handling of asynchronous notifications:
  - SubmitResponse
- To verify the correct handling of user identity information embedded in the SOAP header of Ordering requests.

### 6.2.3 Approach refinements

This test design is structured in a set of test cases each dealing with a single HMA Ordering operation. Each operation will be verified by issuing several requests towards the different configured collections.

In order to run this test design, the TD\_001: Installation & configuration has to be successfully accomplished.

The following test cases have been identified:

- TC\_010\_001: Ordering Service - GetCapabilities
- TC\_010\_010: Ordering Service - GetOptions
- TC\_010\_020: Ordering Service - GetQuotation
- TC\_010\_030: Ordering Service - Submit
- TC\_010\_040: Ordering Service - GetStatus
- TC\_010\_050: Ordering Service - SubmitResponse
- TC\_010\_060: Ordering Service - Cancel
- TC\_010\_070: Ordering Service - DescribeResultAccess

## 6.3 TD\_020: Joint Ordering & Programming Service Verification

### 6.3.1 General

The aim of this Test Design is to verify the correct handling of orders for future products issued via the joint usage of HMA Ordering & Programming ICD.

### 6.3.2 Test Items & Features to be tested

The aim of this test case is to verify correctness of the following operations of Ordering & Programming Services when used together for the ordering of future products:

- Ordering::GetOptions, for getting the ordering parameters corresponding to the feasibility analysis. The feasibility analysis is performed outside OPGW, but it has to be performed by calling the companion SPS instance;
- Ordering::Submit, for submitting the order including the order options and a reference to the tasking request;
- Ordering::GetStatus of submitted order.

### 6.3.3 Approach refinements

The approach is to re-use the test cases defined for testing the Ordering & Programming services alone and then adding the missing test cases.

Of course this test design can be run only when the previous ones have been completed successfully.

List of test cases related to this test design

- TC\_020\_001 - Ordering & Programming Service – Coverage Order



## 7 OPGW Software validation test case specification

This section provides the Test Cases aimed to validate the OPGW software.

For the identification of Test Cases the following naming convention is followed:

- TC\_<TD\_nnn>\_<TC\_nnn> <Title>:
  - <TD\_nnn>: is the progressive number of the Test Design the Test Case belongs to
  - <TC\_nnn>: 3 digits number starting from 1 (and having generally a step by 10)
  - <Title>: title of the test case

### 7.1 TC\_001\_001: OPGW Installation, Configuration, Start-up

#### 7.1.1 Test case identifier

This test case deals with the building of the software, installation on run time environment, configuration and then start-up of the system.

The aim of this test case is to verify correctness of:

- OPGW Installation Document that specifies:
  - procedures for building and deploying the system;
  - procedures the user / operator shall follow for configuring and using the system;
- OPGW delivery kit that stores all configuration items (source files, scripts, configuration files) needed to build and run the system.
- OPGW users ingestion procedure.

#### 7.1.2 Inputs specifications

The following tables specify the collections and sensors configuration for the test scenario:

Collection	Description	Comment
ESA.EECF.ENVISAT_ASA_GMI_1S	ASAR Global Monitoring Stripline collection	This collection allows to order stripline products (i.e. no scene selection parameters to be specified).
ESA.EECF.ENVISAT_ASA_IMx_xS	ASAR Image Mode stripline collection	Only scene orders can be issued.
ESA.EECF.ENVISAT_ASA_IMx_xF	ASAR Image Mode Frame based collection	The product can be ordered as scene, but the scene can be identical to the parent product returned by the catalogue.
ESA.EECF.ENVISAT_ASA_APx_xS	ASAR Alternate Polarization Mode stripline collection	
ESA.EECF.ENVISAT_ASA_APx_xF	ASAR Alternate Polarization Mode Frame based collection	
ESA.EECF.ENVISAT_ASA_WSx_xS	ASAR Wide Swath Mode stripline collection	
ESA.EECF.ENVISAT_ASA_WSx_xF	ASAR Wide Swath Mode Frame based collection	
ESA.EECF.ENVISAT_MER_RR__xS	ENVISAT MERIS Reduced Resolution stripline collection	Only scene orders can be issued.
ESA.EECF.ENVISAT_MER_RR__xF	ENVISAT MERIS Reduced Resolution frame based collection	The product can be ordered as scene, but the scene can be identical to the parent product returned by the catalogue.
ESA.EECF.ENVISAT_MER_FR__xS	ENVISAT MERIS Full Resolution stripline collection	

Collection	Description	Comment
ESA.EECF.ENVISAT_MER_FR__xF	ENVISAT MERIS Full Resolution frame based collection	
ESA.EECF.SPOT_ESA_PAN	SPOT products collection hosted in the ESA User Service Catalogue.	Ordering options: <ul style="list-style-type: none"> <li>o Delivery Medium: file</li> <li>o DeliveryMethod: ftp-pickup</li> </ul> No scene selection
ESA.EECF.SPOT_ESA_MULTI	SPOT products collection hosted in the ESA User Service Catalogue.	Ordering options: <ul style="list-style-type: none"> <li>o Delivery Medium: file</li> <li>o DeliveryMethod: ftp-pickup</li> </ul> No scene selection

**Table 7-1: OPGW collections configuration.**

Because OPGW, before accepting an incoming request, checks the embedded user information, then the OPGW User Database shall be set-up loading at least one user.

Input File Name	File Type	Description	Expected Output Response
Users_001_001_01.xml	MMOHS User export file [AD-09]	MMOHS XML export file carries on at least one user able to accept order for past & future products on: <ul style="list-style-type: none"> <li>o Envisat ASAR</li> <li>o Envisat MERIS</li> <li>o SPOT HRV</li> </ul> Missions / sensors	Ingestion completed successfully.

**Table 7-2: Users ingestion specification.**

Ordering Service Configuration Files:

- Capabilities document as required in §7.2 TC\_010\_001: Ordering Service - GetCapabilities.
- Order Options (productMapping.xml) as required in §7.3 TC\_010\_010: Ordering Service - GetOptions.

### 7.1.3 Outputs specifications

The outputs of this test case are:

- the successful building of the subsystem;
- the successful configuration of all needed items;
- the successful start-up of the subsystem.
- The successful ingestion of MMOHS Users.

### 7.1.4 Test pass - fail criteria

The test is considered no passed, if either compilation or installation errors occurred.

### 7.1.5 Environmental needs

- HW & SW set-up of the test platform as specified in [AD-07];
- Installation and configuration of test tools.

### 7.1.6 Special procedural constraints

None.

### **7.1.7 Interfaces dependencies**

None: this is the first test case to run.

## 7.2 TC\_010\_001: Ordering Service - GetCapabilities

### 7.2.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect the GetCapabilities specification provided in [AD-05].

In order to test the GetCapabilities implementation, the following items are tested:

- Nominal conditions:
  - **Version negotiation;**
  - **Section parameter** is not managed and then the complete service metadata document shall be returned.
  - **Update Sequence** is not supported.
  - **AcceptFormats** is not supported and then always "text / xml" is the provided response format.
- Non nominal conditions:
  - The **VersionNegotiationFailed** and **InvalidParameterValue** exception code are tested.

Additionally it has to be verified whether the received requests are logged.

### 7.2.2 Inputs and Outputs specifications

In the following table the test messages to send to the OPGW and the expected responses are specified.

Input File Name	File Type	Description	Expected Output Response
GetCapabilities_010_001_001.xml	GetCapabilities Request	Message asking unsupported protocol	SOAP Exception specifying unsupported protocol
GetCapabilities_010_001_002.xml	GetCapabilities Request	Message asking unsupported protocol version	SOAP Exception specifying unsupported protocol version
GetCapabilities_010_001_003.xml	GetCapabilities Request	Message asking supported protocol and version: <ul style="list-style-type: none"> <li>• Service: OS</li> <li>• Version: 0.9.5</li> </ul>	Capabilities document (V0.9.5) matching the content of the configured one (the file in the capabilities directory).
GetCapabilities_010_001_004.xml	GetCapabilities Request	The same as the previous one, but including also SectionParameter, UpdateSequence and AcceptVersion parameters	The same as the previous one.

**Table 7-3: TC\_010\_001: Ordering Service - GetCapabilities input and output specifications.**

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-010.

### 7.2.3 Test pass - fail criteria

The test is considered no passed, if:

- The received response is not compliant with the expected output specified in the previous table.

### 7.2.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### 7.2.5 Special procedural constraints

None.

