Open Geospatial Consortium, Inc.

Date: 2010-08-18

Reference number of this document: OGC 10-127r1

Category: OGC Public Engineering Report

Editor: Roger Brackin

OGC® OWS-7 Engineering Report - Aviation Portrayal

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type: OpenGIS® Engineering Report
Document subtype: NA
Document stage: Approved for public release
Document language: English

Preface

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

Contents

Page

1 Introduction 1
1.1 Scope 1
1.2 Document contributor contact points 1
1.3 Revision history 1
1.4 Future work 1
1.5 Forward 2
2 References 2
3 Conventions 2
3.1 Abbreviated terms 2
3.2 UML notation 2
4 Design of the Portrayal Service Chain 3
4.1 Architecture of Portrayal in OWS-7 3
4.2 Portrayal Services Interaction Design 4
4.2.1 Interaction 1: Access via Client Specified SLD to FPS Dynamically: 5
4.2.2 Interaction 2: Pre-built WMS Layers (FPS interrogates the Registry) 6
4.2.3 Interaction 3: Access via Client Applying and SLD to WFS Output: 6
4.3 Portrayal Registry Design 7
4.3.1 Description of Classes of Artifacts 9
5 Implementation 10
6 Implementation of the FPS Specification 10
6.1 Integration Experiments 11
6.1.1 Galdos CSW to Envitia FPS, Luciad, Frequentis Clients using SLD 11
6.1.2 Snowflake WFS, Comsoft WFS, MetoFrance WFS to Envitia FPS 11
6.1.3 Snowflake WFS, Comsoft WFS, MeteoFrance WFS to Alticode FPS 11
6.1.4 Alticode FPS access from Luciad and Frequentis Clients using SLD 11
6.2 Examples of Portrayal 11
6.3 Styling Examples from the Envitia FPS 11
6.4 Portrayal Rules for Coverage Data 13
6.5 FPS and WMS Exploitation in the Luciad Client 15
6.6 FPS and WMS Exploitation in the Frequentis Client 15
7 Functionality and Interoperability Issues Identified 16
7.1 Specialist Geometry Portrayal 16
7.2 Separation of SLD and SE Encoding 16
7.3 Efficiency 17
7.4 Notification Schemes 17
7.5 Open Standards Portrayal of Meteorological Data 17
7.6 WFS 2.0 Changes 18
7.7 Complexity of the AIXM/WXXM Models 18
7.8 Bio-diversity in WFS Implementations 18

8 Conclusion 18
Annex A - Comparison of the OWS-7 and DGIWG Approach 20

Figures

Page

Figure 1 - FPS Sub-system Architecture **4**
Figure 2 - Interaction 1 Sequence Diagram **5**
Figure 3 - Interaction 2 Sequence Diagram **6**
Figure 4 - Interaction 3 Sequence Diagram **7**
Figure 5 - Portrayal Registry Artifacts **8**
Figure 6 Raw WMS Image, and in situ over Imagery (using WMS Chaining) **13**
Figure 7 - Different Portrayal Outputs **13**
Figure 8 - Portrayal of Gridded Coverage Data **14**
Figure 9 - Luciad Client - Electronic Flightbag **15**
Figure 10 - Frequentis Aviation Client **16**

OGC® OWS-7 Engineering Report - Aviation Portrayal

1 Introduction

1.1 Scope

This document describes the requirements, design, technical implementation and technology trialed for the Feature Portrayal service chain used in OWS-7. This includes the interfaces to the OWS Data Services deployed, the feature portrayal servers, the interfaces to clients and the registry information model and interface.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Roger Brackin	Envitia (Editor)
Robin Houtmeyers	Luciad (Contributor)
Bruno Simmenauer	Alticode (Contributor)

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2010-06-08	Draft 1	Roger Brackin		Draft Release
2010-8-10	Public ER	Carl Reed	Various	Ready for public release.

1.4 Future work

Improvements in this document are desirable to include experiences from OGC Member Altcode in producing a Component FPS from scratch from the specification. More detailed contributions from the Clients in using FPS Services and the merits of this against direct use of SLDs. Issues of caching and potential notification schemes in supporting the maintenance of up-to-date caches without polling.

1.5 Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OpenGIS® Web Services Common Standard*

OGC 05-078r4, *Styled Layer Descriptor profile of the Web Map Service Implementation Specification*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

3 Conventions

3.1 Abbreviated terms

The following terms and abbreviations are used in this document:

CS/W - Catalogue Service for the Web

DGIWG - Defence Geospatial Information Working Group

FPS - Feature Portrayal Service

OWS-7 - OGC Web Services Trial 7.

WFS - Web Feature Service

3.2 UML notation

Most diagrams that appear in this document are presented using the Unified Modeling Language (UML) static structure diagrams and sequence diagrams, as described in Subclause 5.2 of [OGC 06-121r3].

4 Design of the Portrayal Service Chain

The following sections describe the feature portrayal service chain produced using standard OGC service compliant components within OWS-7. The service chain consists of a registry, a number of feature portrayal servers (with differing strengths and focuses) and clients. It is supported by a series of Web Feature Services and Web Coverage Services providing data into the portrayal service chain. It is also supported by the design of a portrayal registry within CS/W ebRIM which supports the processing chain.

The main goal of the demonstration architecture is to demonstrate how a portrayal infrastructure could deliver a range of capabilities from different services, and how the portrayal itself can be managed. The OWS-7 feature portrayal subsystem was a serious attempt to show integration of data services (both vector and gridded), portrayal services, Symbol/Style Registries and Clients. As such it required the design of a portrayal registry model (as the most likely candidates from OGC and DGIWG for this are still embryonic) as well as dealing with the issues of multi-vendor source and multi-vendor client integration. Key elements were to show how the architecture could offer significant benefits to the user, in particular separating data delivery and portrayal and then describing the available portrayals in a registry.

The idea was to do this exploiting existing technology as much as possible, but show how the components could be assembled in a subsystem which provided flexibility and could be optimized. The relevant standards and their limitations in supporting this were considered.

The primary design of the sub-system used is described in Ref [1]. This document outlines some of the issues identified in assembling the sub-system. The FPS model seems to operate effectively and address a real set of use cases, in particular the need to have a clear separation of data and portrayal. It also seems to allow a significant level of support for legacy clients. Various orchestration options, (including providing a 'fake service' where the application reads what seems like a standard WMS with fixed layers, but is accessing a FPS looking at a registry for its portrayal) allow different levels of capability to suit different clients, while still keeping the flexibility appears to be valuable.

4.1 Architecture of Portrayal in OWS-7

The general computational architecture used as the basis for experimentation in OWS-7 is shown below. This shows all of the key components of the sub-system.

