

# Open Geospatial Consortium, Inc.

Date: 2008-03-21

Reference number of this document: OGC 07-163

Version: 0.0.0

Category: Engineering Report

Editor(s): Stan Tillman, David Rosinger

## OGC<sup>®</sup> Data View Architecture Engineering Report

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

### Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore 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, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS <sup>®</sup> Discussion Paper
Document subtype:	NA
Document stage:	Approved for release
Document language:	English

## **Preface**

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email 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.

<b>Contents</b>		<b>Page</b>
1	Introduction.....	1
1.1	Scope .....	1
1.2	Document contributor contact points .....	1
1.3	Revision history.....	1
1.4	Future work .....	2
2	References.....	2
3	Terms and definitions .....	2
4	Conventions .....	3
4.1	Abbreviated terms .....	3
4.2	UML notation .....	4
5	Data View Architecture Overview.....	4
6	Data View Architecture .....	6
6.1	Data View Enterprise Viewpoint .....	6
6.2	Data View Information Viewpoint.....	8
6.2.1	UML Class Diagram of Data View Architecture .....	8
6.2.2	Data View XML Encoding .....	9
6.2.3	Modeling Data View Resources in ebRIM.....	13
6.2.4	Modeling Data View Resources in ebRIM, Harvesting Details .....	16
6.2.4.1	Harvesting of NAS QNames .....	16
6.2.4.2	Harvesting of Data Views .....	17
6.2.4.3	Harvesting of NAS/ISO-19115 Metadata .....	17
6.2.4.4	Harvesting a Web Feature Service .....	18
6.3	Data View Computational Viewpoint .....	18
6.3.1	Web Feature Service .....	18
6.3.2	ebRIM Catalog Service.....	18
6.4	Data View Engineering Viewpoint .....	19
6.4.1	Test Bed Data View Use Case .....	20
6.4.2	Client Request Messages Submitted to ebRIM Catalog .....	20
6.4.2.1	Find Data View Objects .....	21
6.4.2.2	Find Feature Types That Are Members of a Data View .....	23
6.4.2.3	Find Available Feature Type Instances That Are Members of a Data View .....	25
6.4.2.4	Find Feature Type Instances (General) .....	27
6.4.2.5	Find the WFS Endpoint which Operates On a Specified Feature Type Instance .....	28
6.4.3	Presentation of Data Views in the Integrated Client.....	29
	Bibliography .....	33

<b>Figures</b>	Page
<b>Figure 1 — Data View Architecture Overview .....</b>	<b>5</b>
<b>Figure 2 — Data View Use Case Diagram .....</b>	<b>7</b>
<b>Figure 3 — Data View Class Diagram.....</b>	<b>9</b>
<b>Figure 4 — Resources in the Data View Architecture .....</b>	<b>14</b>
<b>Figure 5 — Mapping Data View Artifacts to ebRIM, Part 1 .....</b>	<b>15</b>
<b>Figure 6 — Mapping Data View Artifacts to ebRIM, Part 2 .....</b>	<b>16</b>
<b>Figure 7 — Client Interaction with the Catalog .....</b>	<b>19</b>
<b>Figure 8 — Querying Data Views by Keyword .....</b>	<b>29</b>
<b>Figure 9 — Browsing a Selection of Data Views.....</b>	<b>30</b>
<b>Figure 10 — A Feature Type Can Be a Member of Multiple Data Views .....</b>	<b>31</b>
<b>Figure 11 — Selected Features Portrayed in Client.....</b>	<b>32</b>
<b>Figure 12 — Selected Features Grouped by Data View.....</b>	<b>32</b>

<b>Tables</b>	Page
<b>Table 1 — Data View: Agricultural Buildings and Structures .....</b>	<b>13</b>

# OGC® Data View Architecture Engineering Report

## 1 Introduction

### 1.1 Scope

This OGC document presents a summary of the Data View Architecture experiment conducted as part of the Geo-Processing Workflow (GPW) thread in the OWS-5 test bed. The main activities in this experiment were the storage of Data Views in an eBRIM Catalog and the discovery and use of those Data Views by an Integrated Client.

Although the Data View concept presented in this report is of general purpose in nature, the experiment conducted as part of the OWS-5 test bed investigated Data Views as applied specifically to NGA artifacts.

*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.*

### 1.2 Document contributor contact points

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

Name	Organization
Stan Tillman (editor)	Intergraph Corporation
David Rosinger (editor)	Intergraph Corporation
Philippe Duchesne	Leica Geosystems

### 1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2008-02-11	0.0.1	Tillman, Rosinger		First draft.
2008-03-21	1.0.0	Tillman, Rosinger	6.2, 6.3, 6.4	Initial release.

## 1.4 Future work

The participants in this study believe there is the potential to extend the Data View grouping mechanism to other types of data. Within the OWS-5 test bed, Data Views were used to group feature types presented by a Web Feature Service. Further work may be desirable to investigate the possibility of applying Data Views to other types of data such as layers published by a Web Map Service and observation offerings published by a Sensor Observation Service.

## 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, *OGC Web Services Common Specification*. Available from: <[http://portal.opengeospatial.org/files/?artifact\\_id=20040](http://portal.opengeospatial.org/files/?artifact_id=20040)>.

OGC 07-006r1, *OpenGIS<sup>®</sup> Catalogue Service Implementation Specification*. Available from: <[http://portal.opengeospatial.org/files/?artifact\\_id=20555](http://portal.opengeospatial.org/files/?artifact_id=20555)>.

OGC 04-094, *OpenGIS<sup>®</sup> Web Feature Service (WFS) Implementation Specification*. Available from: <[http://portal.opengeospatial.org/files/?artifact\\_id=8339](http://portal.opengeospatial.org/files/?artifact_id=8339)>.

OASIS regrep-rim-3.0-os, *ebXML Registry Information Model, Version 3.0*. Available from: <<http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.pdf>>.

## 3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

### 3.1

#### **catalog**

A mechanism containing interfaces to discover, browse, and query metadata about data, services, and other potential resources.

### 3.2

#### **data view**

A named grouping of a related set of feature types.

**3.3****DoD Discovery Metadata Specification (DDMS)**

Specifies a set of information fields that are to be used to describe any data or service asset, i.e., resource, that is to be made discoverable to the DoD Enterprise, and it serves as a reference for developers, architects, and engineers by laying a foundation for Discovery Services.

**3.4****ebRIM catalog**

A web service that is a profile of the CSW part (Clause 10) of the *OpenGIS® Catalogue Service Implementation Specification* (v2.0.2, OGC-07-006r1). The information model for an ebRIM catalog is ebXML.

**3.5****ebXML Registry Information Model (ebRIM)**

Metadata and content that can be stored in an ebRIM catalog.

**3.6****integrated client**

A software application that has the capability of bringing together offerings from multiple OGC web services. A typical integrated client incorporates a legend, a map view, a service connection manager and a query facility.

**3.7****Web Feature Service (WFS)**

A web service that provides operations for accessing and manipulating geographic features using a request/response pattern using the HTTP protocol.

**4 Conventions****4.1 Abbreviated terms**

CS/W	Catalog Service for the Web
DDMS	DoD Discovery Metadata Specification
DFDD	DGIWG Feature Data Dictionary
DIGEST	Digital Geographic Information Exchange Standard
ebRIM	ebXML Registry Information Model
ebXML	Electronic Business using eXtensible Markup Language
FACC	Feature and Attribute Coding Catalogue
GEOINT	Geospatial Intelligence
GML	Geography Markup Language
GSIP	GEOINT Structure Implementation Profile
GWG	Geospatial Intelligence Standards Working Group

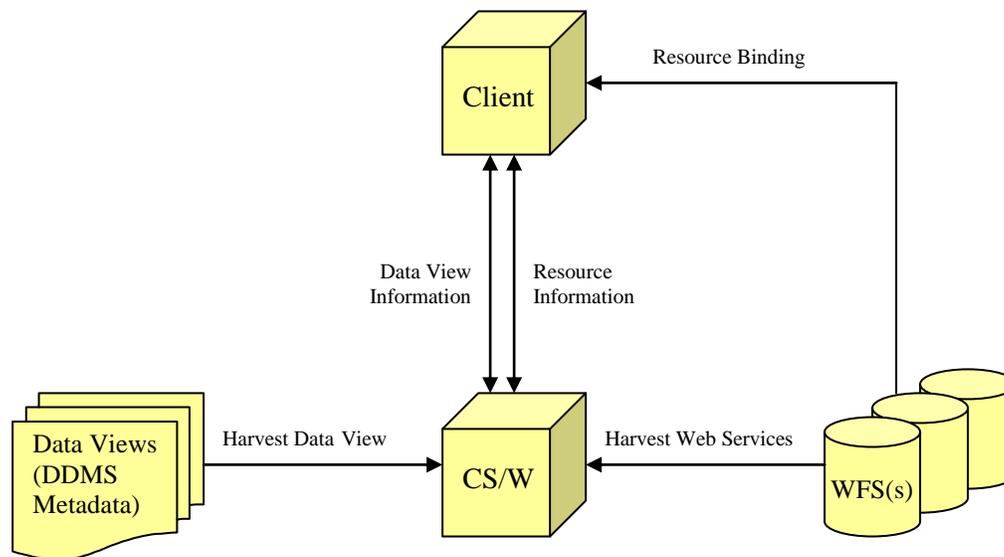
ISO	International Organization for Standardization
MSD	Mission Specific Data
NAS	NSG Application Schema
NEC	NSG Entity Catalog
NFDD	NSG Feature Data Dictionary
NGA	National Geospatial-Intelligence Agency
NSG	National System for Geospatial Intelligence
NSG FC	NSG Feature Catalog
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
UML	Unified Modeling Language
WFS	Web Feature Service
XML	eXtensible Markup Language

#### **4.2 UML notation**

The diagram that appears in Figure 3 in this document is presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

### **5 Data View Architecture Overview**

The principle motivation for the Data View concept addresses the need for clients to be able to access required data without a detailed knowledge of the available service endpoints. The client experience is simplified by presenting only a list of Data Views that represent a collection of actual feature types. The data view can be seen as a grouping or categorization of a set of feature types or layers as shown in Figure 1.