### **7.2.6 Interfaces dependencies**

The following test cases shall be successfully completed:

- TC\_001\_001: OPGW Installation, Configuration, Start-up

## 7.3 TC\_010\_010: Ordering Service - GetOptions

### 7.3.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect the GetOptions specification provided in [AD-05].

In order to test the GetOptions implementation the following items are tested:

- Import of options stored into ServiceDirectory.xml file got from the USMP installation the OPGW is connected to (e.g. muiscx2.esrin.esa.int)
- GetOptions is issued towards configured collections:
  - ESA.EECF.ENVISAT\_ASA\_GMI\_1S
  - ESA.EECF.ENVISAT\_ASA\_IMx\_xS
  - ESA.EECF.ENVISAT\_ASA\_IMx\_xF
  - ESA.EECF.ENVISAT\_MER\_FR\_xS
  - ESA.EECF.ENVISAT\_MER\_FR\_xF
  - ESA.EECF.SPOT\_ESA\_PAN
  - ESA.EECF.SPOT\_ESA\_MULTI
- GetOptions is issued towards a non configured collection:
  - ESA.EECF.ENVISAT\_ASA\_NOT\_CONFIG
- Check whether only requests embedding a correct SAML Token are accepted.

Additionally it has to be verified whether the received requests are logged.

### 7.3.2 Inputs and Outputs specifications

In the following table the test messages to send to the OPGW and the expected responses are specified. All messages shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
ServiceDirectory_010_010001.xml	EOLI SA ServiceDirectory file	EOLI SA ServiceDirectory.xml file valid for muiscx2 (USMP test) platform.	Generation of xml file storing order options in HMA format.
GetOptions_010_010_001.xml	GetOptions Request	Message asking options of ESA.EECF.ENVISAT_ASA_GMI_1S	List of options according to the ServiceDirectory_010_010_001.xml file
GetOptions_010_010_002.xml	GetOptions Request	Message asking options of ESA.EECF.ENVISAT_ASA_IMx_xS	List of options according to the ServiceDirectory_010_010_001.xml file
GetOptions_010_010_003.xml	GetOptions Request	Message asking options of ESA.EECF.ENVISAT_ASA_IMx_xF	List of options according to the ServiceDirectory.xml file
GetOptions_010_010_004.xml	GetOptions Request	Message asking options of ESA.EECF.ENVISAT_MER_FR_xS	List of options according to the ServiceDirectory_010_010_001.xml file
GetOptions_010_010_005.xml	GetOptions Request	Message asking options of ESA.EECF.ENVISAT_MER_FR_xF	List of options according to the ServiceDirectory_010_010_001.xml file
GetOptions_010_010_006.xml	GetOptions Request	Message asking options of ESA.EECF.ENVISAT_ASA_NOT_CONFIG	Response specifying an empty list of options.
GetOptions_010_010_007.xml	GetOptions Request	Message asking options of ESA.EECF.SPOT_ESA_PAN	List of options according to the ServiceDirectory_010_010_001.xml file
GetOptions_010_010_008.xml	GetOptions Request	Message asking options of ESA.EECF.SPOT_ESA_MULTI	List of options according to the ServiceDirectory_010_010_001.xml file

**Table 7-4: TC\_010\_010: Ordering Service - GetOptions input and output specifications.**

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-010.

### **7.3.3 Test pass - fail criteria**

The test is considered no passed, if:

- o The received response is not compliant with the expected output specified in the previous table.

### **7.3.4 Environmental needs**

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### **7.3.5 Special procedural constraints**

None.

### **7.3.6 Interfaces dependencies**

The following test cases shall be successfully completed:

- o TC\_001\_001: OPGW Installation, Configuration, Start-up

## 7.4 TC\_010\_020: Ordering Service - GetQuotation

### 7.4.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect the GetQuotation specification provided in [AD-05].

In order to test the GetQuotation implementation the following items are tested:

- As far as the GetQuotation operation is not actually supported, but it is simulated returning a preconfigured response message, then it has to be verified whether the returned response matches with the configured one.

Additionally it has to be verified whether the received requests are logged.

### 7.4.2 Inputs and Outputs specifications

In the following table the test messages to send to the OPGW and the expected responses are specified. All messages shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
GetQuotation_010_020_001.xml	GetQuotation request	Syntactically correct message asking the quotation of an order	Predefined answer reporting the quotation for that order.

**Table 7-5: TC\_010\_020: Ordering Service - GetQuotation.**

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-010.

### 7.4.3 Test pass - fail criteria

The test is considered no passed, if:

- The received response is not compliant with the expected output specified in the previous table.

### 7.4.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### 7.4.5 Special procedural constraints

None.

### 7.4.6 Interfaces dependencies

The following test cases shall be successfully completed:

- TC\_001\_001: OPGW Installation, Configuration, Start-up



## 7.5 TC\_010\_030: Ordering Service - Submit

### 7.5.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect the Submit specification provided in [AD-05].

In order to test the Submit implementation, the following items are tested:

Nominal conditions:

- Submission of an order with one item of ESA.EECF.ENVISAT\_ASA\_GMI\_1S collection;
- Submission of an order with two items of ESA.EECF.ENVISAT\_ASA\_IMx\_xS collection;
- Submission of an order with two items of ESA.EECF.ENVISAT\_ASA\_IMx\_xF collection;
- Submission of an order with two items of ESA.EECF.ENVISAT\_MER\_RR\_xS collection;
- Submission of an order with two items of ESA.EECF.ENVISAT\_MER\_RR\_xF collection;
- Submission of an order with 5 items: one product for each collection listed above.
- Submission of an order with two items of ESA.EECF.SPOT\_ESA\_PAN collection;
- Submission of an order with two items of ESA.EECF.SPOT\_ESA\_MULTI collection;

Non Nominal conditions:

- Submission of an order with product not existing in catalogue;
- Submission of an order with incorrect order options set;
- Submission of an order from unknown user.

Additionally it is verified whether:

- the order has been submitted to the connected EOLI XML Ordering Server.
- the received requests are logged.

### 7.5.2 Inputs and Outputs specifications

In the following table the test messages to send to the OPGW and the expected responses are specified.

All messages, apart from where explicitly excluded, shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
Submit_010_030_001.xml	Submit request	Order including one item of ESA.EECF.ENVISAT_ASA_GMI_1S without scene selection option and without notification (a standard web browser can be used for submitting the order). The item has to be identified by querying the catalogue with the following parameters: Area: Europe Time: 19/02/2004	Order successfully submitted.

Input File Name	File Type	Description	Expected Output Response
Submit_010_030_002.xml	Submit request	Order including one item of ESA.EECF.ENVISAT_ASA_IMx_xS with scene selection option and without notification (a standard web browser can be used for submitting the order) The item has to be identified by querying the catalogue with the following parameters: Area: Liban Time: 02/11/2006	Order successfully submitted.
Submit_010_030_003.xml	Submit request	Order including one item of ESA.EECF.ENVISAT_ASA_IMx_xF without scene selection option and without notification (a standard web browser can be used for submitting the order) The item has to be identified by querying the catalogue with the following parameters: Time: 06/06/2006	The order submitted to the MUIS shall include the scene selection options automatically generated by the OPGW. Order successfully submitted.
Submit_010_030_004.xml	Submit request	Order including: <ul style="list-style-type: none"> <li>• one item of ESA.EECF.ENVISAT_MER_FR_xS</li> <li>• with scene selection option</li> <li>• including notification on order completion.</li> <li>• ReplyTo shall point to the OPGW (the notification will be sent to itself).</li> </ul>	Order successfully submitted.
Submit_010_030_005.xml	Submit request	Order including: <ul style="list-style-type: none"> <li>• one item of ESA.EECF.ENVISAT_MER_FR_xF collection</li> <li>• without scene selection option</li> <li>• including notification on order completion (to send using the SOAP client provided with OPGW).</li> <li>• ReplyTo shall point to the OPGW (the notification will be sent to itself).</li> </ul> <p>The item has to be identified by querying the catalogue with the following parameters: Area: Liban</p>	The order submitted to the MUIS shall include the scene selection options automatically generated by the OPGW. Order successfully submitted.

Input File Name	File Type	Description	Expected Output Response
Submit_010_030_006.xml	Submit request	Order including: <ul style="list-style-type: none"> <li>one item for each collections above</li> <li>including notification on order completion (to send using the SOAP client provided with OPGW).</li> <li>ReplyTo shall point to the OPGW (the notification will be sent to itself).</li> </ul>	Order successfully submitted.
Submit_010_030_006a.xml	Submit request	Same as previous one, but user with multiple projects and the order specifies the project to account the order.	Order successfully submitted.
Submit_010_030_007.xml	Submit request	Order for ESA.EECF.ENVISAT_ASA_GML_1S including one item not present in catalogue, but with format compliant with EOLI XML resTitle, no scene selection options. Then the message is the same as Submit_010_030_001.xml but the identifier has been set to an item of 1995.	Order rejected.
Submit_010_030_008.xml	Submit request	Order for ESA.EECF.ENVISAT_ASA_GML_1S specifying no processing options (the Envisat ASAR products cannot be ordered as they are from the catalogue). Then the message is the same as Submit_010_030_001.xml but the processing options are removed.	Order rejected.
Submit_010_030_009.xml	Submit request	Order including an incorrect SAML Token specifying a non registered user.	Order rejected.
Submit_010_030_010.xml	Submit request	Order including one item of ESA.EECF.SPOT_ESA_PAN: <ul style="list-style-type: none"> <li>without scene selection option</li> <li>with notification.</li> <li>ReplyTo shall point to the OPGW (the notification will be sent to itself).</li> <li>With order options as specified in Table 7-1: OPGW collections configuration.</li> </ul> <p>The item has to be identified by querying the catalogue with the following parameters: Area: France Time: 27/06/2008</p>	Order successfully submitted.