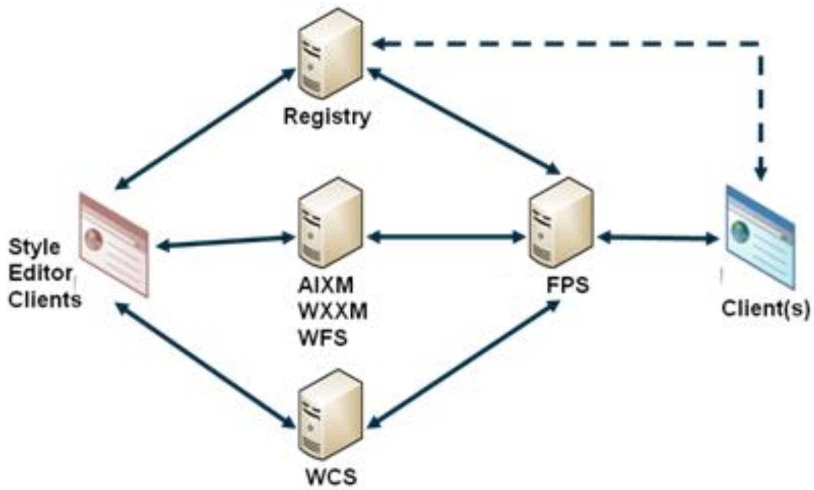

Figure 1 - FPS Sub-system Architecture

With the exception of styling rules for the WCS portrayal, all other component interfaces can be built with existing OGC Services and transactional data transfer formats. This has been proved in the OWS-7 Experimentation. The expectation is that some form of coverage portrayal rule set could be used, but is not defined at present. To demonstrate coverage data (primarily sourced from the 4-D Weather Data Cube), specific portrayal was set up on fixed layers in the FPS (more like traditional WMS, but still external to the data delivery service rather than integrated). There is as yet no standard defining the portrayal of this, but it is a clear target for the SLD SWG. This report contains some recommendations in terms of requirements for such an SLD (and Coverage Portrayal Service).

The implementation of the above was demonstrated using the following components from the specified service providers:

- Web Feature Services (WFS) – Snowflake, ComSoft
- Web Coverage Services (WCS) - NNEW
- Registry (CS/W) – GALDOS Indicio
- Feature Portrayal Services (FPS) – Envitia, Altcode
- Clients – Luciad, Frequentis

4.2 Portrayal Services Interaction Design

The potential portrayal interaction models for the Portrayal Service Chain are shown below. The models addressed are:

- Client Interacts with the FPS to apply its own SLD.
- FPS retrieves published layer requirements from the registry and publishes layers.
- Client retrieves the SLD and applies it internally to generate the portrayal.

4.2.1 Interaction 1: Access via Client Specified SLD to FPS Dynamically:

This interaction is the most complicated in terms of interaction.

1. A system user selects AIXM from the list of products they want to display.
2. The particular use they are interested in involves styling the data in a specific way (for example to support planning an air operation avoiding commercial airspace). This requires a specific styling.
3. The user may locate the relevant dataset from the registry, and then 'ask' for the available portrayals.
4. Specific SLDs relevant to the dataset chosen can be discovered from the registry and the user picks one.
5. The SLD is passed to the FPS as a URL (it is a repository item in the registry). This references symbols/styles which also have URLs in the registry.
6. The portrayed result is presented to the user.

The above sequence is shown below in UML.

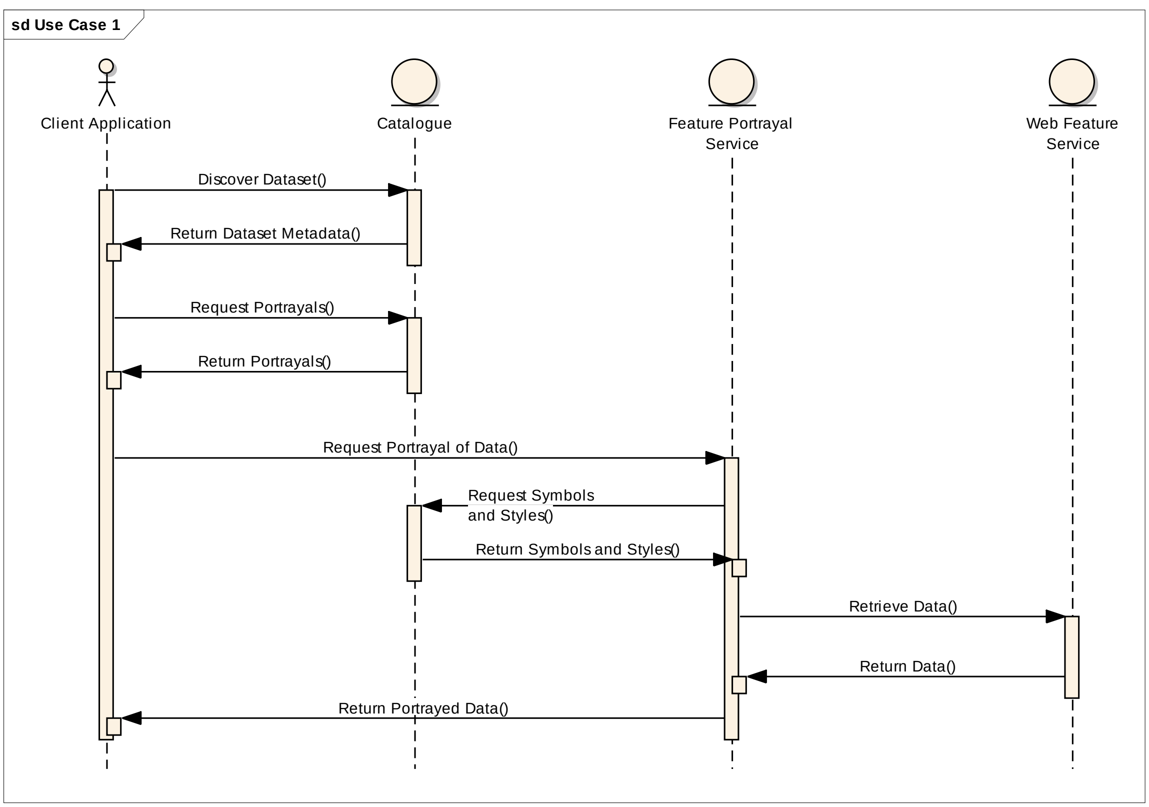

Figure 2 - Interaction 1 Sequence Diagram

4.2.2 Interaction 2: Pre-built WMS Layers (FPS interrogates the Registry)

The FPS is responsible for the portrayal and the client interface is simplified.

1. The registry defines the portrayals relevant to a given product.
2. The Feature Portrayal Service is pointed to given registry dataset and builds and caches the portrayals defined by the SLDs related to the dataset.
3. The user's client performs a "GetCapabilities" and the portrayal options are visible as layers in the FPS.