**Figure 1 — Data View Architecture Overview**

A WFS is a service that offers a collection of feature types. Given a service endpoint for a WFS, a client may bind to a service and browse its feature type offerings. If many such services and feature types are available, building the desired operational picture with an integrated client can be tedious and time-consuming.

The availability of Data Views considerably simplifies the process of creating an operational picture. A Data View defines a grouping of feature types judged to be of value when presented together. By selecting one or more Data Views, an analyst using an integrated client can build an operational picture much more rapidly than by researching and handpicking from among many, many feature types individually. A discovery service such as an OGC catalog can allow a client to find views of interest and to locate the endpoints of WFS services that offer the feature types associated with the selected views.

The Data View concept originates in the *NSG Entity Catalog (NEC)* [1]<sup>1)</sup>. The introduction to *Appendix A - NEC View Thesaurus* presents this example:

- The entities “**Aerial**”, “**Building**”, “**Communication Line**”, and “**Wireless Telecommunication Information**”,
- are associated with the *View* “**Communication Facilities**”.
- The “**Building**” entity is also associated with other Views, *e.g.*, “**General Structures**” and “**Manufacturing Installations**”.

The *NEC View Thesaurus* also states that views may be used with entities in the NSG Application Schema (NAS), which is how we made use of Data Views in this test bed.

---

<sup>1)</sup> Although the term *Data View* is used in this report, the *NEC* defines a shorter term *View* to give meaning to this same concept. The *NEC* also defines a term called *View Group*, but that concept was not explored in this test bed.

## 6 Data View Architecture

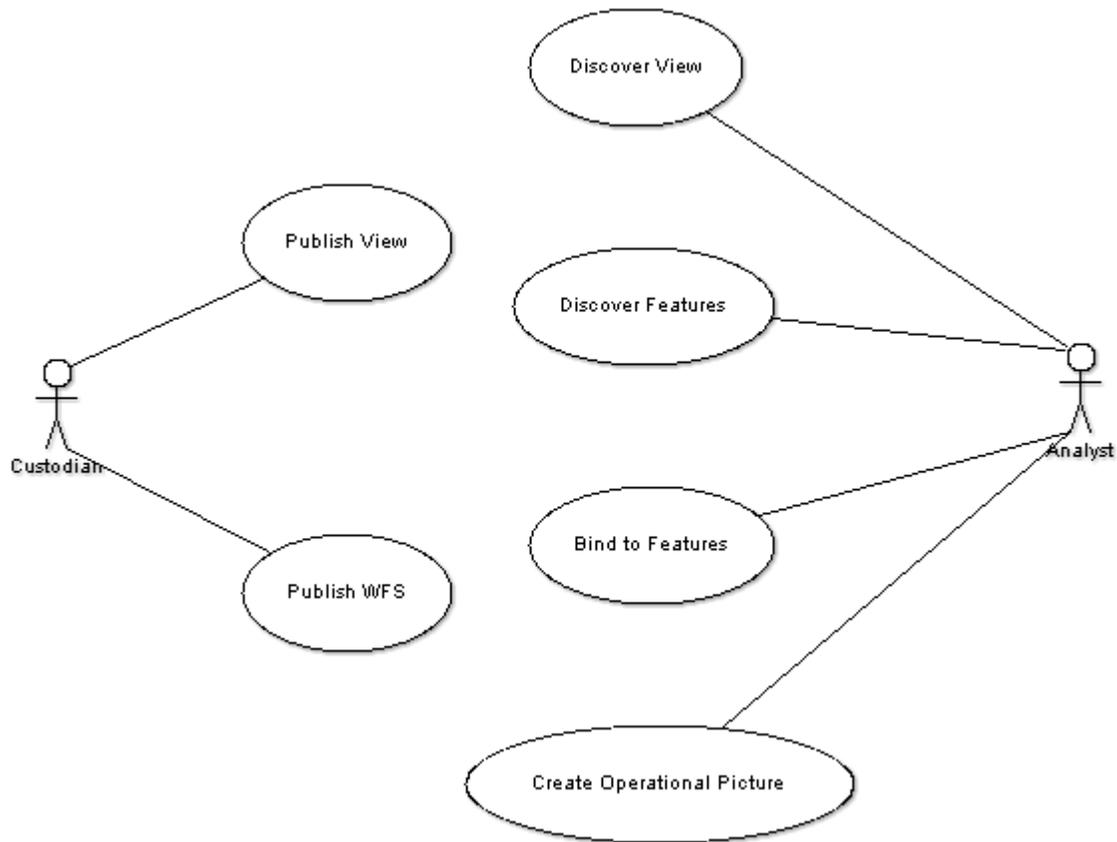
### 6.1 Data View Enterprise Viewpoint

The Enterprise Viewpoint describes the purpose of the architecture.

It is unreasonable to expect all users of a client application to have a ready knowledge of the feature types available through Web Services that comprise the type of view being generated. For this reason, the concept of a Data View has been created. A Data View represents a collection of feature types or layers and provides a way to categorize a set of feature types or layers into groupings.

For example, instead of the user being required to make a number of catalog queries in order to determine the service endpoints that serve ROADS, TRAILS, RAILWAYS and TUNNELS the user can simply select a Data View called “Transportation”. By selecting Transportation, the catalog determines, through associations, what feature types are included in this category and returns the appropriate catalog response identifying the web service(s) from which a user can retrieve that data.

A high-level depiction of the Data View concept is shown in the use case diagram in Figure 2. A data custodian (such as a catalog authority) publishes feature type offerings from web feature services, as well as a set of views, where each view is a grouping of feature types. An analyst (via an Integrated Client software application) may then query those holdings to find one or more views of interest and determine the feature types associated with each those views. Once a feature type is known, the analyst may again query the holdings to determine one or more web feature services that offer that feature type. Finally, the analyst directs the client application to bind to those web feature services, display the selected feature types and ultimately create an operational picture. The Data Views, which provide predefined feature groupings, expedite the process of creating an operational picture.



**Figure 2 — Data View Use Case Diagram**

The following scenario details a typical flow of information based on client requests:

1. The USER has a mission that requires knowledge of Power Generation facilities over a particular geographic area.
2. Using the Client, the USER queries the Catalog for Data View descriptions.
3. The Catalog returns to the Client the Textual Description of all the Data Views. Included in the response is the description of the Power Generation and Transmission Facilities data view and the Communication Facilities data view. Also returned is a listing of the associated “user-friendly” feature type names for the different data views. These “user-friendly” names may or may not match exactly the actual names of the feature types served by a particular WFS.
4. The USER selects the Power Generation and Transmission Facilities data view and the Communications Facilities data view for feature retrieval. The Client queries the Catalog with the data view information and receives a listing of the actual feature type names.
5. The Client queries the Catalog, providing a geographic area of interest, to determine the WFS endpoints that satisfy these features.

6. The Client presents the USER with a list of data view(s) and associated features types indicating availability or no availability.
7. The Client also indicates if specific feature types are available from more than one location. (Some features from Local MSD and some from DVOF data):
  - a. Power Generation and Transmission Facilities
    - i. In common features = Buildings, Power line, Smokestack
    - ii. Features from Local MSD = Buildings, Cooling Tower, Power lines, Power station, Power substation.
    - iii. Features from DVOF = Building, Power line, Power line pylon, Pylon, Smokestack
  - b. Communication Facilities
    - i. In common features = Buildings
    - ii. Features from Local MSD = Buildings, Cable, Tower
    - iii. Features from DVOF = Building, Pylon
8. The USER selects feature types for download.
9. The Client automatically binds to the WFS service(s) to extract those selected feature types and displays the features.

NOTE Based on the flow described above, similar data coming from various WFS resources may represent a duplication of data. If needed, the Client could also execute additional functionality to conflate this data into a single representation. This work is being investigated in the GeoProcessing Workflow thread.