Input File Name	File Type	Description	Expected Output Response
Submit_010_030_011.xml	Submit request	<p>Order including one item of ESA.EECF.SPOT_ESA_MULTI:</p> <ul style="list-style-type: none"> <li>o without scene selection option</li> <li>o with notification.</li> <li>o ReplyTo shall point to the OPGW (the notification will be sent to itself).</li> <li>o With order options as specified in Table 7-1: OPGW collections configuration.</li> </ul> <p>The item has to be identified by querying the catalogue with the following parameters: Area: France Time: 27/06/2008</p>	Order successfully submitted.
Submit_010_030_012.xml	Submit request	Message asking to subscribe to an available subscription.	Order successfully submitted.

**Table 7-6: TC\_010\_030: Ordering Service - Submit input and output specifications.**

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-001.

### 7.5.3 Test pass - fail criteria

The test is considered no passed, if:

- o The received response is not compliant with the expected output specified in the previous table.

### 7.5.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### 7.5.5 Special procedural constraints

None.

### 7.5.6 Interfaces dependencies

The following test cases shall be successfully completed:

- o TC\_001\_001: OPGW Installation, Configuration, Start-up

## 7.6 TC\_010\_040: Ordering Service - GetStatus

### 7.6.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect to the GetStatus specification provided in [AD-05].

In order to test the GetStatus implementation, the following items are tested:

- Order status retrieval by lastUpdate date and other filtering criteria
- Order status retrieval by order identifier
- Checking of the embedded identity information.

Additionally it is verified whether:

- the request has been forwarded to the Inked EOLI XML Ordering Server.
- the received requests are logged.

### 7.6.2 Inputs and Outputs specifications

In the following table the test messages to send to the OPGW and the expected responses are specified. All messages, apart from where explicitly excluded, shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
GetStatus_010_040_001.xml	GetStatus request	GetStatus retrieval by filtering criteria: Last update: Specify the day before the date of test execution	All orders successfully submitted during the test case TC_010_030. Additional items are not possible even if the system the OPGW is connected to is used also by other clients and then other orders can be found, but OPGW filters against the orderReference TAG: only orders flagged with the correct value are returned.
GetStatus_010_040_002.xml	GetStatus request	GetStatus retrieval by last update date: Specify the day after the date of test execution	No order shall be returned.
GetStatus_010_040_003.xml	GetStatus request	GetStatus retrieval by order id: Specify the identifier of the order returned in the response to Submit_010_030_006.xml	The response message shall include the same items submitted with Submit_010_030_006.xml message.
GetStatus_010_040_005.xml	GetStatus request	GetStatus retrieval by order status + last update: Specify all status values + day before test execution	All orders successfully submitted during the test case TC_010_030. Additional items are possible because the system the OPGW is connected to is used also by other clients and then other orders can be found.
GetStatus_010_040_004.xml	GetStatus request	Non nominal condition: Get status of a non existing order	Error message
GetStatus_010_040_010.xml	GetStatus request	Non nominal condition: The included SAML Token is not correct: it includes invalid user.	Error message

Table 7-7: TC\_010\_040: Ordering Service - GetStatus input and output specifications.

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-001.

### **7.6.3 Test pass - fail criteria**

The test is considered no passed, if:

- o The received response is not compliant with the expected output specified in the previous table.

### **7.6.4 Environmental needs**

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### **7.6.5 Special procedural constraints**

None.

### **7.6.6 Interfaces dependencies**

The following test cases shall be successfully completed:

- o TC\_001\_001: OPGW Installation, Configuration, Start-up
- o TC\_010\_030: Ordering Service - Submit

## 7.7 TC\_010\_050: Ordering Service - SubmitResponse

### 7.7.1 Test case identifier

This test case is TC\_010\_050. It is linked to the TD\_010.

It is in charge of verifying the compliance of the OPGW with respect the SubmitResponse specification provided in [AD-05].

The aim of this test is to verify whether the SubmitResponse operation is called by the OPGW on the client who asked this possibility on order submit.

In the test case TC\_010\_030 the following messages asked a notification upon order completion:

- Submit\_010\_030\_004.xml
- Submit\_010\_030\_005.xml
- Submit\_010\_030\_006.xml
- Submit\_010\_030\_010.xml
- Submit\_010\_030\_011.xml

Because the completion of orders is outside the control of OPGW, but depends on ESA MMOHS system, then this interface will be tested simulating the completion of order. Orders sent via the OPGW that need notification are stored in the OPGW database. Then it is sufficient to put a special value in the database to activate the SugbmitResponse notification.

Additionally the "ReplyTo" address specified in the message order submit is that of the OPGW, then the OPGW itself will receive and log the SubmitResponse message.

Additionally it is verified whether:

- the request has been forwarded to the linked EOLI XML Ordering Server.
- the received requests are logged.

### 7.7.2 Inputs and Outputs specifications

The SubmitResponse operation is tested by triggering completion of orders submitted by the messages:

- Submit\_010\_030\_004.xml
- Submit\_010\_030\_005.xml
- Submit\_010\_030\_006.xml
- Submit\_010\_030\_010.xml
- Submit\_010\_030\_011.xml

The received SubmitResponse messages shall include digital signature of the OPGW as specified in [AD-11].

### 7.7.3 Test pass - fail criteria

The test is considered no passed, if:

- The received response is not compliant with the expected output specified in the previous paragraph.

### 7.7.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### 7.7.5 Special procedural constraints

None.

### **7.7.6 Interfaces dependencies**

The following test cases shall be successfully completed:

- TC\_001\_001: OPGW Installation, Configuration, Start-up
- TC\_010\_030: Ordering Service - Submit



## 7.8 TC\_010\_060: Ordering Service - Cancel

### 7.8.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect the Cancel specification provided in [AD-05].

Because the OPGW does not actually implement the Cancel because orders cannot be cancelled via on-line interface on ESA User Services, then the aim of the test is just to verify whether the Cancel request message is correctly accepted and a pre-configured response message is returned back.

Additionally it has to be verified whether the received requests are logged.

### 7.8.2 Inputs and Outputs specifications

All messages, apart from where explicitly excluded, shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
Cancel_010_060_001.xml	Cancel request	Message asking the cancellation of an order. Whatever order id can be specified.	The response message shall suggest calling the EO order help desk.

**Table 7-8: TC\_010\_060: Ordering Service - Cancel input and output specifications.**

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-010.

### 7.8.3 Test pass - fail criteria

The test is considered no passed, if:

- The received response is not compliant with the expected output specified in the previous table.

### 7.8.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### 7.8.5 Special procedural constraints

None.

### 7.8.6 Interfaces dependencies

The following test cases shall be successfully completed:

- TC\_001\_001: OPGW Installation, Configuration, Start-up

## 7.9 TC\_010\_070: Ordering Service - DescribeResultAccess

### 7.9.1 Test case identifier

This test case is in charge of verifying the compliance of the OPGW with respect the DescribeResultAccess specification provided in [AD-05].

Because the OPGW does not actually implement the DescribeResultAccess because OPGW is connected to a system supporting only order via media, then the aim of the test is just to verify whether the DescribeResultAccess request message is correctly accepted and a pre-configured response message is returned back.

Additionally it has to be verified whether the received requests are logged.

### 7.9.2 Inputs and Outputs specifications

All messages, apart from where explicitly excluded, shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
DescribeResultAccess_010_060_001.xml	DescribeResultAccess request		The preconfigured response message shall be returned back.

**Table 7-9: TC\_010\_070: Ordering Service - DescribeResultAccess input and output specifications.**

Performance Verification:

The response message shall be provided with latency compliant with SR-OPGW-PR-010.

### 7.9.3 Test pass - fail criteria

The test is considered no passed, if:

- o The received response is not compliant with the expected output specified in the previous table.

### 7.9.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### 7.9.5 Special procedural constraints

None.

### 7.9.6 Interfaces dependencies

The following test cases shall be successfully completed:

- TC\_001\_001: OPGW Installation, Configuration, Start-up

## 7.10 TC\_020\_001: Ordering & Programming Service – Coverage Order

### 7.10.1 Test case identifier

This test case is in charge of verifying whether the OPGW supports correctly the joint usage of HMA Ordering & Programming ICDs for issuing orders for future products.

In order to test the submission of coverage orders, the following items are tested:

- Get ordering parameters ([AD-05] GetOptions) corresponding to the feasibility analysis. The feasibility analysis has been performed outside the OPGW system, but calling the companion SPS instance;
- Submit the order including the order options and a reference to the tasking request;
- Get the status of submitted order.

