Open Geospatial Consortium

Date: 2011-11-23

Reference number of this document: OGC 11-063r6

Category: Public Engineering Report

Editor(s): Gobe Hobona, Roger Brackin

OGC® OWS-8 Cross Community Interoperability (CCI) Semantic Mediation Engineering Report

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

Preface

The document presents the work completed with respect to the Cross Community Interoperability thread within OWS-8. This Engineering Report describes and evaluates a possible role for interoperability standards within infrastructures based on OGC Web Services.

This document is a deliverable for the OGC Web Services 8 (OWS-8) testbed activity. OWS testbeds are part of OGC's Interoperability Program, a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver proven candidate standards or revisions to existing standards into OGC's Standards Program, where they are formalized for public release. In OGC's Interoperability Initiatives, international teams of technology providers work together to solve specific geoprocessing interoperability problems posed by the Initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments and interoperability support services - all designed to encourage rapid development, testing, validation and adoption of OGC standards.

The OWS-8 sponsors are organizations seeking open standards for their interoperability requirements. After analyzing their requirements, the OGC Interoperability Team recommend to the sponsors that the content of the OWS-8 initiative be organized around the following threads:

* Observation Fusion

* Geosynchronization (Gsync)

* Cross-Community Interoperability (CCI)

* Aviation

More information about the OWS-8 testbed can be found at:

http://www.opengeospatial.org/standards/requests/74

OGC Document [11-139] "OWS-8 Summary Report" provides a summary of the OWS-8 testbed and is available for download:

https://portal.opengeospatial.org/files/?artifact_id=46176

License Agreement

Contents

Page

Figures

Page

OGC® OWS-8 CCI Semantic Mediation Engineering Report

1 Introduction

1.1 Scope

The OWS-8 Cross Community Interoperability (CCI) thread built on progress made in the recent OWS-7 initiative to cover key technology areas that could not be addressed within the scope of that initiative. The OWS-8 CCI thread aimed to increase interoperability within communities sharing geospatial data, including advancing of interoperability among heterogeneous data models, advancing strategies to share styles to provide a more common and automated use of symbology, improvement of KML, and advancing schema automation allowing communities to better share their information artefacts. This OGC engineering report aims to present findings from CCI thread activities towards advancement of semantic mediation involving data retrieved from heterogeneous data models that are available through web services conformant to OGC standards.

The engineering report will briefly introduce relevant details of the semantic web and mediation. The document will make recommendations on establishing a semantic mediation architecture that uses OGC web services and emerging practice from the semantic web community. Based on the scenario adopted by the CCI thread, the document will also discuss the pros and cons of adopting relevant standards. The engineering report will offer recommendations on how specific OGC standards may be adopted or modified in order to support semantic mediation.

1.2 Document contributor contact points

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

Name	Organization
Gobe Hobona	Envitia Ltd.
Roger Brackin	Envitia Ltd.
Luis Bermudez	Open Geospatial Consortium
Robin Houtmeyers	Luciad
Daniel Balog	Luciad
Rob Cass	Compusult
Clemens Portele	interactive instruments GmbH
Paul Birkel	Mitre/NGA
Roberto Lucchi	ESRI

Teddy Matinde	ESRI

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2011/05/25	0.0.1	GH,RB	All	Initial writing
2011/06/23	0.0.2	GH,RB	5.2, 9.5	Added the GeoSPARQL summary and the mapping actions discussion
2011/06/23	0.0.2	RL,DB	11	Added the discussion on the semantic mediator
2011/06/23	0.0.2	LB	6.1,6.2,7	Various edits
2011/06/24	0.0.2	RC	12	Added the resource discovery and catalogue service discussion
2011/08/31	0.0.3	RL,DB	11	Extended discussion on the semantic mediator
2011/08/31	0.0.3	GH,RB	All	Editorial revision
2011/08/31	0.0.4	DB	9.2, 11	Extended details on semantic mediator
2011/09/15	0.0.6	GH	13.2, Appendix A	Inserted Compusult client screenshot. Added Appendix A.

1.4 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.