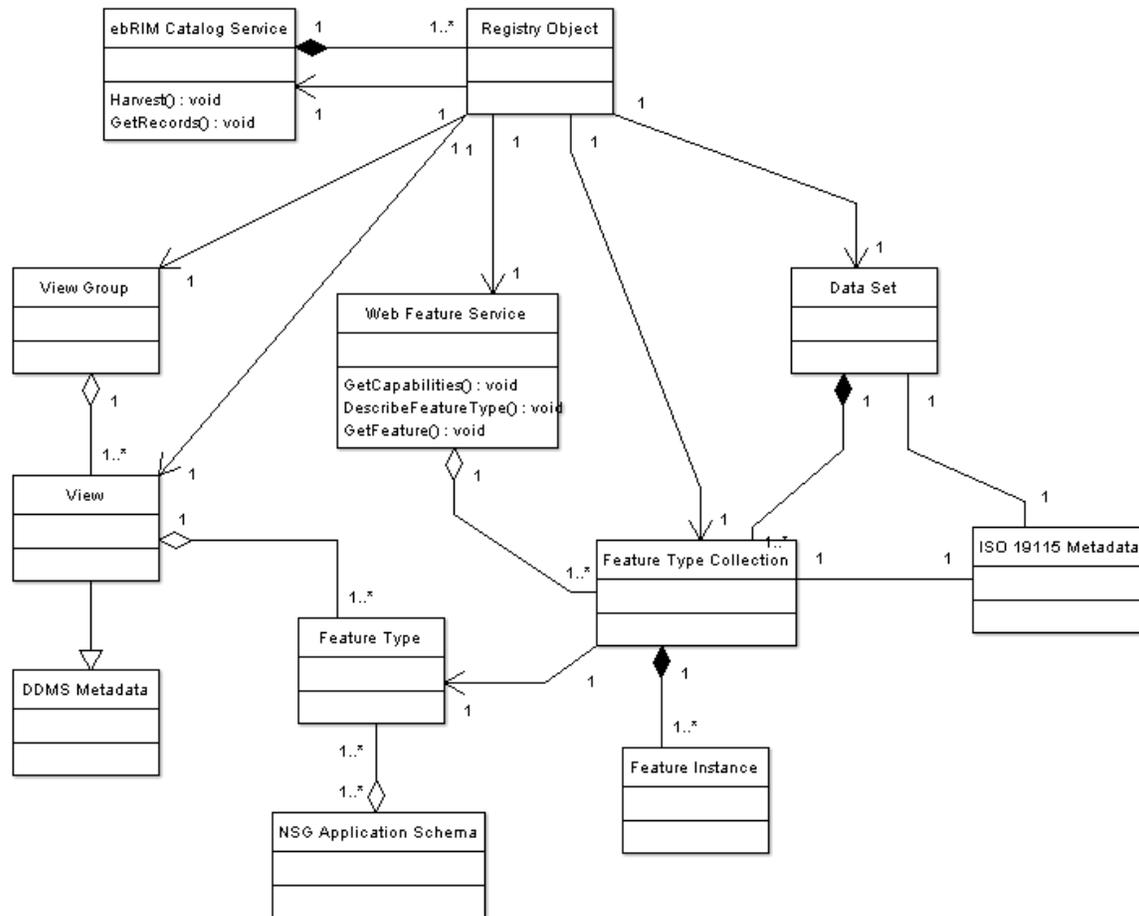
## 6.2 Data View Information Viewpoint

The Information Viewpoint presents a description of the architecture data models.

This section presents a UML model of the Data View Architecture, discusses the information content of a Data View object and the modeling of the Data View Architecture in an eBRIM Catalog.

### 6.2.1 UML Class Diagram of Data View Architecture

The UML class diagram in Figure 3 depicts the data objects that comprise the Data View concept presented in this document. The role of the object named *Registry Object* in this diagram is to capture metadata about the objects it references; the *Registry Object* is a principle building block of an eBRIM Catalog.



**Figure 3 — Data View Class Diagram**

As can be seen in Figure 3, a Data View encoding is based on the DDMS metadata specification. Many Data Views may be defined in a single DDMS metadata file.

### 6.2.2 Data View XML Encoding

The Data View is encoded as DDMS metadata. In a DDMS metadata file, the feature types comprising a data view are defined using namespace-qualified names (QNames) located in elements reachable via the XPath expression

`//ddms:subjectCoverage/ddms:Subject/ddms:category`. These feature types directly refer to feature types defined in the *NSG Application Schema (NAS)* [1]. The following is an example of a DDMS metadata record defining the Data View “Agricultural Buildings and Structures”:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- A DDMS v1.4.1 metacard for a GSIP v1.8.1 View. Its Subjects
are its associated Feature Types (labels and codes). -->
```

```

<ddms:Resource xmlns:ICISM="urn:us:gov:ic:ism:v2"
xmlns:ddms="http://metadata.dod.mil/mdr/ns/DDMS/1.4/">
  <ddms:identifier ddms:qualifier="URL"
ddms:value="http://metadata.dod.mil/mdr/ns/GSIP/1.8.1/view/agricu
ltureBuildingStructure"/>
  <ddms:title ICISM:classification="U" ICISM:ownerProducer="U.S.
National Geospatial-Intelligence Agency">
    Agricultural Buildings and Structures
  </ddms:title>
  <ddms:description ICISM:classification="U"
ICISM:ownerProducer="U.S. National Geospatial-Intelligence
Agency">Man-made features associated with the practice of
agriculture. [description] For example, barns, greenhouses, feed
storage facilities and equipment sheds. The types of features
found in an agricultural area vary with the principal products
being raised, the level of technology, the size of the operation
and the geographic location.</ddms:description>
  <ddms:language ddms:qualifier="ISO 639-1" ddms:value="en"/>
  <ddms:rights ddms:copyright="true"/>
  <ddms:type ddms:value="View"/>
  <ddms:creator ICISM:classification="U"
ICISM:ownerProducer="U.S. National Geospatial-Intelligence
Agency">
    <ddms:Organization>
      <ddms:name>NGA</ddms:name>
      <ddms:phone>+1-703-814-4580</ddms:phone>
      <ddms:email>NCGIS-mail@nga.mil</ddms:email>
    </ddms:Organization>
  </ddms:creator>
  <ddms:publisher ICISM:classification="U"
ICISM:ownerProducer="U.S. National Geospatial-Intelligence
Agency">
    <ddms:Organization>
      <ddms:name>NGA</ddms:name>
      <ddms:phone>+1-703-814-4580</ddms:phone>
      <ddms:email>NCGIS-mail@nga.mil</ddms:email>
    </ddms:Organization>
  </ddms:publisher>

```

```

<ddms:format>
  <ddms:Media>
    <ddms:mimeType>text/xml</ddms:mimeType>
    <ddms:extent/>
    <ddms:medium>digital</ddms:medium>
  </ddms:Media>
</ddms:format>
<ddms:subjectCoverage>
  <ddms:Subject>
    <ddms:category ddms:code="Barn" ddms:label="Barn"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Building" ddms:label="Building"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="BuildingOverhang"
ddms:label="Building Overhang"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="BuildingSuperstructure"
ddms:label="Building Superstructure"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Facility" ddms:label="Facility"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Farm" ddms:label="Farm"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Fence" ddms:label="Fence"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Gate" ddms:label="Gate"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="GrainElevator" ddms:label="Grain
Elevator"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="GrainStorageStructure"
ddms:label="Grain Storage Structure"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Greenhouse"
ddms:label="Greenhouse"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Hopper" ddms:label="Hopper"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="IndustrialFarm"
ddms:label="Industrial Farm"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Installation"
ddms:label="Installation"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="NonBuildingStructure"
ddms:label="Non-building Structure"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Ranch" ddms:label="Ranch"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Shed" ddms:label="Shed"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>

```