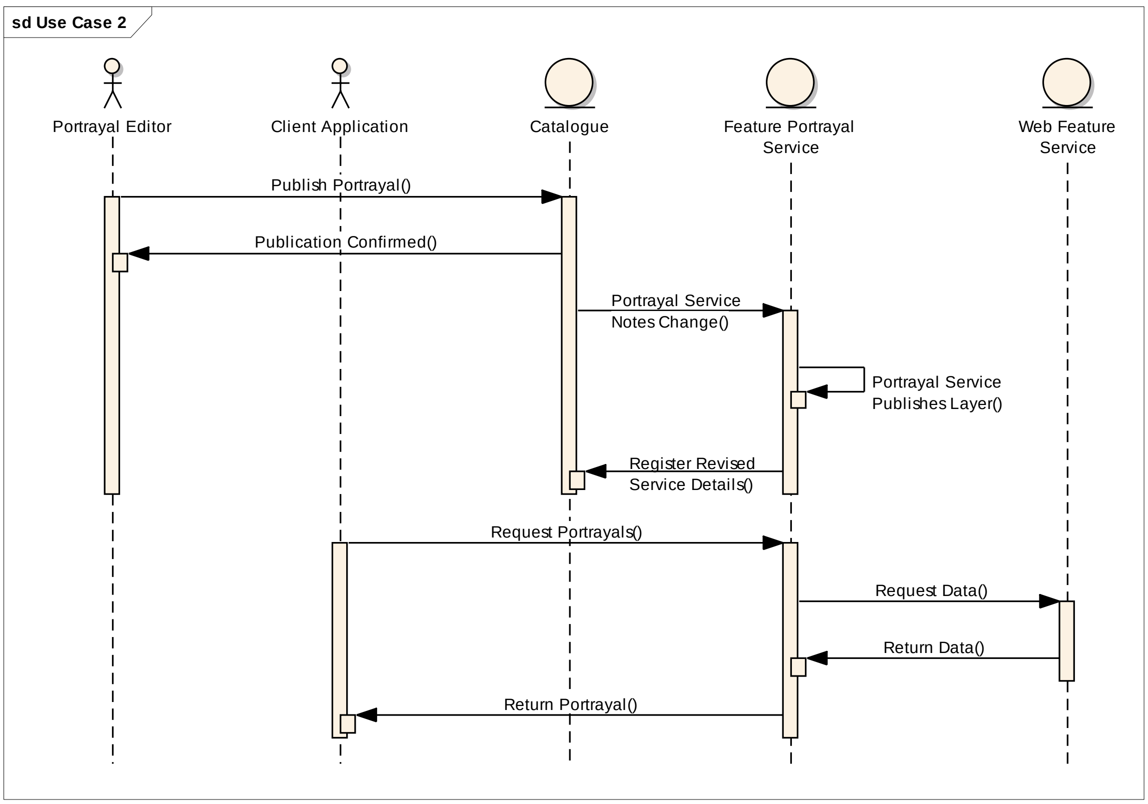

Figure 3 - Interaction 2 Sequence Diagram

4.2.3 Interaction 3: Access via Client Applying and SLD to WFS Output:

In this use case the client needs to perform the portrayal.

4. A system user selects AIXM from the list of products they want to display.
5. The particular use they are interested in involves styling the data in a specific way (for example to support planning an air operation avoiding commercial airspace). This requires a specific styling.

6. The user may locate the relevant dataset from the registry, and then 'ask' for the available portrayals.

7. Specific SLDs relevant to the dataset chosen can be discovered from the registry and the user picks one.

8. The client retrieves the SLD from the registry and uses it to portray the data. The SLD references symbols/styles which also have URLs in the registry. The client retrieves these as necessary.

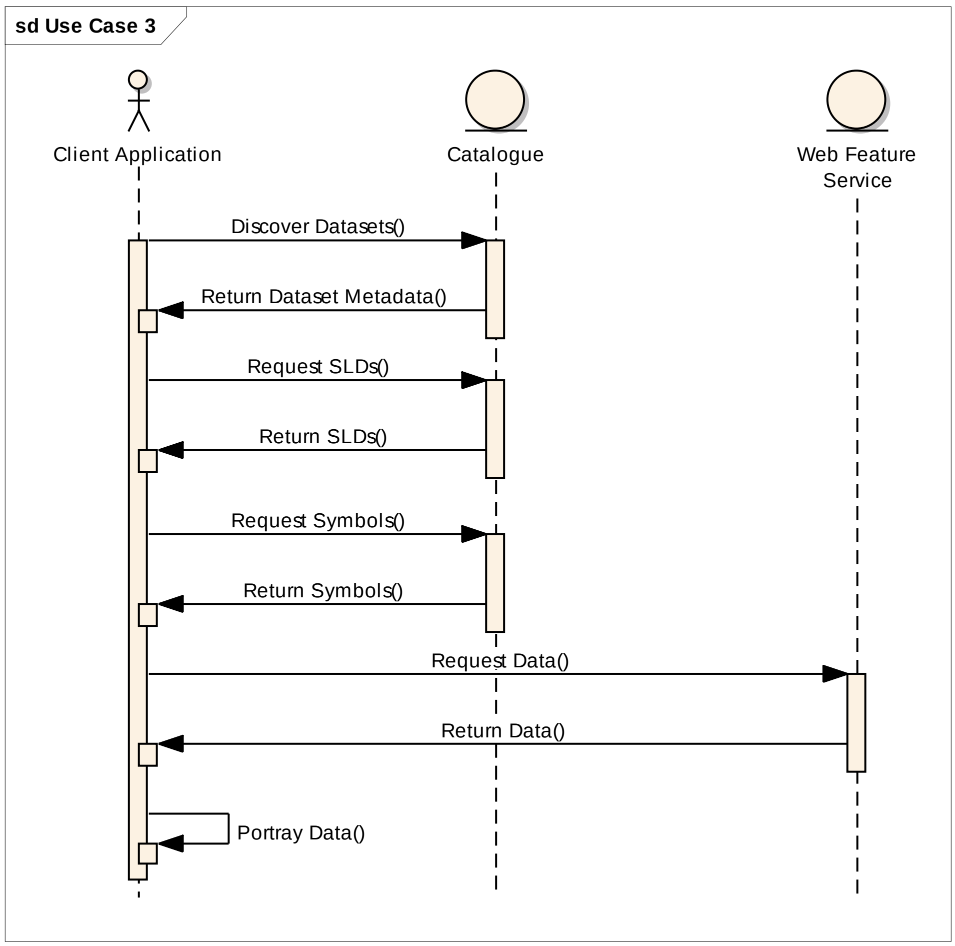

Figure 4 - Interaction 3 Sequence Diagram

4.3 Portrayal Registry Design

The UML below suggests a basic set of model elements which would be needed to support SLD Based portrayal from the registry, focused on definition in ebRIM. The design is purely a prototype to inform more formal design.The only ebRIM artefact in the diagram below that at present has an OGC standard definition is Dataset (present in the OGC ebRIM Basic Package and used in the CSW CIM Extension Package).

The distinction between registry and repository is not made above. The presumption is that each of these artifacts will have a registry and a repository item. Internally these repository items, the SLD for example, will reference a Repository item by URL (e.g the Symbol) using standard XML referencing.

Is there specific value is realizing those relationships as ebXML Associations, at the Registry level? In practice all that is needed in the above are associations to the ProductType from dataset and then to SLD.

A more granular approach to the above would be to define Style fragments, defined per feature type, and then collect these together. This then requires there to be a feature catalogue, the complexity starts to grow. However there is nothing stopping this being developed later or extending the model.