Berners-Lee T., (2006). *Notation 3*, available from http://www.w3.org/DesignIssues/Notation3.html

Envitia,(2011). *Geospatial Intelligence Integrated Reference Architecture (GI2RA)*, available from http://www.envitia.com/admin/uploads/pagedownloads/live/384/UTP1149-%20GI2RA%20Interoperability%20UTP1149-01.pdf

Howard M., Payne S., Sunderland R., (2010). *Technical Guidance for the INSPIRE Schema Transformation Network Service*, available from http://inspire.jrc.ec.europa.eu/documents/Network_Services/JRC_INSPIRE-TransformService_TG_v3-0.pdf

HP, *Jena – A Semantic Web Framework for Java*, available from http://jena.sourceforge.net

ISO,(2005). *ISO 19119: Geographic information – Services*

ISO,(2005). *ISO 19109: Geographic information – Rules for Application Schema*

ISO,(2010). *ISO 19763: Information technology – Metamodel framework for interoperability (MFI) – Part 3: Metamodel for ontology registration*

ISO,(2005). *ISO 19110: Geographic information – Methodology for feature cataloguing*

ISO,(2009). *N 2705: Report from stage 0 Project 19150 Geographic information – Ontology*

ISO, (2010). *ISO/IEC FDIS 19763-3 Information technology - Metamodel framework for interoperability (MFI) Part 3: Metamodel for ontology registration 2nd Edition*

OASIS, (2005). *ebXML Registry Information Model (RIM) v3.0 Standard*

OGC 03-040, *OGC Reference Model (ORM),* 2003-09-16, available from http://portal.opengeospatial.org/files/?artifact_id=3836

OGC 06-121r3, *OpenGIS® Web Services Common Standard*, 2007-02-09, available from http://portal.opengeospatial.org/files/?artifact_id=20040

OGC 07-160r1, *OGC® OWS-5 Conflation Engineering Report,* 2008-09-12, available from http://portal.opengeospatial.org/files/?artifact_id=30064

OGC 09-048r3*, Name type specification – definitions – part 1 – basic name,* 2010-03-31, available from http://portal.opengeospatial.org/files/?artifact_id=37802

OGC 09-138, *Fusion Standards Study Engineering Report,* 2010-03-2, available from http://portal.opengeospatial.org/files/?artifact_id=36177

OGC 10-184, *Fusion Standards Study, Phase 2 Engineering Report,* 2010-12-13, available from http://portal.opengeospatial.org/files/?artifact_id=41094

OGC 11-052r3, *GeoSPARQL - A geographic query language for RDF data,* 2011-07-02, available from http://portal.opengeospatial.org/files/?artifact_id=44722

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] and in OpenGIS® Abstract Specification shall apply. In addition, the following terms and definitions apply.

3.1
conflation
the process of unifying multiple separate sources of data into one integrated all-encompassing result

3.2
data fusion
the act or process of combining or associating data or information regarding one or more entities considered in an explicit or implicit knowledge framework to improve one's capability (or provide a new capability) for detection, identification, or characterization of that entity

3.3
feature
representation of some real world object or phenomenon

3.4
interoperability
capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units [ISO 19119]

3.5
metadata
data about data

3.6
model
abstraction of some aspects of a universe of discourse [ISO 19109]

3.7
ontology
a formal specification of concrete or abstract things, and the relationships among them, in a prescribed domain of knowledge [ISO/IEC 19763]

4 Conventions

4.1 Abbreviated terms

CCI	Cross Community Interoperability
ER	Engineering Report
GIS	Geographic Information System
GI2RA	Geospatial Intelligence Integrated Reference Architecture
GML	Geography Markup Language

OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
OWS	OGC Web Service
OWS-8	OGC Web Services Initiative, Phase 8
NSG	National System for Geospatial-Intelligence
RDF	Resource Description Framework
TDS	Topographic Data Store
TNM	The National Map
SDI	Spatial Data Infrastructure
SKOS	Simple Knowledge Organization System
SOA	Service Oriented Architecture
SRS	Spatial Reference System
SSA	Shared Situational Awareness
URL	Uniform Resource Locator
URN	Uniform Resource Names
USGS	United states Geological Survey
WFS	Web Feature Service
WMS	Web Map Service
WS	Web Service
XML	eXtensible Markup Language

5 Background Information

In this section a review of relevant recent works and standards is presented.