Additionally it has to be verified whether:

- the received requests are logged
- the request has been forwarded to the linked EOLI XML Server

### 7.10.2 Input and Output specifications

In the following table the test messages to send to the OPGW and the expected responses are specified. All messages, apart from where explicitly excluded, shall include a correctly formatted SAML Token specifying the correct user foreseen for the tests (see §7.1).

Input File Name	File Type	Description	Expected Output Response
GetOptions_030_001_001.xml	GetOptions	GetOptions operation asking the tasking parameters related to the feasibilityID returned from the previous message.	
Submit_020_001_002.xml	[AD-05] Submit	Message for submitting an order for future products related to the feasibilityID returned by GetFeasibility_020_030_002.xml.	SubmitAck specifying the identifier of the order.
GetStatus_020_001_003.xml	[AD-05] GetStatus	GetStatus operation for getting the status of the orderId returned by Submit_020_001_002.xml.	Order status related to Submit_020_001_002.xml. Because the order has been just submitted, then the status will be Submitted or Accepted.
	[AD-05] SubmitResponse	Enforced closing of the order.	SubmitResponse message specifying status Completed.

**Table 7-10: TC\_020\_001 - Ordering & Programming Service – Coverage Order input and output specifications**

### 7.10.3 Test pass - fail criteria

The test is considered no passed, if:

- The received response is not compliant with the expected output specified in the previous table.

### 7.10.4 Environmental needs

The hardware and software resources needed to the test case are those specified in [AD-07] and [RD-05].

### **7.10.5 Special procedural requirements**

None.

### **7.10.6 Interfaces dependencies**

The following test cases shall be successfully completed:

- TC\_001\_001: OPGW Installation, Configuration, Start-up

### **7.10.7 Test script**

Specified in the test procedure section.

## 8 OPGW Software validation test procedures

This section provides the Test Procedures aimed to validate the OPGW software.

For the identification of Test Procedures the following naming convention is followed:

- TP\_<TD\_nnn>\_<TC\_nnn>\_<TP\_nnn> <Title>:
  - <TD\_nnn>: is the progressive number of the Test Design the Test Procedure belongs to;
  - <TC\_nnn>: is the progressive number of the Test Case the Test Procedure belongs to;
  - <TP\_nnn>: 3 digits number starting from 1 (and having generally a step by 10)
  - <Title>: title of the test procedures

### 8.1 TP\_001\_001\_001: OPGW Installation, Configuration, Start-up

#### 8.1.1 Purpose

This test procedure implements the following test case:

- TC\_001\_001: OPGW Installation, Configuration, Start-up

#### 8.1.2 Procedure Steps

- Logon on the OPGW workstation.
- Build and deploy the OPGW system following the steps specified in [RD-20].
- Start-up the system following the commands specified in [RD-20].
- Verify that the system has been properly started:
  - TOMCAT is up and running
  - HSQLDB server is up and running
  - CheckOrder process is up and running
- Verify that the consumed memory does not exceed 1GB:
  - Open a UNIX shell on the machine where OPGW is installed
  - **Unix> ps -eaf1**
  - Check that the column SZ related to the java process does not exceed 1048576 value (1GB)
- Shutdown the system following the procedures specified in [RD-20].
- Verify the OPGW has been properly shutdown.

#### 8.1.3 Test script

None.

## 8.2 TP\_010\_001\_001: Ordering Service - GetCapabilities

### 8.2.1 Purpose

This test procedure implements the following test case:

- TC\_001\_001: OPGW Installation, Configuration, Start-up

### 8.2.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:  
**Default HTTP URL: "http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp"**  
**Default HTTPS URL: "https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp"**
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **GetCapabilities** in **Select Operation** list box;
- Send GetCapabilities requests to the OPGW: repeat the following steps for each message specified in [Table 7-3](#):
  - Select **GetCapabilities\_010\_001\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-3](#).
- Verify that responses are provided in less than 5 seconds.

### 8.2.3 Test script

None.

## 8.3 TP\_010\_010\_001: Ordering Service - GetOptions

### 8.3.1 Purpose

This test procedure implements the following test case:

- TC\_010\_010: Ordering Service - GetOptions

### 8.3.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
- Either the productMapping.xml is already available, or it must be generated from the EOLI SA ServiceDirectory.xml file.
- The test is executed using a standard web browser connected to the OPGW Test page.

Steps:

- EOLI SA ServiceDirectory file retrieval:

If the ServiceDirectory.xml of EOLI SA tool is not already available, then get it following these steps:

- Install the EOLI SA tool on the test machine in case it is not already available;
- Configure EOLI SA to connect to the USMP test reference environment **muiscx-aiv.ed.mmpds.esa.int**:
  - Open the EOLI SA ApplicationConfiguration.xml file which is located in:
    - For Windows platform:
      - C:\Documents and Settings\\eolisa
    - For UNIX platform:
      - \$HOME\eolisa
  - Configure the **HostName** and **PortNumber** elements with the address of the USMP installation:
 

```
<HostName>muiscx-aiv.ed.mmpds.esa.int</HostName>
<PortNumber>8081</PortNumber>
```
- Delete the ServiceDirectory.xml file already present in:
  - For Windows platform:
    - C:\Documents and Settings\\eolisa
  - For UNIX platform:
    - \$HOME\eolisa
- Start-up EOLI SA tool;
- log in the EOLI server clicking on Connect button and then filling in the log in window;
- after successful log-in the just downloaded ServiceDirectory.xml can be found in:
  - For Windows platform:
    - C:\Documents and Settings\\eolisa
  - For UNIX platform:
    - \$HOME\eolisa
- Generate the Order Options in the format suitable for OPGW calling the OrderOptionsConverterTool tool following the steps specified in [RD-20]
- Test the implementation of GetOptions operation by following these steps:
  - Login on the Client machine;

- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL:** “http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp”
  - **Default HTTPS URL:** “https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp”
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **GetOptions** in **Select Operation** list box;
- Send GetOptions requests to the OPGW: repeat the following steps for each message specified in [Table 7-4](#):
  - Select **GetOptions\_010\_010\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-4](#).
- Verify that responses are provided in less than 5 seconds.

### 8.3.3 Test script

None.



## 8.4 TP\_010\_020\_001: Ordering Service - GetQuotation

### 8.4.1 Purpose

This test procedure implements the following test case:

- TC\_010\_020: Ordering Service - GetQuotation

### 8.4.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL: "http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp"**
  - **Default HTTPS URL: "https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp"**
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **GetQuotation** in **Select Operation** list box;
- Send GetQuotation requests to the OPGW: repeat the following steps for each message specified in [Table 7-5](#):
  - Select **GetQuotation\_010\_020\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-5](#).
- Verify that responses are provided in less than 5 seconds.

### 8.4.3 Test script

None.

## 8.5 TP\_010\_030\_001: Ordering Service – Submit

### 8.5.1 Purpose

This test procedure implements the following test case:

- TC\_010\_030: Ordering Service - Submit

### 8.5.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
  - The OPGW Database shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.
- The ESA User Services the OPGW is connected to shall be up and running.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL:** “**http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp**”
  - **Default HTTPS URL:** “**https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp**”
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **Submit** in **Select Operation** list box;
- Send Submit requests to the OPGW: repeat the following steps for each message specified in [Table 7-6](#):
  - Select **Submit\_010\_030\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Check the EOLI message sent to ESA GS complies with the input HMA message;
  - Check the EOLI response sent by ESA GS;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-6](#).
- Verify that responses are provided in less than 5 seconds plus the time taken by the ESA GS to process the EOLI XML request.
- Verify the exchanged messages are correctly logged by OPGW:
  - \$TOMCAT\_HOME/webapps/opgw/hma/logs/  
In this directory the received and sent messages are logged.
  - \$TOMCAT\_HOME/webapps/opgw/hma/logs/hma\_log.xml  
This file logs the activity of the services (what operations are triggered, the invocation of MUIS-DSM, etc.).  
This path can be configured within the log4j.properties file.

### 8.5.3 Test script

None.

## 8.6 TP\_010\_040\_001: Ordering Service - GetStatus

### 8.6.1 Purpose

This test procedure implements the following test case:

- TC\_010\_040: Ordering Service - GetStatus

### 8.6.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.
- The ESA User Services the OPGW is connected to shall be up and running.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL:** “**http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp**”
  - **Default HTTPS URL:** “**https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp**”
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **GetStatus** in **Select Operation** list box;
- Send GetStatus requests to the OPGW: repeat the following steps for each message specified in **Table 7-7**:
  - Select **GetStatus\_010\_040\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in **Table 7-7**.
- Verify the HMA request messages are translated as specified in the [AD-07] in the corresponding EOLI XML one and vice-versa by checking the OPGW log directory.
- Verify that responses are provided in less than 5 seconds plus the time taken by the ESA GS to process the EOLI XML request.

### 8.6.3 Test script

None.