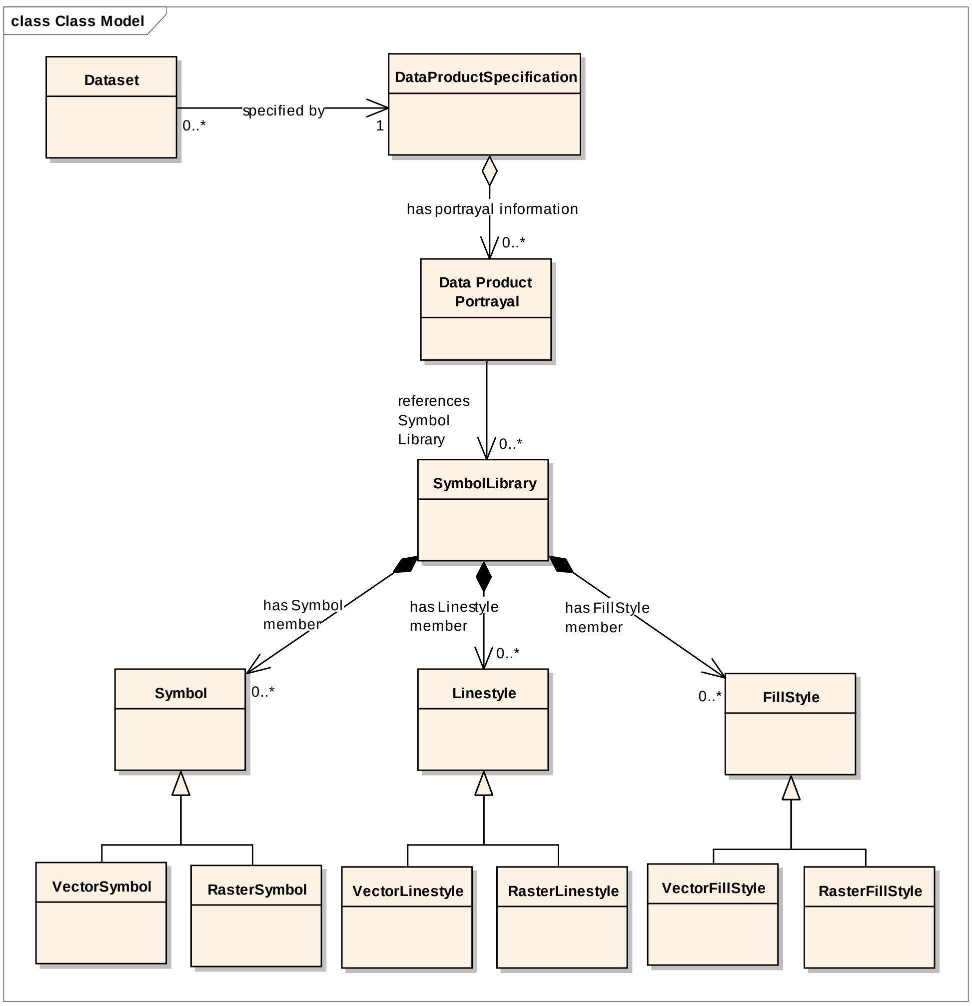

Figure 5 - Portrayal Registry Artifacts

4.3.1 Description of Classes of Artifacts

Class	Description	ebRIM Implementation
Dataset	Represents a geospatial dataset, referencing metadata, services etc which relate to it.	Extrinsic Object, Source: CSW, Basic Package.
DataProductSpecification	Object which represents the definition of a data product defining the contents and character of a dataset.	Extrinsic Object, classified as a 'Data Product Specification'.
DataProductPortrayal	Represents a collection of styling rules against a set of feature types from a given product specification.	Extrinsic Object, classified as a 'Data Product Specification'. Repository item will hold an SLD Document.
SymbolLibrary	Represents the collection of symbols, linestyles and fillstyles which describe a symbol library.	Extrinsic Object, classified by 'SymbolLibrary'. Repository Item could be a GML Dictionary listing all of the Symbols in the Library.
Symbol	Abstract class for a symbol definition	Node in a classification scheme.
VectorSymbol	Class representing a vector symbol, encoded in SVG.	Extrinsic object, with the repository item containing the symbol definition in XML
RasterSymbol	Class representing a raster symbol, encoded in bitmap form.	Extrinsic object, with the repository item containing the symbol definition in raster form
Linestyle	Abstract class for a linestyle definition	Node in a classification scheme.
VectorLinestyle	Class representing a vector linestyle, encoded in SVG.	Extrinsic object, with the repository item containing the symbol definition in XML
RasterLinestyle	Class representing a raster linestyle, encoded in bitmap form.	Extrinsic object, with the repository item containing the linestyle definition in raster form
FillStyle	Abstract class for a fillstyle definition	Node in a classification scheme.
VectorFillstyle	Class representing a vector fillstyle, encoded in SVG.	Extrinsic object, with the repository item containing the symbol definition in XML
RasterFillstyle	Class representing a raster fillstyle, encoded in bitmap form.	Extrinsic object, with the repository item containing the fillstyle definition in raster form

Table 1 – Portrayal Registry Artefact Descriptions

One of the key external stakeholders in the area of Portrayal Registries is DGIWG. They are building a portrayal registry model which embraces NATO and other Symbology and provides a register of these. DGIWG agreed to review the portrayal registry design of OWS-7 and their comments are detailed in Annex A.

5 Implementation

The UML model will be implemented as a set of ebRIM Classes, Associations and Classifications (within the 'Type' Classification Scheme) which can be loaded into a CSW-ebRIM registry.

The symbol library itself was implemented sticking to Raster Symbols/Styles/Fills as this is the most generally supported mode. Association classes for Symbol, LineStyle and FillStyle are not relevant as this stage.

The specific references within the data product portrayal document (styled layer descriptor) reference the specific symbols. All that is required at a registry level is to identify the symbol/style library required, potentially to allow broad dependencies to be identified.

The class Data Product Portrayal could be sub-classed too, as potentially there are other definition forms than the SLD. For the purposes of OWS-7 it is probable that this will be limited to SLD.

In terms of use case, Feature Portrayal Services can potentially support use case 1 and 2 (with use case 3 being focused on clients capable of interpreting SLDs such as the Luciad Client, and/or clients capable of accessing a CS/W). Use-case 1 requires the client to be able to issue a request with an SLD. It is also potentially a requirement that it retrieves the SLD from the registry. It is possible that the SLDs could be simply passed to and held locally in the clients (avoiding the need for a registry interface), but reference central symbols in the symbol library. In this case there would be no need for the clients to interact with the registry at all. In addition client access to a registry URL looks like any other URL so clients only need registry access to search for SLDs, not to access them.

Given the above, it is only be necessary to implement the classes under 'Symbol Library' for OWS-7 trials as DataProductSpecification and DataProductPortrayal are only required to support discovery of portrayal definitions (as opposed to symbols and styles referenced by those portrayals).

6 Implementation of the FPS Specification

The FPS model seems to operate effectively and addresses a real set of use cases, in particular the need to have a clear separation of data and portrayal. It also seems to allow a significant level of support for legacy clients. Various orchestration options, (including providing a 'fake service' where the application reads what seems like a standard WMS with fixed layers, but is accessing a FPS looking at a registry for its portrayal) allow different levels of capability to suit different clients, while still keeping the flexibility appears to be valuable.