5.1 Geographic Features

A fundamental unit of geographic information is a feature. The OGC Reference Model (ORM) states that “a feature is an abstraction of a real world phenomenon. A geographic feature is a feature associated with a location relative to the Earth. A digital representation of the real world can be thought of as a set of features” [OGC 03-040]. This approach to modeling geographic features is consistent with the international standard ISO 19110 which provides a standard framework for organizing and reporting the classification of features and ISO 19109 which gives rules for creating application schemas. Together, these standards provide the principles for the definition of features. An illustration of the process of abstraction from reality to feature instances is presented in Figure 1. As illustrated, phenomena in the real world are classified into feature types which are then instantiated as features.

Figure 2 shows the OGC approach to modeling geographic features. The approach provides conceptual schemas that define abstract feature types and provide the foundation for domain experts to develop application schemas that are used to capture data about feature instances.

An application schema is a set of conceptual schema for data required by one or more applications. The application schema contains instances of types defined in the general feature model. For example, the general feature model defines the concept of feature type while an application schema defines specific feature types such as Road or Lake. The set of feature types defined for an information community are presented in a feature catalogue.

A feature catalogue contains definitions and descriptions of the feature types, feature attributes, and feature associations occurring in one or more sets of geographic data, together with any feature operations that may be applied. The feature catalogue provides the definition of geographic features at the type level, not the encoding or physical representation of individual instances of each type.

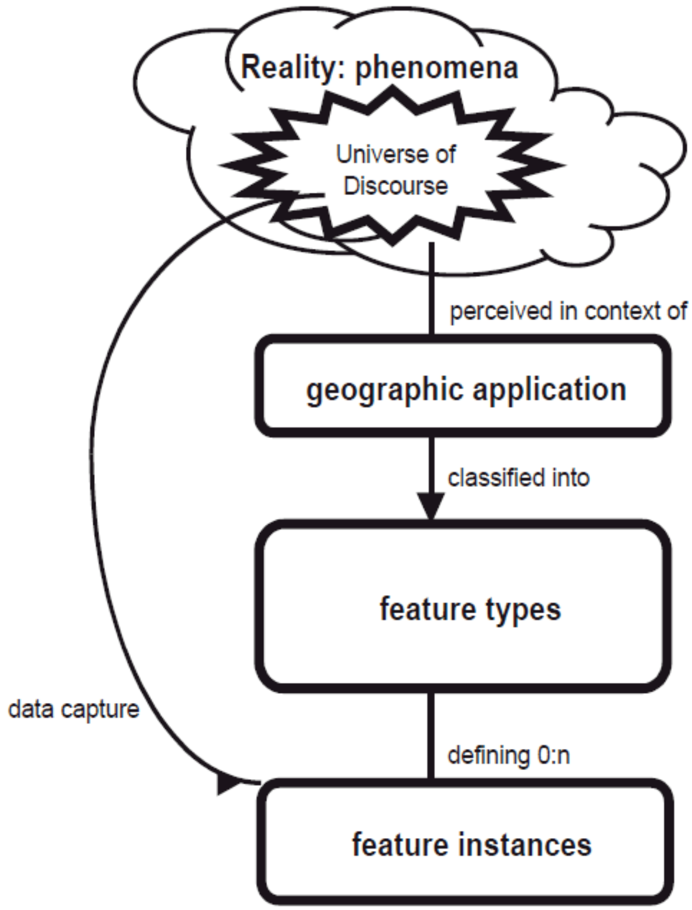

Figure 1. The process from universe of discourse to data (ISO 19109)

Within OWS-8 CCI semantic mediation, both application schema and feature catalogues play a key role. As shall be described later in this document, semantic mediation involves transforming one or more datasets into a dataset based on a different data model, which is effectively a different application schema. Further, an understanding of the definition of feature types is required for interpreting semantics of the data, therefore the feature catalogue also plays a role.