## 8.7 TP\_010\_050\_001: Ordering Service - SubmitResponse

### 8.7.1 Purpose

This test procedure implements the following test case:

- TC\_010\_050: Ordering Service - SubmitResponse

### 8.7.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
  - The OPGW Database shall be up and running
  - The CheckOrder tool shall up and running

Steps:

- Log in on the machine where the OPGW run;
- Connect to the database:
  - `unix>cd $HSQLDB_HOME/bin`
  - `unix>SqlTool.sh`
- From the SQL prompt, check whether at least the orders submitted during the TP\_010\_030\_001 are there:
  - `sql>select * from order_tbl;`
- update the database for triggering the notification:
  - `sql>\i ../../OPGW/scripts/ForceOrdersNotification.sql`
- wait for that the CheckOrder tool starts the notification process (the time-out is 10 minutes) or kill the process and restart it again for quick check.
- Check whether the SubmitResponse messages have been received by the OPGW and logged in the proper directory.
- Verify the HMA request messages are translated as specified in the [AD-07] in the corresponding EOLI XML one and vice-versa by checking the OPGW log directory.

### 8.7.3 Test script

None.

## 8.8 TP\_010\_060\_001: Ordering Service - Cancel

### 8.8.1 Purpose

This test procedure implements the following test case:

- TC\_010\_060: Ordering Service - Cancel

### 8.8.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL: "http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp"**
  - **Default HTTPS URL: "https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp"**
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **Cancel** in **Select Operation** list box;
- Send Cancel requests to the OPGW: repeat the following steps for each message specified in [Table 7-8](#):
  - Select **Cancel\_010\_060\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-8](#).
- Verify that responses are provided in less than 5 seconds.

### 8.8.3 Test script

None.

## 8.9 TP\_010\_070\_001: Ordering Service - DescribeResultAccess

### 8.9.1 Purpose

This test procedure implements the following test case:

- TC\_010\_070: Ordering Service - DescribeResultAccess

### 8.9.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Order service shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL:** “http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp”
  - **Default HTTPS URL:** “https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp”
- Select **local** in **Select Server** list box;
- Select **ordering** in **Select Service** list box;
- Select **DescribeResultAccess** in **Select Operation** list box;
- Send DescribeResultAccess requests to the OPGW: repeat the following steps for each message specified in [Table 7-9](#):
  - Select **DescribeResultAccess\_010\_060\_[XXX].xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-9](#).
- Verify that responses are provided in less than 5 seconds.

### 8.9.3 Test script

None.

## 8.10 TP\_020\_001\_001: Ordering & Programming Service – Coverage Order

### 8.10.1 Purpose

This test procedure implements the following test case:

- TP\_020\_001\_001: Ordering & Programming Service – Coverage Order

### 8.10.2 Procedure Steps

Pre-conditions:

- The OPGW shall be up and running:
  - The Programming service shall be up and running
  - The Ordering service shall be up and running
  - The OPGW Database shall be up and running
  - The CheckOrder tool shall be up and running
- The test is executed using a standard web browser connected to the OPGW Test page.

Steps:

- Login on the Client machine;
- Open a standard web browser (Firefox / Internet Explorer are OK).
- Connect to the OPGW test page:
  - **Default HTTP URL:** “http://<OPGW system IP>:8080/opgw/hma/web/Test.jsp”
  - **Default HTTPS URL:** “https://<OPGW system IP>:8443/opgw/hma/web/Test.jsp”
- Send **GetOptions\_020\_001\_001.xml** message:
  - Select **Ordering** in **Select Service** list box;
  - Select **GetOptions** in **Select Operation** list box;
  - modify the **GetOptions\_020\_001\_001.xml** for specifying the feasibilityID returned by the call to the companion SPS instance;
  - Select **GetOptions\_020\_001\_001.xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-10](#).
- Send **Submit\_020\_001\_002.xml** message:
  - Select **Ordering** in **Select Service** list box;
  - Select **Submit** in **Select Operation** list box;
  - modify the **Submit\_020\_001\_002.xml** for specifying the feasibilityID returned by the call to the companion SPS instance;
  - Select **Submit\_020\_001\_002.xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-10](#).
- Send **GetStatus\_020\_001\_003.xml** message:
  - Select **Ordering** in **Select Service** list box;
  - Select **GetStatus** in **Select Operation** list box;

- modify the **GetStatus\_020\_001\_003.xml** for specifying the orderId returned by the previous call;
  - Select **GetStatus\_020\_001\_003.xml** in **Select Request** list box;
  - Send the request to OPGW by clicking on **Send Request** button;
  - Verify the server response is displayed in the **Response** text area;
  - Verify the result is compliant with the expected output specified in [Table 7-10](#).
- Force order notification following the same steps described in  
TP\_010\_050\_001: Ordering Service - SubmitResponse

### 8.10.3 Test script

None.



## 9 OPGW Software validation analysis, inspection, review of design

This section reports the manual steps to be performed for verifying requirements which verification method is not Test.

The following table reports the “Non testable” requirements from [AD-07] with the descriptions of manual verification activities to be performed.

For each requirement it is defined the verification method using the classification in §5.4.

Identifier	Description	Verification Method	Inspection Activity
SR-OPGW-DI-001	The OPGW should be developed on the HMA Skeleton.	D	HMA Skeleton is composed of Java sources structured within the following packages:  be\spacebel\hma\skeleton  be\spacebel\hma\util  be\spacebel\hma\wsaddressing  then if OPGW is implemented in Java and those packages are used, then we can assume that OPGW has been implemented re-using that software.
SR-OPGW-DI-010	The OPGW development shall be based on re-use of existing software component ESA GS Ordering & Programming Prototype [RD-06].	D	This is by definition: same Software Name, it is just another version.
SR-OPGW-DI-020	OPGW development activities shall be carried out according ESA ECSS standards [AD-04] with the tailoring specified in the HMA FO Task 4 Proposal [AD-08].	D	It is verified by checking whether the delivered documents are compliant with the ECSS standards [AD-04]: <ul style="list-style-type: none"> <li>• SRS, for requirements</li> <li>• SDD, for design</li> <li>• SVS, for test plan &amp; procedures</li> </ul>
SR-OPGW-PB-001	The OPGW shall be developed using open source COTS whenever possible.	D	The verification will be done by checking the OPGW functions w.r.t. the implementation.  Main items to be verified: <ul style="list-style-type: none"> <li>• Application server</li> <li>• RDBMS</li> <li>• XML processor</li> <li>• Etc.</li> </ul>
SR-OPGW-QA-001	The development of the OPGW component shall follow ECSS standards [AD-04] with the tailoring specified in the HMA FO Task 4 proposal [AD-08].	D	Same as SR-OPGW-DI-020

Identifier	Description	Verification Method	Inspection Activity
SR-OPGW-QA-010	<p>The OPGW component will be provided with the following documentation:</p> <ul style="list-style-type: none"> <li>▪ Software Requirement Specification Document (this document)</li> <li>▪ Software Design Document</li> <li>▪ Software Validation Test Specification</li> </ul>	D	To check whether the listed documents are delivered with OPGW software.
SR-OPGW-DD-001	The OPGW should be based on HSQLDB RDBMS.	I	During the OPGW installation to check whether the RDBMS is actually managed via HSQLDB or not.
SR-OPGW-RM-001	<p><b>Nominal</b> availability of OPGW shall be 24 hours a day, 7 days a week to external users.</p> <p>Comment: This means that no operation must require shutdown of the service (unless for re-configuration / software update).</p>	D	It will be verified during the operational life of the OPGW system.

Table 9-1: Inspection procedures.

## 10 OPGW Validation test platform requirements

### 10.1 Test Platform

The execution of test requires theoretically two workstations:

- one for the OPGW under test;
- one for the test client of OPGW interfaces.

In this way the operation environment of the system where the main client, the SSE, runs on a remote machine is simulated.

However if two machines are not available, only one workstation can be sufficient for running both the OPGW and the test client.

The Hardware and software requirements for OPGW component are specified in [AD-07] and are reported hereafter:

Hardware resources:

- CPU: at least INTEL P4 2.4 GHz or equivalent;
- RAM: at least 1GB (1 GB is needed for ORACLE COTS)
- At least 600 MB of free disk space (to hold the OPGW & ORACLE WSM);
- Graphic Adapter: no specific needs.

Software resources:

- Operating System: Linux RedHat ES 5 Update 2 (being a Java development, the system can be built and operated also on Windows XP OS).
- Tomcat 6
- Ant 1.6.5
- Java 5
- XML Beans 2.2.0
- HSQLDB 1.8.0.7

For the client machine the following resources are recommended:

Hardware resources:

- any PC;

Software resources:

- Operating System: any Linux or Windows XP.
- Java 5
- Google earth

#### 10.1.1 Network Architecture

For carrying out the acceptance test activities the following network requirements have to be complied:

- The server / workstation hosting the ESA User Services shall be accessible from the OPGW then the ESA firewall shall be configured accordingly.
- The client machine shall be allowed to connect to the OPGW. Because the OPGW is possibly listening to non standard ports, then even this point has to be verified on the network configuration where the client machine is connected to.

#### 10.1.2 Test tool

As test tool will be used the OPGW built-in test tool for sending SOAP requests from HTML page.