6.1 Integration Experiments

The integration of the FPS to the Web Feature Servers proved relatively easy. The AIXM Schema is relatively complicated and one issue is the time taken to download or even interpret the schema if there no a-priori knowledge of its content exists, but it is GML compliant and therefore readable in capable WFS Clients. The complexity of the schemas makes references to parts of it in the SLD rules relatively unattractive (but this is not too relevant to end users).

6.1.1 Galdos CSW to Envitia FPS, Luciad, Frequentis Clients using SLD

The FPS correctly read portrayal information from the GALDOS Registry. In fact the requirement was for all clients to do this, but we don't believe any other specific implementations have been tried at this point. Clients which can reference an SLD by URL (e.g Luciad and Frequentis) can implicitly access the registry to display the results (as the SLD in the registry looks like a simple URL). This was, it is believed tried and successful.

6.1.2 Snowflake WFS, Comsoft WFS, MetoFrance WFS to Envitia FPS

The Envitia FPS successfully read the Snowflake WFS (AIXM), Comsoft WFS (AIXM) and Meteo France WFS (WXXM) contents and render them effectively. No significant issues were identified here, and the result was credibly rendered AIXM/WXXM. The only issue identified was the potential latency and load from an un-cached component FPS due to the load added to WFSs. Thus it was mostly operated in cached mode.

6.1.3 Snowflake WFS, Comsoft WFS, MeteoFrance WFS to Alticode FPS

The Alticode FPS was developed from specification and implemented the standard as described, providing a pure FPS chaining any incoming request to the WFS and implementing any SLD completely on demand. No caching was performed. This clearly worked correctly, and offers the maximum flexibility with a gaurentee of completely up-to-date data. As described below, this approach is hard to scale.

6.1.4 Alticode FPS access from Luciad and Frequentis Clients using SLD

The Alticode FPS was exercised by these clients successfully using a number of SLDs.

6.2 Examples of Portrayal

6.3 Styling Examples from the Envitia FPS

A number of examples of SLDs were produced. A typical query using an SLD is shown below. This invokes a query which colours the runways dependent on their length.

```
http://217.33.30.11/envitia.wms.ows7.aixm2/wms.aspx?
  REQUEST=GetMap&
  SERVICE=WMS&
  VERSION=1.3.0&
  FORMAT=image/png&
  BGCOLOR=0xFFFFFF&
  TRANSPARENT=TRUE&
  CRS=EPSG:4326&
  BBOX-97.3703,32.809,-97.3516,32.8317&
  WIDTH=968&HEIGHT=843&
  LAYERS=Layers.9ADDBBB9&
  SLD=http://registry.galdosinc.com/ows7/query?
      request=GetRepositoryItem
      &id=urn:envitia:com:sld:runway_seg_len_long
```

The actual SLD Contents includes the fragment to style the SLD as shown below.

```
<sld:Rule>
  <ogc:Filter>
    <ogc:PropertyIsGreaterThanOrEqualTo>
      <ogc:PropertyName>
        timeSlice/RunwayElementTimeSlice/length
      </ogc:PropertyName>
      <ogc:Literal>2133.6</ogc:Literal>
    </ogc:PropertyIsGreaterThanOrEqualTo>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>
      timeSlice/RunwayElementTimeSlice/length/@uom
      </ogc:PropertyName>
      <ogc:Literal>M</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <sld:PolygonSymbolizer>
    <sld:Geometry>
      <ogc:PropertyName>geometry</ogc:PropertyName>
    </sld:Geometry>
    <sld:Fill>
      <sld:CssParameter name="fill">#ffff66</sld:CssParameter>
    </sld:Fill>
    <sld:Stroke>
      <sld:CssParameter name="stroke-width">1.0</sld:CssParameter>
      <sld:CssParameter name="stroke">#ff6600</sld:CssParameter>
    </sld:Stroke>
  </sld:PolygonSymbolizer>
</sld:Rule>
```


This styles any runway longer than 2133.6 metres as yellow, otherwise leaving it as grey. The result of the query is as shown below using the Envitia FPS, together with a second portrayal including a NASA Pan Sharpened Global view Imagery WMS as a background.

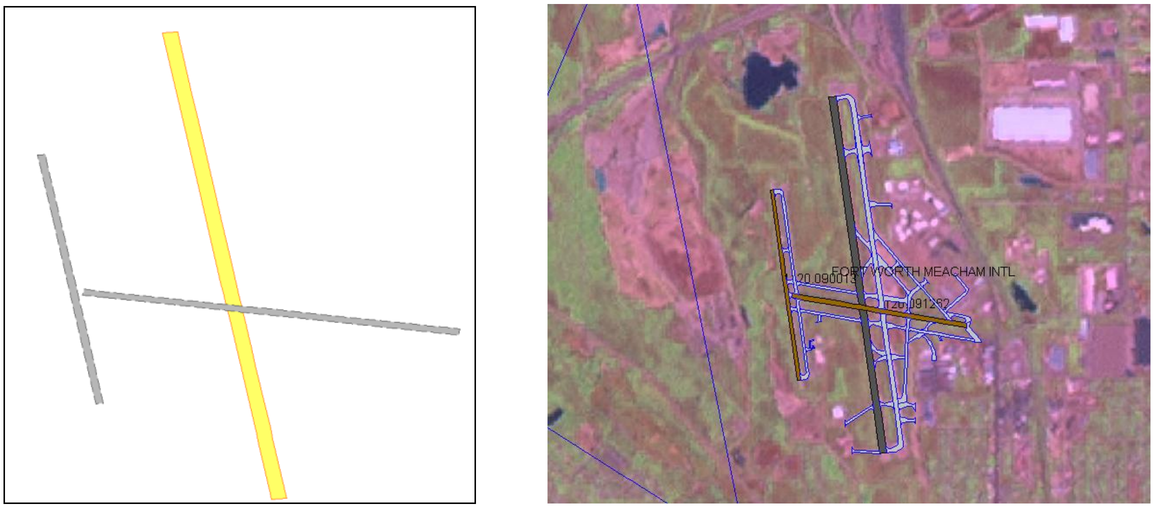

Figure 6 Raw WMS Image, and in situ over Imagery (using WMS Chaining)

The two examples below show different portrayal rules applied to runway and obstructions. In the two cases shown the runway is first styled as a single colour and in the second case if the surface type is concrete, tarmac or a loose surface. In addition towers are shown, and set to yellow if they have a light on them.