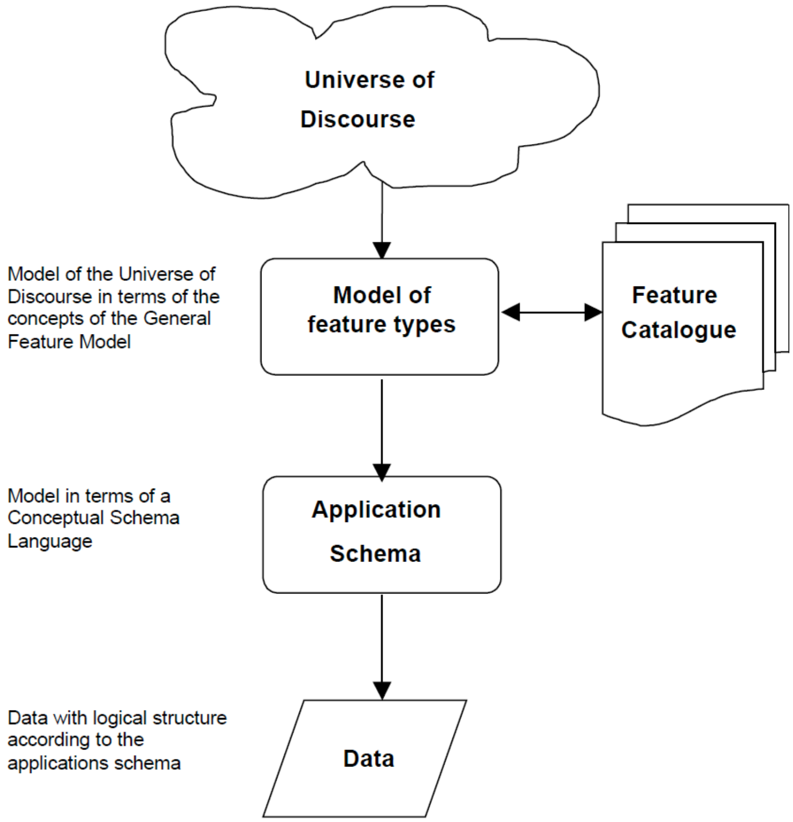

Figure 2. Modeling Geographic Information (ISO 19109)

5.2 OGC Fusion Standards Study

As part of the concept preparation of the recent OWS-7 initiative, the OGC conducted a Fusion Standards Study [OGC 09-138]. The goal of the study was to define and develop fusion standards to give analysts an environment where they can use interoperable tools to analyze, process and exploit two or more different types of data or products from the same or multiple sensors and databases from one client. Some of the objectives of the study were to develop new capabilities or exploit current capabilities for fusing information from multiple sensors, from multiple sources, and from multiple INTs in ways that dramatically improve the ability to detect, identify, locate, and track objects. The study examined sensor fusion, object/feature fusion and decision fusion. The following recommendations from both Phases 1 and 2 of the object/feature fusion component of the study have been identified as relevant to semantic mediation:

1. Define a conceptual model of feature lifecycle – beyond conflation (from Phase 1)

2. Standardize metadata for provenance and uncertainty (from Phase 1)

3. Develop common data models supporting feature fusion (from Phase 1)

4. Define a portfolio of feature fusion services (from Phase 1)
5. Develop WPS Profiles for Geoprocessing Fusion (from Phase 2)
6. Further develop rule-based geoprocessing to an OGC Best Practice (from Phase 2)
7. Registries for Object/Feature Fusion (from Phase 2)

5.3 OGC GeoSPARQL Standard

The GeoSPARQL standard, which supports representation and querying of geospatial data on the Semantic Web, is currently under development within the OGC. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data. The GeoSPARQL specification comprises several different components. A core component defines top-level RDFS/OWL classes for spatial objects. A geometry component defines RDFS data types for serializing geometry data, RDFS/OWL classes for geometry object types, geometry-related RDF properties, and non-topological spatial query functions for geometry objects. A geometry topology component defines topological query functions. A topological vocabulary component defines RDF properties for asserting topological relations between spatial objects, and a query rewrite component defines rules for transforming a simple triple pattern that tests a topological relation between two features into an equivalent query involving concrete geometries and topological query functions.

The OWS-8 CCI activity was an early adopter of the draft GeoSPARQL standard. Consequently, some aspects of the draft standard that have been included in OWS-8 may already be obsolete at the time of publishing this Engineering Report.

5.4 OWS-5 Data Conflation

In OWS-5, a data conflation process was implemented (OGC 07-160r1). The process involved conflation of two datasets with dissimilar but known schemas. One dataset was considered to be a baseline dataset, consisting of the majority of the features, and generally made up of static data, that generally did not change over time. The other dataset was considered as an "update" dataset consisting of either data that had changed in the baseline, or data that was to be added to the baseline. The output of the process was a third dataset, in the schema of the baseline dataset, where all features "duplicated" in the two datasets had been removed. The definition of "duplicated" depends on application-specific business rules. The OWS-5 conflation report identifies two approaches for feature mapping, namely on-demand mapping and ontology supported conflation. On-demand mapping involves use of schemas in defining the rules and actions for conflation. Ontology supported conflation involves translation of both incoming schemas to a common conceptual schema before the conflation process is attempted. The OWS-5 conflation work did not include ontology supported conflation but acknowledged it as an area for future research. The semantic mediation work undertaken in OWS-8 builds on the OWS-5 results by implementing a process that uses ontologies to support mediation between different datasets.