## 11 Requirements verification matrix

For each requirement it is defined the verification method using the classification in §5.4.

### 11.1 Requirements vs. Test cases traceability matrix

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-FN-001	The OPGW shall allow its clients to submit product orders including archived order items.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-001	The OPGW shall allow its clients to submit product orders including archived order items.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-FN-005	The OPGW shall allow its clients to submit product orders including subscription.  Note: With HMA-E the ESA GS is able to support on-line subscriptions, then OPGW will rely on this function.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-005	The OPGW shall allow its clients to submit product orders including subscription.  Note: With HMA-E the ESA GS is able to support on-line subscriptions, then OPGW will rely on this function.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-FN-010	The access to Ordering functionality shall be provided by a Gateway which translates incoming HMA Ordering request in EOLI XML requests and sends them to the ESA GS EOLI XML Ordering Server.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-010	The access to Ordering functionality shall be provided by a Gateway which translates incoming HMA Ordering request in EOLI XML requests and sends them to the ESA GS EOLI XML Ordering Server.	T	TC_010_040: Ordering Service - GetStatus
SR-OPGW-FN-010	The access to Ordering functionality shall be provided by a Gateway which translates incoming HMA Ordering request in EOLI XML requests and sends them to the ESA GS EOLI XML Ordering Server.	T	TC_020_001: Ordering & Programming Service – Coverage Order

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-FN-020	OPGW shall support ordering of products from the following missions / sensors: <ul style="list-style-type: none"> <li>• Envisat ASAR</li> <li>• Envisat MERIS</li> <li>• SPOT</li> </ul>	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-FN-020	OPGW shall support ordering of products from the following missions / sensors: <ul style="list-style-type: none"> <li>• Envisat ASAR</li> <li>• Envisat MERIS</li> <li>• SPOT</li> </ul>	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-020	OPGW shall support ordering of products from the following missions / sensors: <ul style="list-style-type: none"> <li>• Envisat ASAR</li> <li>• Envisat MERIS</li> <li>• SPOT</li> </ul>	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-FN-030	OPGW shall support ordering of products returned by the External OGC Catalogue configured in the same HMA Follow-on reference environment.  Note: A companion HMA Catalogue must be set-up in order to allow the discovery and then the ordering of products.	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-FN-030	OPGW shall support ordering of products returned by the External OGC Catalogue configured in the same HMA Follow-on reference environment.  Note: A companion HMA Catalogue must be set-up in order to allow the discovery and then the ordering of products.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-500	OPGW shall accept requests including the SAML token as specified in the DAIL UM ICD document.	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-FN-500	OPGW shall accept requests including the SAML token as specified in the DAIL UM ICD document.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-500	OPGW shall accept requests including the SAML token as specified in the DAIL UM ICD document.	T	TC_020_001: Ordering & Programming Service – Coverage Order

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-FN-510	OPGW shall extract user profile information from the SAML Token embedded in HMA requests.	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-FN-510	OPGW shall extract user profile information from the SAML Token embedded in HMA requests.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-510	OPGW shall extract user profile information from the SAML Token embedded in HMA requests.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-FN-520	OPGW shall verify the signature embedded in the received SAML Token.	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-FN-520	OPGW shall verify the signature embedded in the received SAML Token.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-520	OPGW shall verify the signature embedded in the received SAML Token.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-FN-530	OPGW shall decrypt SAML assertions included in the SAML Token sent from the HMA Client.	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-FN-530	OPGW shall decrypt SAML assertions included in the SAML Token sent from the HMA Client.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-FN-530	OPGW shall decrypt SAML assertions included in the SAML Token sent from the HMA Client.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-FN-540	In order that the HMA request may be fulfilled by the ESA ground segment, the ESA user, to which the HMA request is targeted to, shall have the grants for: querying and ordering (past and future) at least the following product types: <ul style="list-style-type: none"> <li>• querying and ordering (past) Envisat ASAR products</li> <li>• querying and ordering (past) Envisat MERIS products</li> <li>• querying and ordering (past) SPOT products</li> </ul>	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-IF-001	The OPGW shall provide access to the Ordering Functionality of ESA MM GS through interfaces compliant with the HMA Ordering ICD.  Note: See the following requirements for the list of supported functions and restrictions.	T	TC_010_030: Ordering Service - Submit

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-IF-001	<p>The OPGW shall provide access to the Ordering Functionality of ESA MM GS through interfaces compliant with the HMA Ordering ICD.</p> <p>Note: See the following requirements for the list of supported functions and restrictions.</p>	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-IF-010	<p>To support HMA Ordering ICD, the OPGW shall implement the following operations:</p> <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_001: Ordering Service - GetCapabilities
SR-OPGW-IF-010	<p>To support HMA Ordering ICD, the OPGW shall implement the following operations:</p> <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-IF-010	<p>To support HMA Ordering ICD, the OPGW shall implement the following operations:</p> <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_020: Ordering Service - GetQuotation

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_030: Ordering Service - Submit
SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_040: Ordering Service - GetStatus
SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_050: Ordering Service - SubmitResponse
SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_060: Ordering Service - Cancel



Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_010_070: Ordering Service - DescribeResultAccess
SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T	TC_020_001: Ordering & Programming Service - Coverage Order
SR-OPGW-IF-020	OPGW shall be able to provide order notification by calling the SubmitResponse operation implemented by the client (i.e. the entity that called the Submit operation).	T	TC_010_050: Ordering Service - SubmitResponse
SR-OPGW-IF-030	The Capabilities data returned by GetCapabilities operation shall be extracted from the repository of OPGW configuration data.	T	TC_010_001: Ordering Service - GetCapabilities
SR-OPGW-IF-040	The order options stored in OPGW shall be generated by converting the ServiceDirectory.xml file (managed by EOLI SA / EOLI Server).  The ServiceDirectory.xml file has to be manually provided by the operator. <ul style="list-style-type: none"> <li>▪ OPGW will answer to GetOptions request by accessing to the locally stored converted file.</li> </ul>	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-IF-050	GetQuotation operation shall be implemented by calling the built-in testing capabilities of the HMA Skeleton.	T	TC_010_020: Ordering Service - GetQuotation

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-IF-060	Submit operation shall be implemented by calling the processProductOrderRequest EOLI XML operation of EOLI XML Order Server mapping the HMA Ordering request and response messages on the EOLI XML request and response messages according to the table at §9.2.4	T	TC_010_030: Ordering Service - Submit
SR-OPGW-IF-060	Submit operation shall be implemented by calling the processProductOrderRequest EOLI XML operation of EOLI XML Order Server mapping the HMA Ordering request and response messages on the EOLI XML request and response messages according to the table at §9.2.4	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-IF-070	In Submit operation the following values of HMA Ordering/Submit/statusNotification attribute shall be supported for the management of asynchronous notifications: <ul style="list-style-type: none"> <li>• None</li> <li>• Final</li> </ul> It is desirable to support “all” value.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-IF-070	In Submit operation the following values of HMA Ordering/Submit/statusNotification attribute shall be supported for the management of asynchronous notifications: <ul style="list-style-type: none"> <li>• None</li> <li>• Final</li> </ul> It is desirable to support “all” value.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-IF-080	For providing order status notification after order submission, the OPGW should poll the EOLI XML Order Server to get the order status.	T	TC_010_050: Ordering Service - SubmitResponse
SR-OPGW-IF-090	GetStatus operation shall be implemented by calling the processOrderMonitorRequest operation of EOLI XML Order Server mapping the HMA Ordering request and response messages on the EOLI XML request.	T	TC_010_040: Ordering Service - GetStatus
SR-OPGW-IF-100	Cancel operation shall be implemented by calling the built-in testing capabilities of the HMA Skeleton.	T	TC_010_060: Ordering Service - Cancel
SR-OPGW-IF-110	DescribeResultAccess operation shall be implemented by calling the built-in testing capabilities of the HMA Skeleton.	T	TC_010_070: Ordering Service - DescribeResultAccess
SR-OPGW-OP-001	The OPGW shall provide the operator with an interface allowing checking logs of received requests.	T	TC_010_030: Ordering Service - Submit

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-OP-001	The OPGW shall provide the operator with an interface allowing checking logs of received requests.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-OP-010	The configuration data needed to the OPGW shall be kept within ASCII files.	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-OP-020	The OPGW shall log received request and provided responses.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-OP-020	The OPGW shall log received request and provided responses.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-PR-001	In case of requests which are managed by a straightforward translation between the HMA and EOLI XML protocol i.e.: <ul style="list-style-type: none"> <li>Ordering::Submit</li> <li>Ordering::GetStatus</li> </ul> the response time shall not be greater than: <ul style="list-style-type: none"> <li>Total Response Time &lt; 5 seconds + Time Spent by ESA GS to process the EOLI requests.</li> </ul>	T	TC_010_030: Ordering Service - Submit
SR-OPGW-PR-001	In case of requests which are managed by a straightforward translation between the HMA and EOLI XML protocol i.e.: <ul style="list-style-type: none"> <li>Ordering::Submit</li> <li>Ordering::GetStatus</li> </ul> the response time shall not be greater than: <ul style="list-style-type: none"> <li>Total Response Time &lt; 5 seconds + Time Spent by ESA GS to process the EOLI requests.</li> </ul>	T	TC_010_040: Ordering Service - GetStatus
SR-OPGW-PR-001	In case of requests which are managed by a straightforward translation between the HMA and EOLI XML protocol i.e.: <ul style="list-style-type: none"> <li>Ordering::Submit</li> <li>Ordering::GetStatus</li> </ul> the response time shall not be greater than: <ul style="list-style-type: none"> <li>Total Response Time &lt; 5 seconds + Time Spent by ESA GS to process the EOLI requests.</li> </ul>	T	TC_020_001: Ordering & Programming Service – Coverage Order