```
<ddms:category ddms:code="Stable" ddms:label="Stable"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
  <ddms:category ddms:code="Wall" ddms:label="Wall"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
    <ddms:category ddms:code="Windmill" ddms:label="Windmill"
ddms:qualifier="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"/>
      </ddms:Subject>
    </ddms:subjectCoverage>
  <ddms:security ICISM:classification="U"
ICISM:ownerProducer="U.S. National Geospatial-Intelligence
Agency"/>
</ddms:Resource>
```

For clarity, Table 1 shows the feature type content present in the XML encoding of the Data View.

**Table 1 — Data View: Agricultural Buildings and Structures**

xmlns:nas="http://metadata.dod.mil/mdr/ns/GSIP/schema/nas"

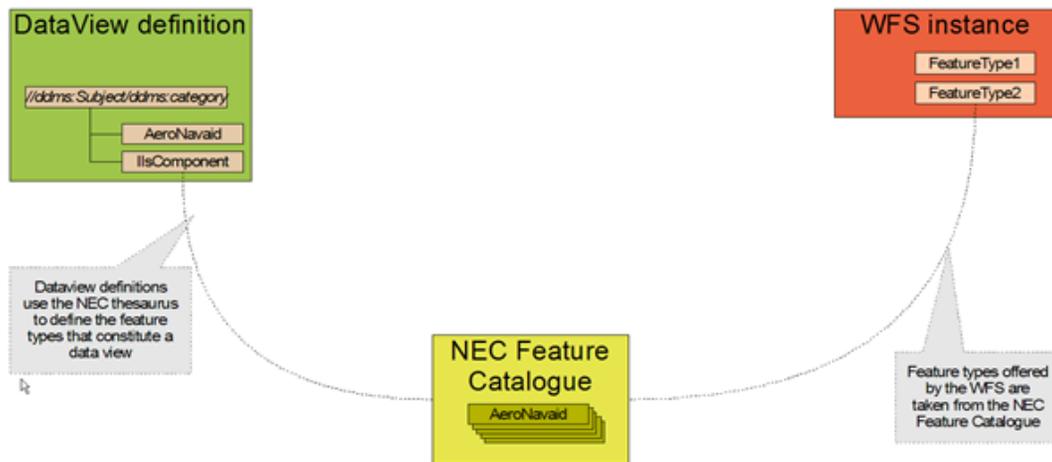
Label	Code
Barn	nas:Barn
Building	nas:Building
Building Overhang	nas:BuildingOverhang
Building Superstructure	nas:BuildingSuperstructure
Facility	nas:Facility
Farm	nas:Farm
Fence	nas:Fence
Gate	nas:Gate
Grain Elevator	nas:GrainElevator
Grain Storage Structure	nas:GrainStorageStructure
Greenhouse	nas:Greenhouse
Hopper	nas:Hopper
Industrial Farm	nas:IndustrialFarm
Installation	nas:Installation
Non-Building Structure	nas:NonBuildingStructure
Ranch	nas:Ranch
Shed	nas:Shed
Stable	nas:Stable
Wall	nas:Wall
Windmill	nas:Windmill

### 6.2.3 Modeling Data View Resources in ebRIM

In OWS-5, the ebRIM profile of the Catalog 2.0.2 specification was selected to satisfy the discovery goals for the Data View architecture. This section describes the modeling of the Data View architecture resources in ebRIM.

Figure 4 depicts the resources in the Data View architecture. These resources are:

- The Data View objects which are encoded as DDMS metadata
- The NEC feature catalog that defines the feature types involved, and
- The WFS services that serve instances of those feature types.



**Figure 4 — Resources in the Data View Architecture**

**NOTE** From an implementation perspective, the Data View objects and WFS services in this architecture relate most directly to feature types as defined in the NSG Application Schema (NAS). The NAS is an XML encoding of the NEC.

In order to store resources in an eBRIM Catalog, those resources must first be mapped into the eBRIM information model. Here is an overview of the mapping into eBRIM of the resources in the Data View architecture:

- The NEC feature catalog becomes an eBRIM classification scheme.
- Each of the feature types defined in the NEC becomes an eBRIM classification node.
- Each Data View (encoded as DDMS metadata) is mapped to an eBRIM extrinsic object.
- For each NEC feature type that is member of a data view, the corresponding classification node is associated with the data view extrinsic object.
- Each WFS service is harvested into the catalog using the usual eBRIM structure for OGC services. [7] (In particular, for OWS-5, each OGC feature type object in the catalog is classified by the corresponding NEC classification node.)

Figure 5 and Figure 6 depict the mapping of the Data View resources into ebRIM.

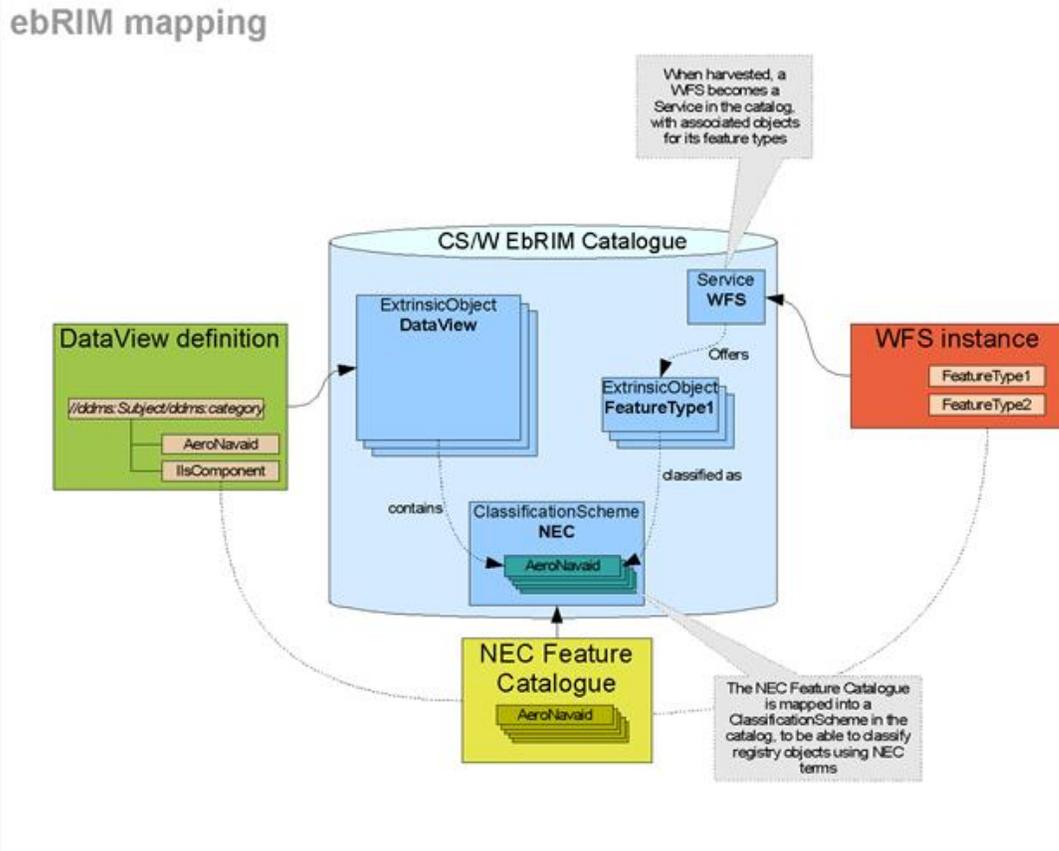


Figure 5 — Mapping Data View Artifacts to ebRIM, Part 1

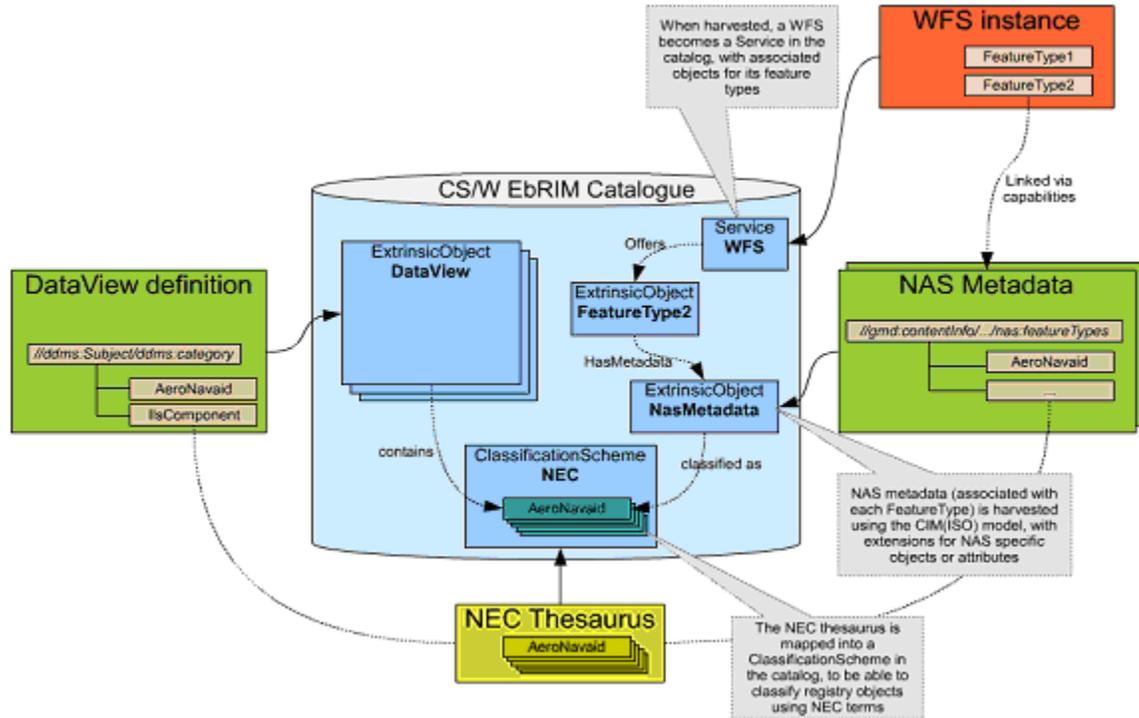


Figure 6 — Mapping Data View Artifacts to ebRIM, Part 2

#### 6.2.4 Modeling Data View Resources in ebRIM, Harvesting Details

This section contains details of the harvesting of resources into the ebRIM catalog.

##### 6.2.4.1 Harvesting of NAS QNames

When referenced in some resource to be harvested, NAS *QNames* are turned into concepts (ebRIM *ClassificationNodes*), belonging to an ebRIM *ClassificationScheme* identified by the namespace of the *QName*. Within the catalog used in the OWS-5 test bed, this mechanism is a feature that allows for easy reference by URI of externally defined concepts, such as concepts defined in online OWL files, taxonomies, etc.

Then, when harvesting a resource that contains a reference to some NAS *QName* (e.g. <http://metadata.dod.mil/mdr/ns/GSIP/schema/nas#AeroNavaid>), a *ClassificationNode* named *AeroNavaid* is created, belonging to the *ClassificationScheme* uniquely identified by <http://metadata.dod.mil/mdr/ns/GSIP/schema/nas>. (The UUID of the *ClassificationScheme* is actually the MD5 hash of this URI, as specified in ebRIM 3.0 for the conversion of URIs into UUIDs).

If the *ClassificationScheme* and/or *ClassificationNode* already exist, they are reused.

Thus, harvesting the DDMS Data View definitions creates or reuses concepts from the NEC Thesaurus; the same happens when harvesting NAS metadata. Therefore, Data

Views and NAS instances are actually linked in the ebRIM structure via those *ClassificationNodes*.

The NAS *ClassificationScheme*, as harvested in the OWS-5 ebRIM Catalog, can be found at:

<http://dev.ionicsoft.com:8085/ionicwrs/wrs/WRS?request=GetRecordById&id=urn:F6318082-AC97-F6A0-FF07-D3A50332AFEF&outputSchema=EBRIM>

where urn:F6318082-AC97-F6A0-FF07-D3A50332AFEF is the MD5 hash of <http://metadata.dod.mil/mdr/ns/GSIP/schema/nas> .

#### 6.2.4.2 Harvesting of Data Views

Data Views, when harvested in the catalog, are stored in *ExtrinsicObjects* complying with the following structure:

```
[objectType=urn:x-ogc:specification:
  csw-ebRim:ObjectType:Schema:XML-Schema:DDMSResource]
  |---- [Slot] identifier
  |---- [Slot] medium
  |---- [Slot] copyrights
  |---> Creator ---> [Organization]
```

(At the moment, Data Views are classified with the NAS *ClassificationNodes* that are found in the *ddms:Subject/ddms:category* of the Data Views definitions. This should change in favor of an association, like *containsFeatureType*.)

As a sample, the *Aeronautical En Route Navigational Aids* Data View has the UUID urn:uuid:f38273f2-f31f-4c1e-90a4-6950dfdd4e58 .

It can be retrieved here:

<http://dev.ionicsoft.com:8085/ionicwrs/wrs/WRS?request=GetRecordById&id=urn:uuid:f38273f2-f31f-4c1e-90a4-6950dfdd4e58&outputSchema=EBRIM>

#### 6.2.4.3 Harvesting of NAS/ISO-19115 Metadata

NAS metadata is defined as an extension of ISO19115 metadata. On the other hand, the CIM Extension Package defines the mapping from ISO19115 metadata to the equivalent ebRIM structure.

Therefore, NAS metadata is harvested in the catalogue using the CIM mapping, with extra elements added to the CIM structure where needed.

```
[objectType=urn:x-ogc:specification:
  csw-ebRim-nas:ObjectType:NasDataMetadata]
```

```
|---- (various CIM slots and associations)
|----> AssociationType_Contains ----> [Content]
```

In this structure, the Content object type is a custom object type, added to hold, through classifications, the NEC *ClassificationNodes* that are found in the /gmd:MD\_FeatureCatalogueDescription/nas:featureTypes elements of the NAS metadata documents.

The sample NAS metadata document bundled with the schema has been harvested in the catalog:

<http://dev.ionicsoft.com:8085/ionicwrs/wrs/WRS?request=GetRecordById&id=urn:uuid:16e859f1-993d-433e-9c67-6bb21798af96&outputSchema=EBRIM>

#### 6.2.4.4 Harvesting a Web Feature Service

Web Services are generally harvested into a catalog using a request based on the service type [7]. For example, the following is a GET request to harvest the Snowflake WFS into the OWS-5 catalog:

[http://dev.ionicsoft.com/ows4catalog/wrs/WRS?request=harvest&version=2.0.0&ResourceType=WFS&Source=http://demo.snowflakesoftware.com:8080/ows5\\_features\\_inline/Features&outputSchema=EBRIM](http://dev.ionicsoft.com/ows4catalog/wrs/WRS?request=harvest&version=2.0.0&ResourceType=WFS&Source=http://demo.snowflakesoftware.com:8080/ows5_features_inline/Features&outputSchema=EBRIM)

### 6.3 Data View Computational Viewpoint

The Computational Viewpoint is concerned with the functional decomposition of the system into a set of distinct components that interact at interfaces.

For OGC, such components are generally realized as web services. The web services of principle concern in this exploration of the Data View architecture are the Web Feature Service and the eBRIM Catalog Service.

#### 6.3.1 Web Feature Service

For the OWS-5 test bed, Snowflake Software deployed a web feature (WFS) service that served sample Local Mission Specific Data (MSD) provided by NGA. The Local MSD data provided was based on a subset of the NSG Application Schema (NAS).

A Data View associates a group of feature types defined in an application schema. The Data Views constructed for experimentation in the test bed contained feature types in the NAS, a subset of which was published by the Snowflake WFS.

#### 6.3.2 eBRIM Catalog Service

Leica Geosystems deployed an eBRIM catalog service to provide discovery services for the OWS-5 test bed.

The catalog deployed in the OWS-5 test bed offered an ebRIM 1.0.0 profile of Catalog 2.0.2 specification as defined in [4].

#### 6.4 Data View Engineering Viewpoint

The Engineering Viewpoint describes specific components linked by a communications network and is concerned with the interaction of distinct computational objects.

In the OWS-5 test bed, these are the components that played a role in the Data View architecture experiment:

- An ebRIM Catalog 1.0.0, provided by Leica Geosystems
- A Web Feature Service (WFS 1.1.0), provided by Snowflake Software, and
- An Integrated Client, provided by Intergraph Corporation.

Figure 7 depicts the high-level interaction between the Integrated Client and the ebRIM Catalog service. The Integrated Client submits a query (request message) to the catalog and receives a result set (response message) containing metadata for the resources of interest.

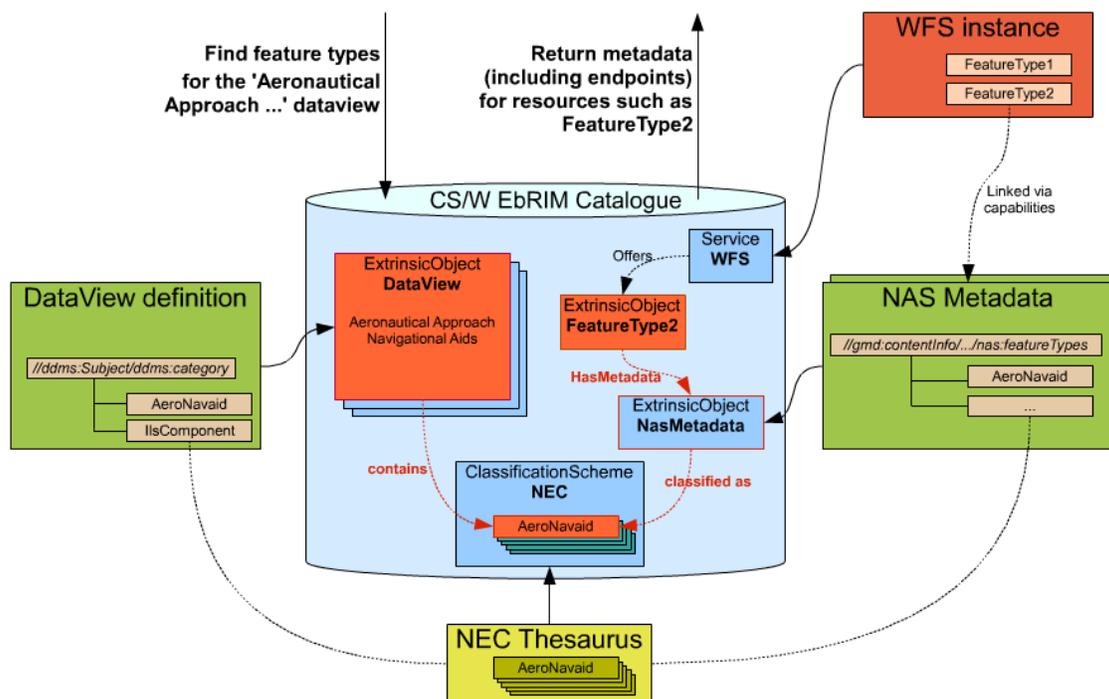


Figure 7 — Client Interaction with the Catalog

#### **6.4.1 Test Bed Data View Use Case**

Here is a list of the questions asked of the ebRIM Catalog by the Integrated Client to implement the test bed use case for the Data View architecture:

1. What are the available data views?
2. What are the feature types that are members of a given data view?
3. Of those feature types that are members of a given data view, which are actually being served by one or more WFS services?
4. What are the URLs of the WFS services that serve a given Feature Type?

#### **6.4.2 Client Request Messages Submitted to ebRIM Catalog**

The following clauses describe in detail the request messages sent by the Integrated Client to the ebRIM Catalog service to demonstrate the Data View architecture use case implemented by Intergraph.

### 6.4.2.1 Find Data View Objects

This request finds all Data View ExtrinsicObjects where the Data View is classified in the "nas" *ClassificationScheme* and the name of the Data View contains the string "Trans."

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords maxRecords="5"
  outputFormat="application/xml; charset=UTF-8"
  outputSchema="EBRIM" resultType="results"
  version="2.0.0"
  xmlns="http://www.opengis.net/cat/csw">
  <Query typeNames="ExtrinsicObject Classification
    ClassificationScheme">
    <ElementSetName typeNames="ExtrinsicObject">
      summary
    </ElementSetName>
    <Constraint version="1.0.0">
      <ogc:Filter
        xmlns:ebxml=
          "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.5"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns:ogc="http://www.opengis.net/ogc">
        <ogc:And>
          <ogc:PropertyIsLike escape="\\" singleChar="_"
            wildCard="%">
            <ogc:PropertyName>
              /ExtrinsicObject/@objectType</ogc:PropertyName>
            <ogc:Literal>%DDMSResource%</ogc:Literal>
          </ogc:PropertyIsLike>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>
              /ExtrinsicObject/@id
            </ogc:PropertyName>
            <ogc:PropertyName>
              /Classification/@classifiedObject
            </ogc:PropertyName>
          </ogc:PropertyIsEqualTo>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>
              /Classification/@classificationScheme
            </ogc:PropertyName>
            <ogc:PropertyName>
              /ClassificationScheme/@id</ogc:PropertyName>
          </ogc:PropertyIsEqualTo>
        </ogc:And>
      </ogc:Filter>
    </Constraint>
  </Query>
</GetRecords>
```

```

    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>
        /ClassificationScheme/Name/LocalizedString/@value
      </ogc:PropertyName>
      <ogc:Literal>nas</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsLike escape="\\" singleChar="_"
      wildCard="%">
      <ogc:PropertyName>
        /ExtrinsicObject/Name/LocalizedString/@value
      </ogc:PropertyName>
      <ogc:Literal>%Trans%</ogc:Literal>
    </ogc:PropertyIsLike>
  </ogc:And>
</ogc:Filter>
</Constraint>
</Query>
</GetRecords>