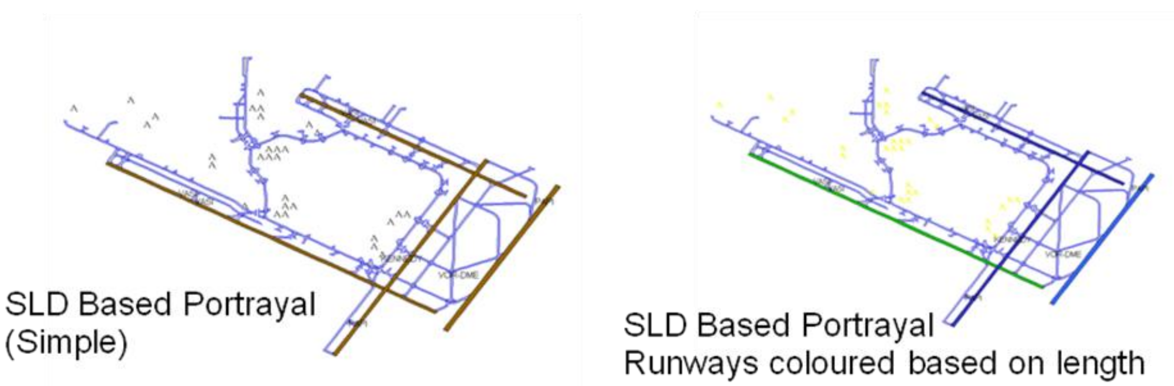

Figure 7 - Different Portrayal Outputs

6.4 Portrayal Rules for Coverage Data

At present no well defined model exists for specifying a coverage portrayal. Envitia's FPS does include the ability to define rules for portrayal but these are defined in a proprietary language (XML based description. A series of such rules were defined to allow the portrayal of coverages and applied. A series of services were therefore deployed which portray gridded (4D Weather Data Cube) data. An example of such a portrayal is shown below (with the time axis accessible).

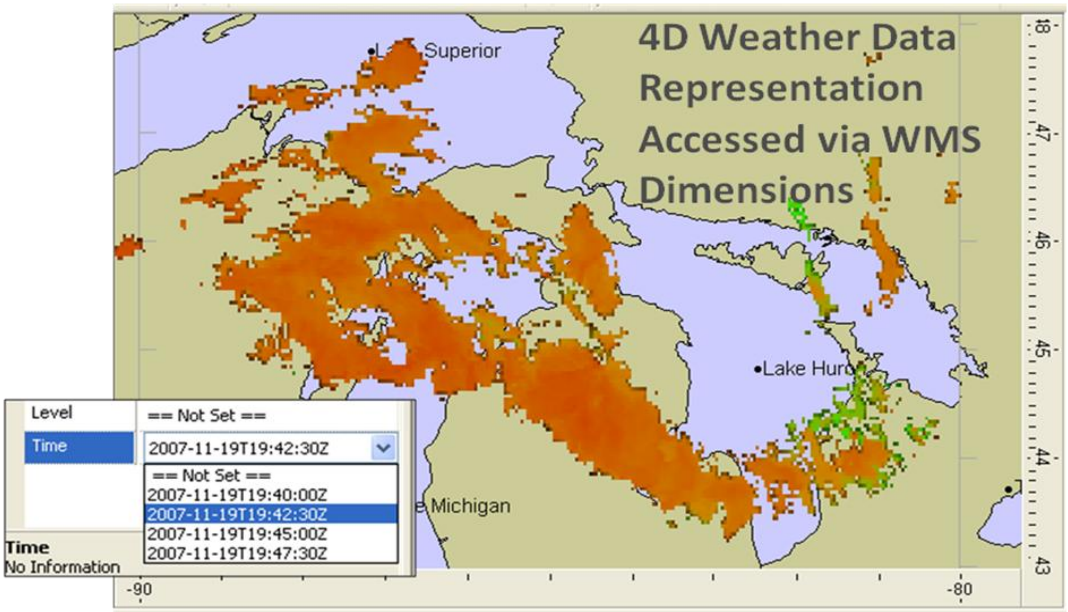

Figure 8 - Portrayal of Gridded Coverage Data

The presentation requires a number of complex transformations:

- The coverage data supplied via the 4D Weather Data Cube server is provided in projected form. It is necessary for the FPS to translate the coordinate system in order to present it in a lat/long WPS as required by the clients.
- The coverage data covers a range of times. The FPS therefore needs to propagate these through as WMS dimensions (there are both level and time dimensions as well as the X-Y dimensions on the data making it 4D nature.
- The actual portrayal needs to be defined, It could be, for example contours, colour banding (as shown), wind or current arrows, pure text or a range of other presentations. No well defined specification exists for such coverage portrayal. Envitia have a proprietary encoding in XML which describes it and this was used. The goal should be to develop and test such an encoding if such data is to be exploited in an open way.

The nature of the above processes though, and the typical size of coverages, make ‘on the fly’ generation difficult and so caching is again important. However the problem with Met data is easier than with other data as the update time is often well defined (you can predict when new data will be available.

6.5 FPS and WMS Exploitation in the Luciad Client

The picture below shows the Luciad Client exploiting the Envitia FPS to show the 4D Weather Data Cube, and the Meteo France FPS showing volcanic Ash coverage. The planned route is also shown.

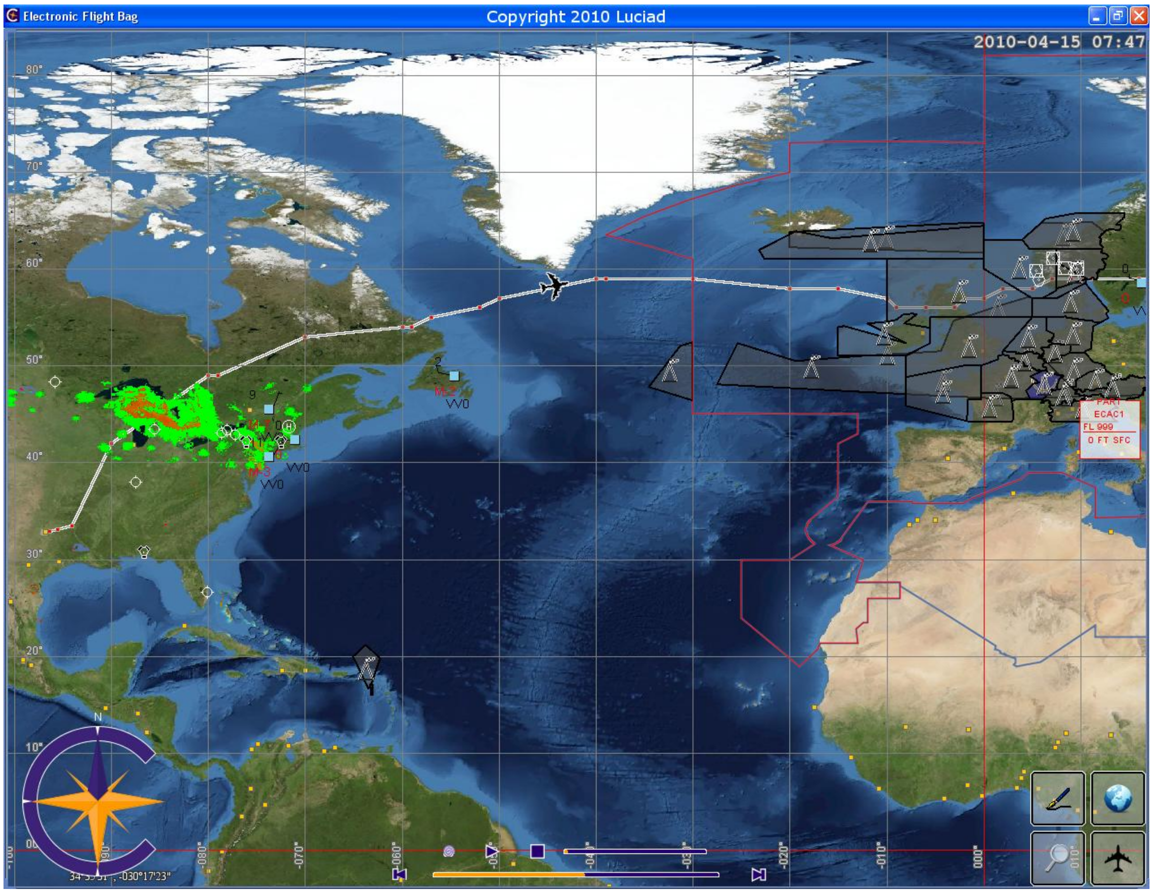

Figure 9 - Luciad Client - Electronic Flightbag

Luciad also provided a prototype FPS for experimentation purposes. In addition the Luciad client was able to directly access the SLDs and portray the features directly, but accessing the FPS has some merits, and allowed verification that different portrayal methods were consistent.