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-PR-010	<p>In case of requests completely managed by OPGW (also by simulation):</p> <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> <p>the response time shall not be greater than 5 seconds.</p>	T	TC_010_001: Ordering Service - GetCapabilities
SR-OPGW-PR-010	<p>In case of requests completely managed by OPGW (also by simulation):</p> <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> <p>the response time shall not be greater than 5 seconds.</p>	T	TC_010_010: Ordering Service - GetOption
SR-OPGW-PR-010	<p>In case of requests completely managed by OPGW (also by simulation):</p> <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> <p>the response time shall not be greater than 5 seconds.</p>	T	TC_010_020: Ordering Service - GetQuotation
SR-OPGW-PR-010	<p>In case of requests completely managed by OPGW (also by simulation):</p> <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> <p>the response time shall not be greater than 5 seconds.</p>	T	TC_010_060: Ordering Service - Cancel

Identifier	Description	Verification Method	Test Case Identifier
SR-OPGW-PR-010	In case of requests completely managed by OPGW (also by simulation): <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> the response time shall not be greater than 5 seconds.	T	TC_010_070: Ordering Service - DescribeResultAccess
SR-OPGW-RM-010	OPGW shall reject syntactically invalid requests.	T	TC_010_030: Ordering Service - Submit
SR-OPGW-RM-010	OPGW shall reject syntactically invalid requests.	T	TC_020_001: Ordering & Programming Service – Coverage Order
SR-OPGW-RS-001	The OPGW shall run on x86 PC with Linux Red Hat ES 5.2 OS.	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-RS-010	The OPGW shall run on PC with 1GB RAM	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-RS-020	The OPGW source code should be written in Java (as implied by the Skeleton).	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-RS-030	The OPGW shall run on PC with Apache Tomcat (as implied by the Skeleton).	T	TC_001_001: OPGW Installation, Configuration, Start-up
SR-OPGW-SP-001	The access to OPGW resources HMA Order Service shall be allowed only to registered HMA Clients.	T	TC_010_030: Ordering Service - Submit

Table 11-1: Requirements vs. Test Cases traceability matrix.

## 11.2 Test cases vs. Requirements traceability matrix

Test Case Identifier	Identifier	Description	Verification Method
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-FN-540	In order that the HMA request may be fulfilled by the ESA ground segment, the ESA user, to which the HMA request is targeted to, shall have the grants for: querying and ordering (past and future) at least the following product types: <ul style="list-style-type: none"> <li>querying and ordering (past) Envisat ASAR products</li> <li>querying and ordering (past) Envisat MERIS products</li> <li>querying and ordering (past) SPOT products</li> </ul>	T
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-IF-040	The order options stored in OPGW shall be generated by converting the ServiceDirectory.xml file (managed by EOLISA / EOLI Server).  The ServiceDirectory.xml file has to be manually provided by the operator. <ul style="list-style-type: none"> <li>OPGW will answer to GetOptions request by accessing to the locally stored converted file.</li> </ul>	T
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-OP-010	The configuration data needed to the OPGW shall be kept within ASCII files.	T
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-RS-001	The OPGW shall run on x86 PC with Linux Red Hat ES 5.2 OS.	T
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-RS-010	The OPGW shall run on PC with 1GB RAM	T
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-RS-020	The OPGW source code should be written in Java (as implied by the Skeleton).	T
TC_001_001: OPGW Installation, Configuration, Start-up	SR-OPGW-RS-030	The OPGW shall run on PC with Apache Tomcat (as implied by the Skeleton).	T

Test Case Identifier	Identifier	Description	Verification Method
TC_010_001: Ordering Service - GetCapabilities	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T
TC_010_001: Ordering Service - GetCapabilities	SR-OPGW-IF-030	The Capabilities data returned by GetCapabilities operation shall be extracted from the repository of OPGW configuration data.	T
TC_010_001: Ordering Service - GetCapabilities	SR-OPGW-PR-010	In case of requests completely managed by OPGW (also by simulation): <ul style="list-style-type: none"> <li>• Ordering::GetCapabilities</li> <li>• Ordering::GetOptions</li> <li>• Ordering::GetQuotation</li> <li>• Ordering::Cancel</li> <li>• Ordering::DescribeResultAccess</li> </ul> the response time shall not be greater than 5 seconds.	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-FN-020	OPGW shall support ordering of products from the following missions / sensors: <ul style="list-style-type: none"> <li>• Envisat ASAR</li> <li>• Envisat MERIS</li> <li>• SPOT</li> </ul>	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-FN-030	OPGW shall support ordering of products returned by the External OGC Catalogue configured in the same HMA Follow-on reference environment.  Note: A companion HMA Catalogue must be set-up in order to allow the discovery and then the ordering of products.	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-FN-500	OPGW shall accept requests including the SAML token as specified in the DAIL UM ICD document.	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-FN-510	OPGW shall extract user profile information from the SAML Token embedded in HMA requests.	T

Test Case Identifier	Identifier	Description	Verification Method
TC_010_010: Ordering Service - GetOption	SR-OPGW-FN-520	OPGW shall verify the signature embedded in the received SAML Token.	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-FN-530	OPGW shall decrypt SAML assertions included in the SAML Token sent from the HMA Client.	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>GetCapabilities</li> <li>GetOptions</li> <li>GetQuotation</li> <li>Submit &amp; SubmitResponse</li> <li>GetStatus</li> <li>Cancel &amp; CancelResponse</li> <li>DescribeResultAccess</li> </ul>	T
TC_010_010: Ordering Service - GetOption	SR-OPGW-PR-010	In case of requests completely managed by OPGW (also by simulation): <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> the response time shall not be greater than 5 seconds.	T
TC_010_020: Ordering Service - GetQuotation	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>GetCapabilities</li> <li>GetOptions</li> <li>GetQuotation</li> <li>Submit &amp; SubmitResponse</li> <li>GetStatus</li> <li>Cancel &amp; CancelResponse</li> <li>DescribeResultAccess</li> </ul>	T
TC_010_020: Ordering Service - GetQuotation	SR-OPGW-IF-050	GetQuotation operation shall be implemented by calling the built-in testing capabilities of the HMA Skeleton.	T



Test Case Identifier	Identifier	Description	Verification Method
TC_010_020: Ordering Service - GetQuotation	SR-OPGW-PR-010	In case of requests completely managed by OPGW (also by simulation): <ul style="list-style-type: none"> <li>Ordering::GetCapabilities</li> <li>Ordering::GetOptions</li> <li>Ordering::GetQuotation</li> <li>Ordering::Cancel</li> <li>Ordering::DescribeResultAccess</li> </ul> the response time shall not be greater than 5 seconds.	T
TC_010_030: Ordering Service - Submit	SR-OPGW-FN-001	The OPGW shall allow its clients to submit product orders including archived order items.	T
TC_010_030: Ordering Service - Submit	SR-OPGW-FN-005	The OPGW shall allow its clients to submit product orders including subscription.  Note: With HMA-E the ESA GS is able to support on-line subscriptions, then OPGW will rely on this function.	T
TC_010_030: Ordering Service - Submit	SR-OPGW-FN-010	The access to Ordering functionality shall be provided by a Gateway which translates incoming HMA Ordering request in EOLI XML requests and sends them to the ESA GS EOLI XML Ordering Server.	T
TC_010_030: Ordering Service - Submit	SR-OPGW-FN-020	OPGW shall support ordering of products from the following missions / sensors: <ul style="list-style-type: none"> <li>Envisat ASAR</li> <li>Envisat MERIS</li> <li>SPOT</li> </ul>	T
TC_010_030: Ordering Service - Submit	SR-OPGW-FN-030	OPGW shall support ordering of products returned by the External OGC Catalogue configured in the same HMA Follow-on reference environment.  Note: A companion HMA Catalogue must be set-up in order to allow the discovery and then the ordering of products.	T
TC_010_030: Ordering Service - Submit	SR-OPGW-FN-500	OPGW shall accept requests including the SAML token as specified in the DAIL UM ICD document.	T