```

For this query, the OWS-5 ebRIM Catalog returned a set of Data View *ExtrinsicObjects* whose names were:

- \* Inland Water Transportation and Associated Features
- \* Pedestrian and/or Animal Transportation and Associated Features
- \* Power Generation and Transmission Facilities
- \* Transportation Support Structures

### 6.4.2.2 Find Feature Types That Are Members of a Data View

This request finds all Feature Types that are members of the Data View named "Power Generation and Transmission Facilities."

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords maxRecords="100"
  outputFormat="application/xml; charset=UTF-8"
  outputSchema="EBRIM"
  resultType="results"
  version="2.0.0"
  xmlns="http://www.opengis.net/cat/csw">
  <Query typeNameNames="ClassificationNode_cnFeaType
Classification_clDataView ExtrinsicObject_eoDataView">
  <ElementSetName typeNameNames="$cnFeaType">
    full
  </ElementSetName>
  <Constraint version="1.0.0">
    <ogc:Filter xmlns:ebxml=
      "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.5"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:And>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>
            $cnFeaType/@id
          </ogc:PropertyName>
          <ogc:PropertyName>
            $clDataView/@classificationNode
          </ogc:PropertyName>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>
            $clDataView/@classifiedObject
          </ogc:PropertyName>
          <ogc:PropertyName>
            $eoDataView/@id
          </ogc:PropertyName>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
    </Constraint>
  </Query>
</GetRecords>
```

```

      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>
          $eoDataView/@id
        </ogc:PropertyName>
        <ogc:Literal>
          <!-- Power Generation and Transmission Facilities -->
          urn:uuid:1662bf05-33bc-44ac-8b8e-f78d3795155c
        </ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:And>
  </ogc:Filter>
</Constraint>
</Query>
</GetRecords>

```

For this query, the OWS-5 ebRIM Catalog returned a set of Feature Type *ClassificationNodes* with the following codes:

```

* Building
* BuildingOverhang
* BuildingSuperstructure
* CoolingFacility
* CoolingTower

* EntranceExit
* Facility
* HeatingFacility
* Installation
* NonBuildingStructure

* NuclearReactorContainment
* OverheadObstruction
* PointOfChange
* PowerLine
* PowerLinePylon

* PowerStation
* PowerSubstation
* Pylon
* PylonInfo
* Smokestack

* SolarFarm
* SolarPanel
* WindFarm
* Windmill
* Windmotor

```

### 6.4.2.3 Find Available Feature Type Instances That Are Members of a Data View

This request finds all available Feature Type instances that are members of the Data View named "Power Generation and Transmission Facilities." (In OWS-5, the Feature Type instances in the catalog were those harvested from the Snowflake WFS.)

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords maxRecords="100"
  outputFormat="application/xml; charset=UTF-8"
  outputSchema="EBRIM" resultType="results"
  version="2.0.0"
  xmlns="http://www.opengis.net/cat/csw">
  <Query typeNames="ExtrinsicObject_eoFeaType
ExtrinsicObject_eoDataView Classification_clDataView
Classification_clFeatureType">
    <ElementSetName typeNames="$eoFeaType">
      full
    </ElementSetName>
    <Constraint version="1.0.0">
      <ogc:Filter
        xmlns:ebxml=
          "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.5"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns:ogc="http://www.opengis.net/ogc">
        <ogc:And>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>
              $eoDataView/@id
            </ogc:PropertyName>
            <ogc:Literal>
              <!-- Power Generation and Transmission Facilities -->
              urn:uuid:1662bf05-33bc-44ac-8b8e-f78d3795155c
            </ogc:Literal>
          </ogc:PropertyIsEqualTo>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>
              $clDataView/@classifiedObject
            </ogc:PropertyName>
            <ogc:PropertyName>
              $eoDataView/@id
            </ogc:PropertyName>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
      </Constraint>
    </Query>
  </GetRecords>
```

```

    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>
        $sclDataView/@classificationNode
      </ogc:PropertyName>
      <ogc:PropertyName>
        $sclFeatureType/@classificationNode
      </ogc:PropertyName>
    </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>
      $sclFeatureType/@classifiedObject
    </ogc:PropertyName>
    <ogc:PropertyName>
      $seoFeaType/@id
    </ogc:PropertyName>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>
      $seoFeaType/@objectType
    </ogc:PropertyName>
    <ogc:Literal>
      urn:x-ionic:internal-
identifier:ObjectType:FeatureType
    </ogc:Literal>
  </ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>
</Constraint>
</Query>
</GetRecords>

```

For this query, the OWS-5 ebRIM Catalog returned a set of Feature Type *ExtrinsicObjects* (harvested from the Snowflake WFS) with the following names:

```

Building
CoolingTower
PowerLine
PowerStation
PowerSubstation
Smokestack

```

#### 6.4.2.4 Find Feature Type Instances (General)

This request finds all available Feature Type instances whose name contains the string "Power".

**NOTE** The query in 6.4.2.3 was actually used to find Features Types for the test bed use case, but this example request is included here to simply show Feature Type discovery without regard to Data View membership.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords maxRecords="3"
  outputFormat="application/xml; charset=UTF-8"
  outputSchema="EBRIM"
  resultType="results"
  version="2.0.0"
  xmlns="http://www.opengis.net/cat/csw">
  <Query typeNames="ExtrinsicObject">
    <ElementSetName>full</ElementSetName>
    <Constraint version="1.0.0">
      <ogc:Filter xmlns:ebxml=
        "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.5"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns:ogc="http://www.opengis.net/ogc">
        <ogc:And>
          <ogc:PropertyIsLike escape="\ " singleChar="_"
            wildCard="%">
            <ogc:PropertyName>
              /ExtrinsicObject/@objectType
            </ogc:PropertyName>
            <ogc:Literal>%FeatureType%</ogc:Literal>
          </ogc:PropertyIsLike>
          <ogc:PropertyIsLike escape="\ " singleChar="_"
            wildCard="%">
            <ogc:PropertyName>
              /ExtrinsicObject/Name/LocalizedString/@value
            </ogc:PropertyName>
            <ogc:Literal>%Power%</ogc:Literal>
          </ogc:PropertyIsLike>
        </ogc:And>
      </ogc:Filter>
    </Constraint>
  </Query>
</GetRecords>
```

For this query, the OWS-5 ebRIM Catalog returned a set of Feature Type *ExtrinsicObjects* with the following names:

- \* PowerLine
- \* PowerStation
- \* PowerSubstation

#### 6.4.2.5 Find the WFS Endpoint which Operates On a Specified Feature Type Instance

This request finds the WFS service endpoint for the Feature Type instance named “PowerStation”.