6.6 FPS and WMS Exploitation in the Frequentis Client

The Frequentis Client also exploited the FPS capabilities. The client used the Open-Layers toolkit to access the FPSs as well as providing a range of aeronautical capabilities (Route display, event handling etc. The figures below show the client accessing the FPS for background map, 4d Weather Data Cube Data and detailed runway portrayal.

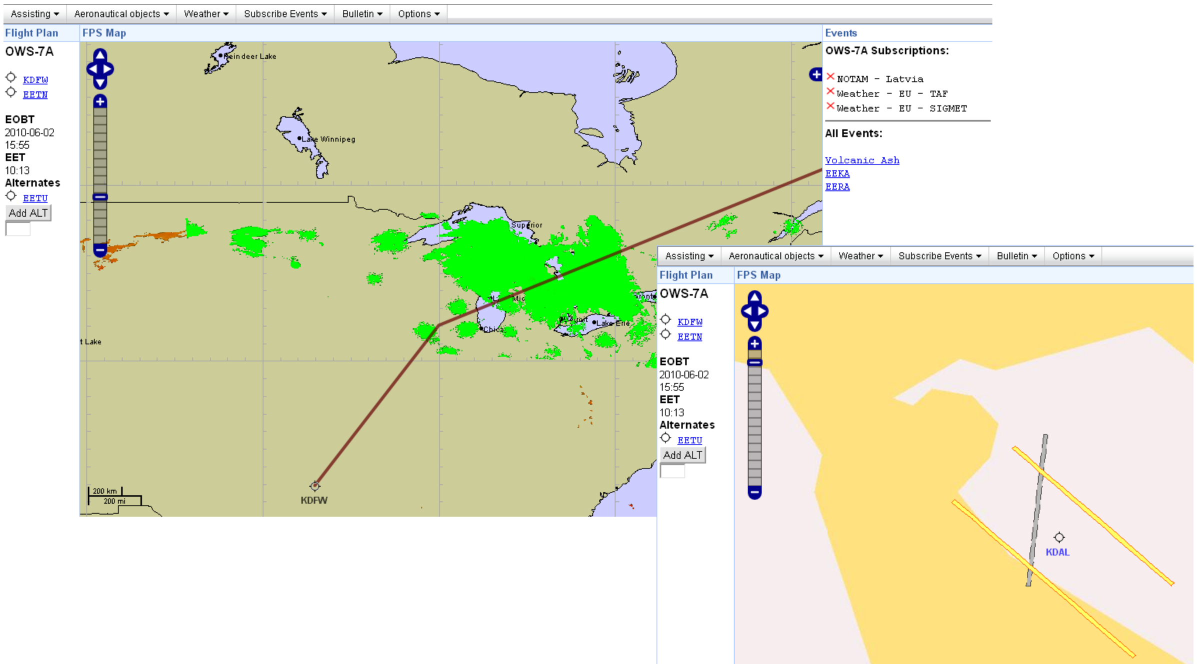

Figure 10 - Frequentis Aviation Client

7 Functionality and Interoperability Issues Identified

The following are some of the issues from OWS-7 portrayal experiments which either need standards revision or further experimentation to identify the detail.

7.1 Specialist Geometry Portrayal

Specialist geometry is often present in aeronautical products. This often requires custom renderers and/or converters with domain specific knowledge. The rendering can be dependent on the aircraft characteristics (e.g turning rate). One approach is to pre-render specialist geometries in an 'informative' form in the data itself. This makes generic portrayal possible (Envitia/Galdos pioneered this approach within the NATO Core GIS Data Preparation Project). A second approach is to include specialist rules invoking custom portrayal in the SLD. This is possible within the revision of ISO19117 but is not yet available in SLD. This issue applies more generally. If a portrayal engine is capable of more accurate representation of symbology (better than say the bitmap provided in the SLD) it should be able to detect from the SLD that the intention is that this is used, but the bitmap represents an acceptable alternative. This allows high capability clients to draw items perfectly, but generic clients also to support portrayal at a reduced capability.

7.2 Separation of SLD and SE Encoding

The SLD SWG separated the component of the SLD related to the service connection from the SLD describing portrayal (the later known as SE). However it is impossible for the FPS to exploit this separation as it only accepts an SLD document via the command. A significant problem with this is that a different SLD is needed for each and every service (since the URLs are included in the SLD). This significantly limits the flexibility of an SLD in the registry and duplicates content in a number of SLDs. As a result the capability to supply an SLD in-line using the HTTP Post method was examined (which means the SLD is retrieved from the registry, processed to change addresses and then passed in-line. One problem with this relates to performance, identified below, but it also puts an unnecessary burden on the client.

7.3 Efficiency

A significant issue is efficiency. If a feature portrayal server is hitting a Web Feature Server for all rendering, then firstly the load on the WFS will be significant, as will be the processing load in the FPS. In addition the most significant issue is latency (time from request to response). Because of the chain of FPS to WFS in a pure 'Component' FPS the time to respond to an FPS getMap will be significant. One approach to mitigating this is to support caching in the FPS. The key issue with this is making sure the cache remains current. To do this some form of subscription notification and change detection in the WFS is necessary.

In addition it is desirable that the FPS has a-priori knowledge of the SLDs it will have to render in order to ensure it doesn't have to prepare a new rendering every time. The registry is helpful here. If the FPS is aware that an SLD passed is a registry reference (an ebRIM repository item URL) it knows this will not change (the registry change procedure would allocate a new URL for another version). As a result it can cache the SLD and avoid repeated load. This optimization is not possible with an in-line SLD.

7.4 Notification Schemes

There seems potentially to be some merit in investigating the opportunity for the FPS to pick up change events from a WFS. This would allow it to pick up changes and update the cache, or re-query when necessary without polling. However this requires both the WFS infrastructure and FPS to support this. It also may only be necessary on very dynamic layers or ones where currency is critical. Given though that eventing is a well developed concept in the Aviation area it seems worth investigating.

7.5 Open Standards Portrayal of Meteorological Data

Clearly coverage data, and meteorological data coverages in particular, require some work to allow open standards compliant FPSs to portray them using open definitions. In addition some of the portrayals are complicated and computationally expensive (e.g. contouring) and so caching is necessary to achieve any reasonable level of performance.

The SLD SWG have been looking at this, and obviously an extension to SLD would be one way. In addition the WCPS standard could play a part in some of the portrayal options. Further investigation is needed in this area.