Test Case Identifier	Identifier	Description	Verification Method
TC_010_030: Service - Submit	Ordering SR-OPGW-FN-510	OPGW shall extract user profile information from the SAML Token embedded in HMA requests.	T
TC_010_030: Service - Submit	Ordering SR-OPGW-FN-520	OPGW shall verify the signature embedded in the received SAML Token.	T
TC_010_030: Service - Submit	Ordering SR-OPGW-FN-530	OPGW shall decrypt SAML assertions included in the SAML Token sent from the HMA Client.	T
TC_010_030: Service - Submit	Ordering SR-OPGW-IF-001	The OPGW shall provide access to the Ordering Functionality of ESA MM GS through interfaces compliant with the HMA Ordering ICD.  Note:  See the following requirements for the list of supported functions and restrictions.	T
TC_010_030: Service - Submit	Ordering SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T
TC_010_030: Service - Submit	Ordering SR-OPGW-IF-060	Submit operation shall be implemented by calling the processProductOrderRequest EOLI XML operation of EOLI XML Order Server mapping the HMA Ordering request and response messages on the EOLI XML request and response messages according to the table at §9.2.4	T
TC_010_030: Service - Submit	Ordering SR-OPGW-IF-070	In Submit operation the following values of HMA Ordering/Submit/statusNotification attribute shall be supported for the management of asynchronous notifications: <ul style="list-style-type: none"> <li>• None</li> <li>• Final</li> </ul> It is desirable to support "all" value.	T
TC_010_030: Service - Submit	Ordering SR-OPGW-OP-001	The OPGW shall provide the operator with an interface allowing checking logs of received requests.	T
TC_010_030: Service - Submit	Ordering SR-OPGW-OP-020	The OPGW shall log received request and provided responses.	T

Test Case Identifier	Identifier	Description	Verification Method
TC_010_030: Ordering Service - Submit	SR-OPGW-PR-001	In case of requests which are managed by a straightforward translation between the HMA and EOLI XML protocol i.e.: <ul style="list-style-type: none"> <li>Ordering::Submit</li> <li>Ordering::GetStatus</li> </ul> the response time shall not be greater than: <ul style="list-style-type: none"> <li>Total Response Time &lt; 5 seconds + Time Spent by ESA GS to process the EOLI requests.</li> </ul>	T
TC_010_030: Ordering Service - Submit	SR-OPGW-RM-010	OPGW shall reject syntactically invalid requests.	T
TC_010_030: Ordering Service - Submit	SR-OPGW-SP-001	The access to OPGW resources HMA Order Service shall be allowed only to registered HMA Clients.	T
TC_010_040: Ordering Service - GetStatus	SR-OPGW-FN-010	The access to Ordering functionality shall be provided by a Gateway which translates incoming HMA Ordering request in EOLI XML requests and sends them to the ESA GS EOLI XML Ordering Server.	T
TC_010_040: Ordering Service - GetStatus	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>GetCapabilities</li> <li>GetOptions</li> <li>GetQuotation</li> <li>Submit &amp; SubmitResponse</li> <li>GetStatus</li> <li>Cancel &amp; CancelResponse</li> <li>DescribeResultAccess</li> </ul>	T
TC_010_040: Ordering Service - GetStatus	SR-OPGW-IF-090	GetStatus operation shall be implemented by calling the processOrderMonitorRequest operation of EOLI XML Order Server mapping the HMA Ordering request and response messages on the EOLI XML request.	T
TC_010_040: Ordering Service - GetStatus	SR-OPGW-PR-001	In case of requests which are managed by a straightforward translation between the HMA and EOLI XML protocol i.e.: <ul style="list-style-type: none"> <li>Ordering::Submit</li> <li>Ordering::GetStatus</li> </ul> the response time shall not be greater than: <ul style="list-style-type: none"> <li>Total Response Time &lt; 5 seconds + Time Spent by ESA GS to process the EOLI requests.</li> </ul>	T

Test Case Identifier	Identifier	Description	Verification Method
TC_010_050: Ordering Service - SubmitResponse	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T
TC_010_050: Ordering Service - SubmitResponse	SR-OPGW-IF-020	OPGW shall be able to provide order notification by calling the SubmitResponse operation implemented by the client (i.e. the entity that called the Submit operation).	T
TC_010_050: Ordering Service - SubmitResponse	SR-OPGW-IF-080	For providing order status notification after order submission, the OPGW should poll the EOLI XML Order Server to get the order status.	T
TC_010_060: Ordering Service - Cancel	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T
TC_010_060: Ordering Service - Cancel	SR-OPGW-IF-100	Cancel operation shall be implemented by calling the built-in testing capabilities of the HMA Skeleton.	T
TC_010_060: Ordering Service - Cancel	SR-OPGW-PR-010	In case of requests completely managed by OPGW (also by simulation): <ul style="list-style-type: none"> <li>• Ordering::GetCapabilities</li> <li>• Ordering::GetOptions</li> <li>• Ordering::GetQuotation</li> <li>• Ordering::Cancel</li> <li>• Ordering::DescribeResultAccess</li> </ul> the response time shall not be greater than 5 seconds.	T

Test Case Identifier	Identifier	Description	Verification Method
TC_010_070: Ordering Service - DescribeResultAccess	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T
TC_010_070: Ordering Service - DescribeResultAccess	SR-OPGW-IF-110	DescribeResultAccess operation shall be implemented by calling the built-in testing capabilities of the HMA Skeleton.	T
TC_010_070: Ordering Service - DescribeResultAccess	SR-OPGW-PR-010	In case of requests completely managed by OPGW (also by simulation): <ul style="list-style-type: none"> <li>• Ordering::GetCapabilities</li> <li>• Ordering::GetOptions</li> <li>• Ordering::GetQuotation</li> <li>• Ordering::Cancel</li> <li>• Ordering::DescribeResultAccess</li> </ul> the response time shall not be greater than 5 seconds.	T
TC_020_001: Ordering & Programming Service - Coverage Order	SR-OPGW-FN-001	The OPGW shall allow its clients to submit product orders including archived order items.	T
TC_020_001: Ordering & Programming Service - Coverage Order	SR-OPGW-FN-005	The OPGW shall allow its clients to submit product orders including subscription.  Note: With HMA-E the ESA GS is able to support on-line subscriptions, then OPGW will rely on this function.	T
TC_020_001: Ordering & Programming Service - Coverage Order	SR-OPGW-FN-010	The access to Ordering functionality shall be provided by a Gateway which translates incoming HMA Ordering request in EOLI XML requests and sends them to the ESA GS EOLI XML Ordering Server.	T
TC_020_001: Ordering & Programming Service - Coverage Order	SR-OPGW-FN-020	OPGW shall support ordering of products from the following missions / sensors: <ul style="list-style-type: none"> <li>• Envisat ASAR</li> <li>• Envisat MERIS</li> <li>• SPOT</li> </ul>	T

Test Case Identifier	Identifier	Description	Verification Method
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-FN-500	OPGW shall accept requests including the SAML token as specified in the DAIL UM ICD document.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-FN-510	OPGW shall extract user profile information from the SAML Token embedded in HMA requests.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-FN-520	OPGW shall verify the signature embedded in the received SAML Token.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-FN-530	OPGW shall decrypt SAML assertions included in the SAML Token sent from the HMA Client.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-IF-001	The OPGW shall provide access to the Ordering Functionality of ESA MM GS through interfaces compliant with the HMA Ordering ICD.  Note:  See the following requirements for the list of supported functions and restrictions.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-IF-010	To support HMA Ordering ICD, the OPGW shall implement the following operations: <ul style="list-style-type: none"> <li>• GetCapabilities</li> <li>• GetOptions</li> <li>• GetQuotation</li> <li>• Submit &amp; SubmitResponse</li> <li>• GetStatus</li> <li>• Cancel &amp; CancelResponse</li> <li>• DescribeResultAccess</li> </ul>	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-IF-060	Submit operation shall be implemented by calling the processProductOrderRequest EOLI XML operation of EOLI XML Order Server mapping the HMA Ordering request and response messages on the EOLI XML request and response messages according to the table at §9.2.4	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-IF-070	In Submit operation the following values of HMA Ordering/Submit/statusNotification attribute shall be supported for the management of asynchronous notifications: <ul style="list-style-type: none"> <li>• None</li> <li>• Final</li> </ul> It is desirable to support “all” value.	T

Test Case Identifier	Identifier	Description	Verification Method
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-OP-001	The OPGW shall provide the operator with an interface allowing checking logs of received requests.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-OP-020	The OPGW shall log received request and provided responses.	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-PR-001	In case of requests which are managed by a straightforward translation between the HMA and EOLI XML protocol i.e.: <ul style="list-style-type: none"> <li>Ordering::Submit</li> <li>Ordering::GetStatus</li> </ul> the response time shall not be greater than: <ul style="list-style-type: none"> <li>Total Response Time &lt; 5 seconds + Time Spent by ESA GS to process the EOLI requests.</li> </ul>	T
TC_020_001: Ordering & Programming Service – Coverage Order	SR-OPGW-RM-010	OPGW shall reject syntactically invalid requests.	T

**Table 11-2: Test Cases vs. Requirements traceability matrix**