5.5 UK MoD GI2RA

In 2007 the United Kingdom Ministry of Defence (MoD) awarded Envitia (formerly known as TENET Technology Ltd) a three year research project, Geospatial Intelligence Integrated Reference Architecture (GI2RA), to investigate the future provision and dissemination of environmental or geospatial information and intelligence. A component within GI2RA involved information modeling and data harmonization. A harmonized data model was developed based on various defense formats such as Additional Military Layers (AML), Digital Aeronautical Flight Information File (DAFIF) and NGA Vector Map (VMAP). Using the Envitia Content Model Mapper (CMM), the feature types and attributes from the source data models were then mapped to the harmonized data model.

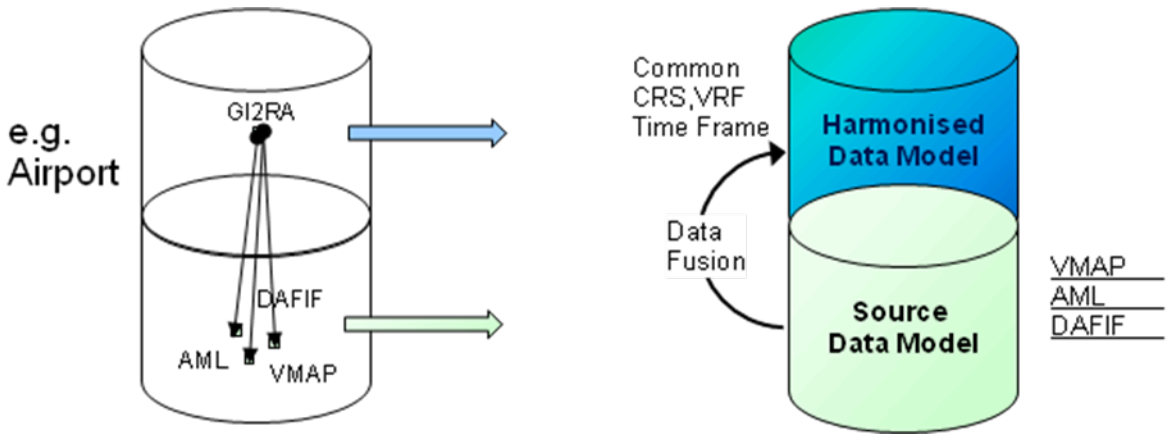

Figure 3. Concept view of the GI2RA harmonization and fusion process

6 CCI Source Data

In this section a brief description of the source datasets supplied by the thread sponsors is presented.

6.1 NGA Topographic Data Store

The Topographic Data Store (TDS) is developed by the NGA as part of the National System for GEOINT (NSG). The TDS offers an extension to the NSG Entity Catalog (NEC) that: identifies specific content of the NEC that shall be obligatory for geospatial intelligence producers using this specification, and specifies the conditions under which this geospatial intelligence shall be collected by producers for use in net-centric data exchange with other NSG participants. The NEC specifies the domain data model for feature-based geospatial intelligence that determines the common semantic content of the NSG despite varying physical realizations across DoD/IC systems (i.e., regardless of whether geospatial features are represented as an image, a multi-dimensional grid of values, or a set of one or more vector shapes). The NSG TDS Content Specification identifies the topographic content of the Geospatial Intelligence Knowledge Base (GKB) that serves as the common DoD/IC virtual geospatial information environment on the Global Information Grid (GIG). The schema classifies feature types according to themes for elevation, ports and harbours, physiography, military structures, foundation, ocean environment, aircraft facility, boundaries, hydrography, transportation, vegetation, agricultural, cultural, inlandwater, population and information.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:tds="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0"
    targetNamespace="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds
    /3.0" elementFormDefault="qualified">
    <xsd:element name="RailwayGeocurve"
      substitutionGroup="gml:_Feature"
      type="tds:RailwayGeocurveType" />
  <xsd:complexType name="RailwayGeocurveType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="geometry"
              type="gml:CurvePropertyType" />
          <xsd:element name="facilityOperationalStatus">
            <xsd:simpleType>
              <xsd:restriction base="string">
                  <xsd:enumeration
                    value="continuous" />
                  <xsd:enumeration
                    value="noInformation" />
                  <xsd:enumeration
                    value="notInOperation" />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
            [...more elements...]
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