7.6 WFS 2.0 Changes

There have been a range of WFS Systems supporting WFS 1.1 and 2.0. In reality few problems were found using the new WFS specification. The addition of SOAP on some services also was reasonably easy to handle given it is well defined. However neither detailed security capabilities or optimizations that the WFS 2.0 specification offers were really exploited. The idea of stored queries has some merit, particularly in the context of FPS, but no time was available to experiment with this. This should be carried forward as an activity. Snowflake have provided some examples, and Luciad and Envitia have indicated that they are prepared to experiment once core OWS-7 activities are complete.

7.7 Complexity of the AIXM/WXXM Models

It is clear that AIXM does include some very complex information modeling, exploiting many of the features of GML. The schemas, focused on formal and complete modeling of the problem domain, do not exactly play to populist requirements. While they may be exactly what is needed for the aeronautical community, the ability to integrate them into more general use cases where the clients are less tuned to the application area is a problem. For example, even in OWS-7 special steps had to be taken to ensure geometries could be visualized by all clients (removing complete arcs from geometries). In addition the deep nesting, referencing, time-slicing make the data complex to interpret. In general the WFS assessors (FPS and Client providers) managed to read the data, but not without some beads of sweat on their brows.

7.8 Bio-diversity in WFS Implementations

While there is clearly a level of variation in capability, standard compliance and support etc in FPS, the fact is that this is relatively new. There still seems to be a wide range of variation in standards implementation across the 6 or 7 WFSs used in OWS-7, ranging from those offering a broad range of options and solid compliance (for example Snowflake) and other robust services from Meteo France and Comsoft through to more marginal services capabilities. But even within the more robust, compliant players there was still significant variability. It was more plug and pray than plug and play. We believe that the issues (which do relate in many cases to issues with the standards and the compliance test effectiveness) need to be discussed in detail.

8 Conclusion

The design and implementation of the OWS-7 Feature Portrayal Service Chain was achievable using existing standards, i.e. the standards and available technologies did cover the bulk of the requirements. A number of specific issues were identified which need further experimentation/review in SWGs. These include:

- Separation of URLs and styling such that an SLD can be passed to an FPS and separately the URLs.

- Implementation of Coverage Styling Definition as an Open Standard and the development of the CPS (Coverage Portrayal Service) further.
- A more extensive review of the security issues. Portrayal limited by security rules for example only allowing access to specific capabilities of the FPS.
- Consideration of WFS Interoperability (detail) issues
- A deeper review of the detailed design of the registry, and the development of a working group focused on developing an ebRIM extension package to describe portrayal and symbol registration.

As an outcome though, it is believed that OWS-7 trials have proven that an effective and functional capability can be produced and would provide significant benefits.

Annex A - Comparison of the OWS-7 and DGIWG Approach

The comments below came from a DGIWG review of the original paper produced for OWS-7 suggesting the design for the Portrayal Registry. DGIWG are involved in a more extensive development, but were interested to review the proposed design for OWS-7 experimentation.

Comparison of the OWS-7/DGIWG Approach

Differences in approach between DGIWG Draft Documents and the OWS-7 document:

- OWS-7 is working with "symbol libraries" not individual rules as DGIWG plans to do. (There is a note about this being a possible extension in the section "Design of the Model in the Registry".)

- The OWS-7 registry is designed to be provided as a CSW service; DGIWG plans to do separate ebRIM-based interface using ebRS. The idea is to support a wider community and identifying that there is little that is truely ‘geospatial’ in relation to the information being held (SLDs, Symbol Libraries etc).

- DGIWG plans to support returning the style rules in multiple encodings, based on an internal ISO 19117-inspired data model.

- OWS-7 has modelled the Data Product Specification. We need something similar. Is this the right artefact? (DGIWG are inclined towards referencing a "Feature Catalogue" or "Application Schema" but it is not clear if there will be a registry of application schemas. DGIWG have concluded this is probably necessary so they can be authoritively referenced, even if they are produced by other communities.

- There is no UC corresponding to our "5.1 Using a Portrayal Registry with a ‘Component’ WMS Server" (where we pass in the portrayal registry info in the SLD UserStyle).

Comparison of the OWS-7/DGIWG Use Cases

DGIWG compared the use cases in the OWS-7 technical note with use cases they had prepared:

- UC1: Similar to the appendix in DGIWG DPRS spec. Client retrieves SLD, uses it in request to FPS. (Although in DGIWG solution client only receives "fragment" of an SLD.)

- UC2: Similar to "5.2 Using a Portrayal Registry with an ‘Integrated’ WMS Server" in DFPS spec, but the different portrayals are exposed as layers instead of named styles. Another difference compared with the DGIWG thoughts: Portrayal registry pushes changes to FPS. In our approach, the client includes the registry URL in the GetCapabilities request which causes the FPS to go and talk to the registry.

- UC3: Similar to UC1, but the client does the portrayal itself so there is no FPS involved. (But DGIWG agrees that it's good to have a separate use case for this; "shows that it works with these kinds of clients as well".)

- UC4: Registry maintenance. This will in the DGIWG case be done using the RegManTool (DGIWG are stakeholders in the development of this).

Other Observations from NGA

Specific comments were also made in relation to the document itself (comments made by a consultant to NGA).

- Observation: First, I am not clear on the aggregation between Data Product Specification and Data Product Portrayal. The way I understand the UML, a Data Product Specification has a number of Data Product Portrayals and because of the navigability of the association a Data Product Specification knows about all of its aggregate Data Product Portrayals but the Data Product Portrayals are ignorant of their aggregating Data Product Specification. This contradicts the approach we have been taking where the feature data (Dataset) and moreover the feature data specification (Data Product Specification) are ignorant of their portrayal. For us it is the rules which make the connection between data and symbols (portrayal). I would prefer the navigability of the association between Data Product Specification and Data Product Portrayal to be reversed and for it to be merely an association and not an aggregation.

- Response: The UML is used to indicate the conceptual character of the relationship not navigability. This is an issue around the actual use of UML.

- Observation: Second, it appears to me that the association between the Data Product Portrayal and the Symbol Library is an over simplification. Of course this may be intentional. But, since it is, in my understanding, the rules of the Data Product Portrayal which reference the Symbols, I would prefer to see the Data Product Portrayal composed of Portrayal Rules and then the Portrayal Rules referencing the Symbol/Linestyle/Fill Style classes, instead of the Data Product Portrayal referencing the Symbol Library.

- Response: This is intentional. The approach taken is to use the registry not as a formal definition but as a discovery tool. The artefacts stored in the repository form the normative components. The goal of say the link from SLD to Symbol library is to indicate which symbol libraries an SLD is using rather than to be the definition of the specific symbols used. This can be gleaned from the SLD itself.

In general the issues highlighted point to the distinct need for a well focussed and reviewed design for artifacts supporting specification within a registry. In particular feature types, application schemas, portrayal etc. This area is an obvious candidate for an ebRIM Extension package. The current CIM package only offers the equivalent of an ISO19115/119 profile. The work here has shown that an ebRIM registry is ideal for supporting the complex models required around portrayal but needs more agreement on the structural design to achieve interoperability.