```
<?xml version='1.0' encoding='UTF-8'?>
<GetRecords maxRecords="1"
  outputFormat="application/xml; charset=UTF-8"
  outputSchema="EBRIM"
  resultType="results"
  version="2.0.0"
  xmlns="http://www.opengis.net/cat/csw">
<Query typeNames="Service Association">
  <ElementSetName typeNames="Service">
    full
  </ElementSetName>
  <Constraint version="1.0.0">
    <ogc:Filter
      xmlns:ebxml=
        "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.5"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:And>
        <ogc:PropertyIsLike escape="\ " singleChar="_ "
          wildCard="%">
          <ogc:PropertyName>
            /Association/@associationType
          </ogc:PropertyName>
          <ogc:Literal>%OperatesOn%</ogc:Literal>
        </ogc:PropertyIsLike>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>
            /Association/@sourceObject
          </ogc:PropertyName>
          <ogc:PropertyName>/Service/@id</ogc:PropertyName>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>
            /Association/@targetObject
          </ogc:PropertyName>
          <ogc:Literal>
            <!-- PowerStation -->
            urn:uuid:1b3df6a9-c22f-4d42-978b-fcfef8fb21c6
          </ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:And>
    </ogc:Filter>
  </Constraint>
</Query>
</GetRecords>
```

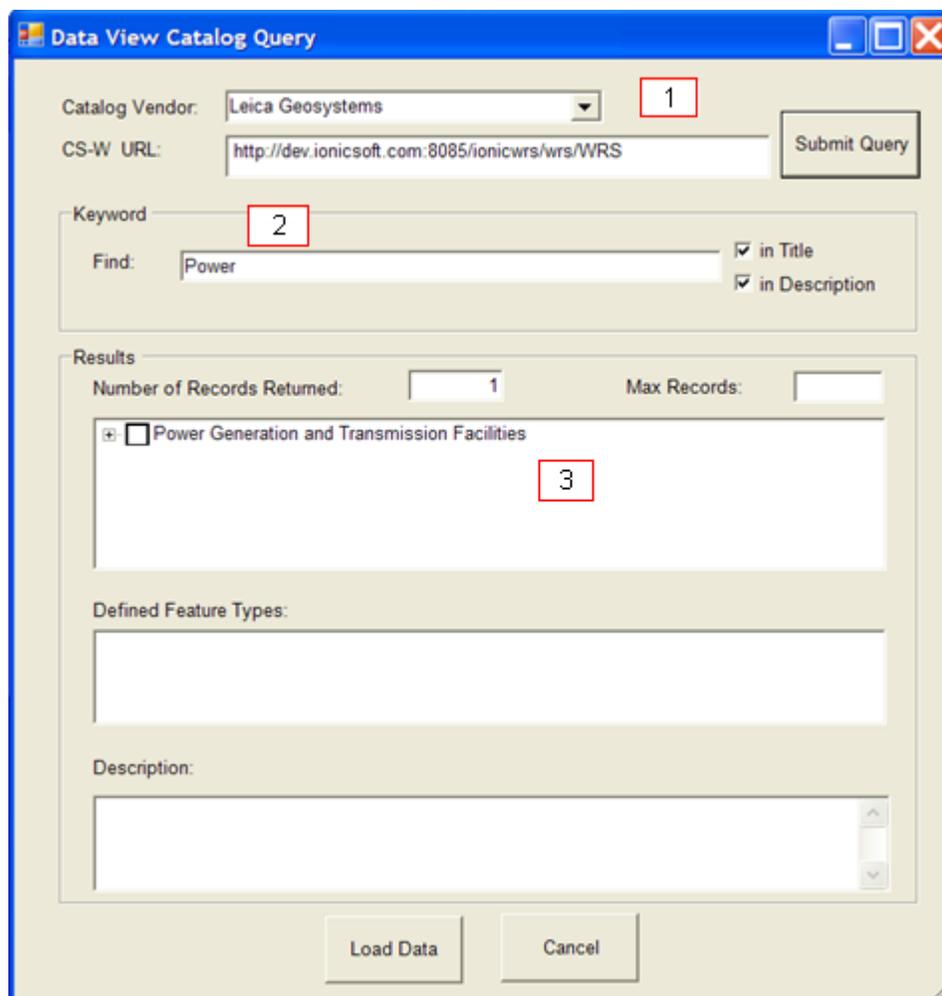
For this query, the OWS-5 ebRIM Catalog returned a *Service* object containing the following endpoint

[http://demo.snowflakesoftware.com:8080/ows5\\_features\\_inline//Features?](http://demo.snowflakesoftware.com:8080/ows5_features_inline//Features?)

### 6.4.3 Presentation of Data Views in the Integrated Client

This section presents the handling of Data Views and the associated Feature Types in the Integrated Client provided by Intergraph Corporation.

Figure 8 depicts the Data View query dialog offered by the Integrated Client. The user selects a catalog endpoint (1), enters the keyword “Power,” (2) and the name of the single Data View matching that keyword is returned from the ebRIM Catalog (3).

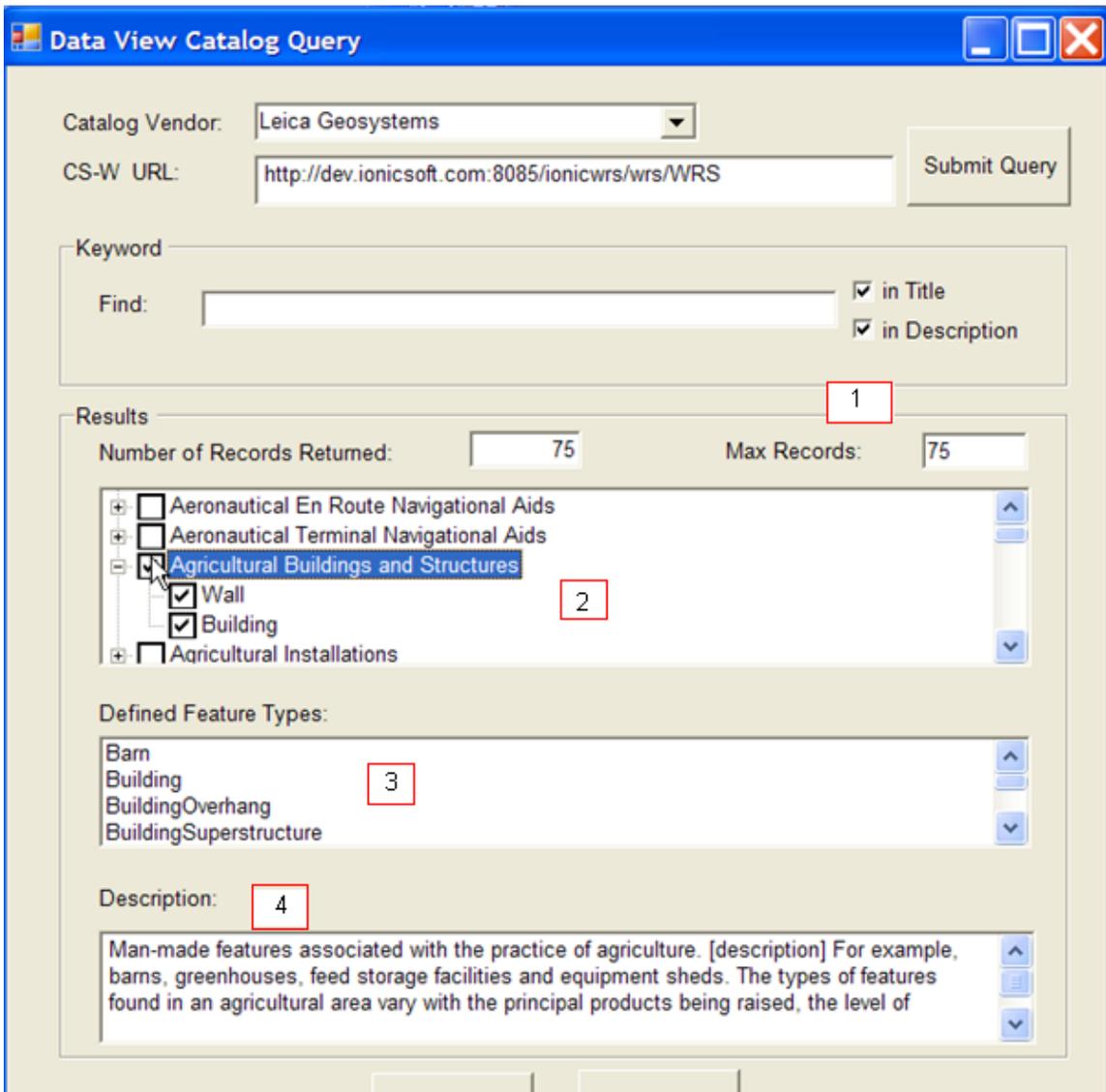


**Figure 8 — Querying Data Views by Keyword**

Figure 9 demonstrates the ability to browse a larger selection of Data Views in the catalog. The user enters a max-records threshold (1), and the names of a selection of Data Views are presented. Here the user also has the opportunity to explore the Feature Type instances that have been harvested in the catalog (from the Snowflake WFS) that belong to each Data View (2).

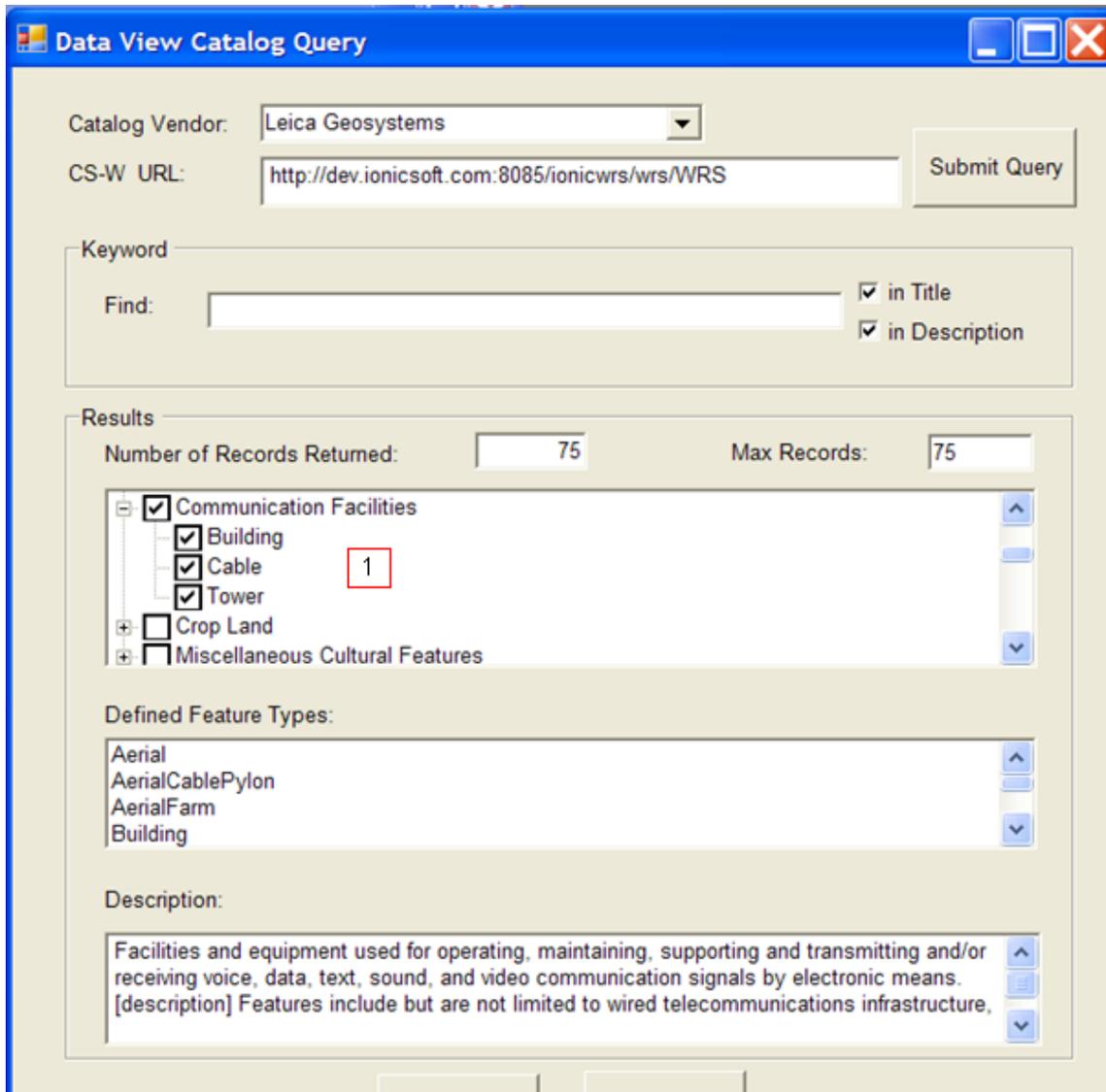
The user is also presented with the list of *all* of the Feature Types that are members of the Data View (3). (However, as can be seen in (2), Feature Type *data* exists for only “Wall” and “Building.”)

Finally, the user is presented with an overall description of the selected Data View (4).



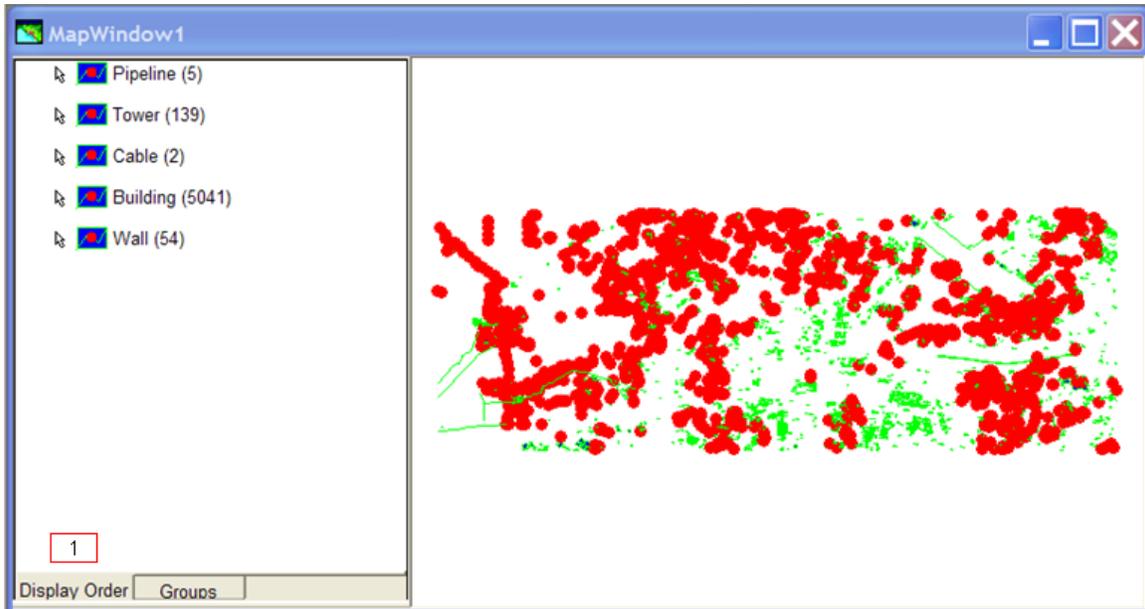
**Figure 9 — Browsing a Selection of Data Views**

Figure 10 demonstrates that a Feature Type can have membership in more than one Data View. In this example, “Building” is a member of “Communication Facilities” (1), and as shown previously in Figure 9-(2), “Building” is also member of “Agricultural Buildings and Structures.”



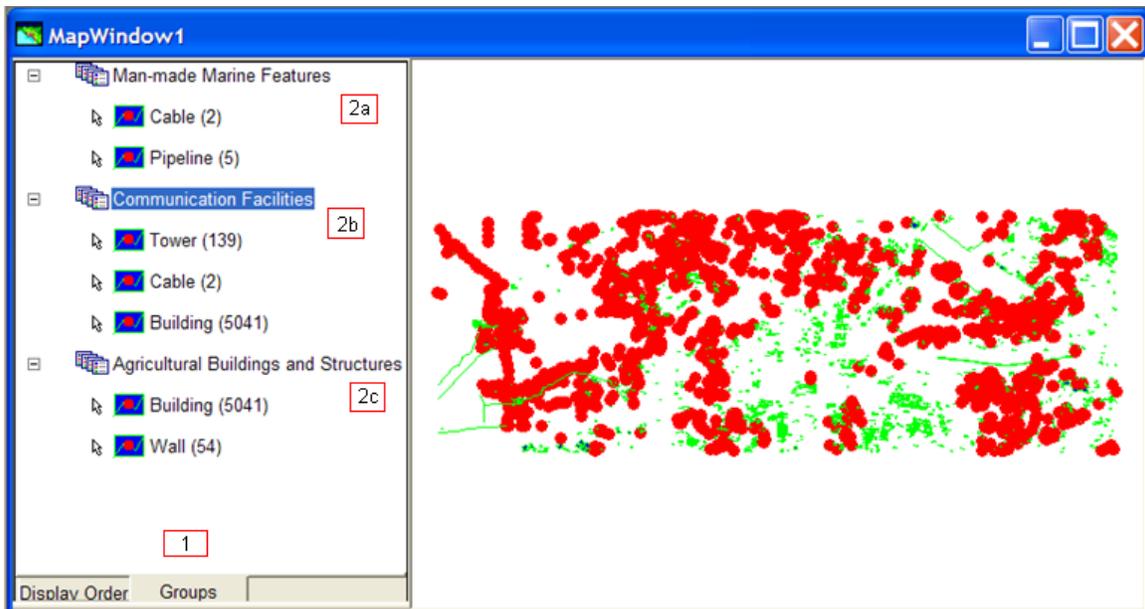
**Figure 10 — A Feature Type Can Be a Member of Multiple Data Views**

Figure 11 depicts the portrayal of the selected Feature Types in the client, and shows the ability of the user to arrange the Feature Types in the legend to govern the display order (1).



**Figure 11 — Selected Features Portrayed in Client**

And finally, in Figure 12, the user may see the selected Feature Types grouped (1) via their membership in the selected Data Views (2a)(2b)(2c). Note that Feature Types “Cable” and “Building” have membership in more than one Data View.



**Figure 12 — Selected Features Grouped by Data View**

## Bibliography

- [1] GWG *GSIP Fact Sheet*, 2007-05-10. Available from:  
<[http://www.gwg.nga.mil/documents/files/GSIP\\_FactSheet.doc](http://www.gwg.nga.mil/documents/files/GSIP_FactSheet.doc)>
- [2] GWG Standards and Registries web site. Located at:  
<[http://www.gwg.nga.mil/stds\\_regs.html](http://www.gwg.nga.mil/stds_regs.html)>.
- [3] Department of Defense Discovery Metadata Specification Home Page. Located at: <<http://metadata.dod.mil/mdr/irs/DDMS/>>.
- [4] OGC 07-110, *OpenGIS<sup>®</sup> Web Registry Service (WRS) – Part 1: ebRIM profile of CSW*, Version 1.0.0 (release candidate), 2007-11-21.
- [5] OGC 07-114, *OpenGIS<sup>®</sup> Web Registry Service (WRS) – Part 2: Basic extension package*, Version 1.0.0 (release candidate), 2007-11-21.
- [6] OGC 07-028r1, *GEOINT Structure Implementation Profile Schema Processing*, Version 0.5, 2007-05-02. Available from:  
<[http://portal.opengeospatial.org/files/?artifact\\_id=21628](http://portal.opengeospatial.org/files/?artifact_id=21628)>
- [7] OGC 05-084, *Catalog 2.0 Accessibility for OWS-3*. Available from:  
<[http://portal.opengeospatial.org/files/?artifact\\_id=12597](http://portal.opengeospatial.org/files/?artifact_id=12597)>