The TDS is produced at different levels of abstraction e.g. global, regional, local and others. For OWS-8, the Local TDS (LTDS) was adopted. The LTDS contains primarily topographic features that are typically extracted to the local level at 1:50K and 1:100K scales. The LTDS is published in both Microsoft Access (ESRI Geodatabase) and OGC Geography Markup Language (GML) formats. For OWS-8, a sample LTDS dataset was supplied by NGA in GML format through a WFS hosted by Interactive Instruments GmbH. Following the GML specification, each LTDS feature type is described as an xsd:complexType within the XSD returned by the WFS DescribeFeatureType response. Each feature property is described by an xsd:element entry in the XSD. Feature properties that are constrained to specific enumerants are annotated with xsd:enumeration axioms. This means that enumerations that are reused by different feature properties are also repeated for each feature property that uses them.

6.2 USGS The National Map

The National Map (TNM) is a collaborative effort among the USGS and other Federal, State, local, and Tribal partners to improve and deliver topographic information for the Nation. With applications ranging from recreation to scientific analysis to emergency response, the National Map is one of the most comprehensive topographic data products available. The USGS publishes the National Map through web services that include online viewers and downloadable data. The geographic information available from The National Map includes orthoimagery (aerial photographs), elevation, geographic names, hydrography, boundaries, transportation, structures, and land cover. The National Map website allows for downloading data in a variety of formats include shapefiles, ESRI personal geodatabases (version 9.1 and version 10).

Simple feature class
Struct_Point

Geometry *Point*
Contains M values *No*
Contains Z values *No*

Field name	Data type	Allow nulls	Default value	Domain	Prec-ision	Scale	Length
Shape	Geometry	Yes					
OBJECTID	Object ID						
Permanent_Identifier	String	Yes					40
Source_FeatureID	String	Yes					40
Source_DatasetID	String	Yes					40
Source_DataDesc	String	Yes					100
Source_Originator	String	Yes					130
Data_Security	Short integer	Yes	0	Sec_Classification Domain	0		
Distribution_Policy	String	Yes	E4	Distribution_Policy Domain			4
LoadDate	Date	Yes			0	0	8
FType	Long integer	No	730		0		
FCode	Long integer	Yes	73002		0		
Name	String	Yes					60
IsLandmark	Long integer	Yes	2	YesNo Domain	0		
PointLocationType	Long integer	Yes	1	PointLocationType Domain	0		
AdminType	Long integer	Yes		OwnerClass Domain	0		
AddressBuildingName	String	Yes					60
Address	String	Yes					75
City	String	Yes					40
State	String	Yes					2
Zipcode	String	Yes					10
GNIS_ID	String	Yes					10
Foot_PermID	String	Yes					40
Complex_PermID	String	Yes					40
DDD_PermID	String	Yes					40

Figure 4. Table structure of the TNM Struct_Point class (source: USGS)

USGS also offers data of selected themes from the national map through WFS. The themes were redeployed by the USGS from their native database formats to WFS and GML specifications and the FGDC/ANSI Geographic Information Framework Data Content Standards. These services provide query and retrieval of specific Framework features and their attributes into a form usable for analysis in Geographic Information Systems (GIS).

The original TNM model is designed as a relational database. Each of its relational tables contains multiple feature types that share the same table structure. For example, the TNM relational table Struct_Point contains feature types for health facilities, education facilities, emergency service facilities and others. As illustrated in Figure 4, a record in the table describes an instance of the feature type indicated by the 'FType' property. This approach differs from that of the TDS which has all records in a feature collection belonging to the same feature type. For OWS-8, it was decided to extract some of the feature types into independent feature types in order to make the conceptual model of the TNM more apparent. The rest of this report refers to this revised model of the TNM.

Enumerations within the TNM are modeled as shared codelists that are reusable between multiple feature properties. This means that the TNM model does not repeat definitions of enumerations. Although, use of shared codelists between different feature properties is more efficient it was observed that WFS DescribeFeatureType responses are not able to reference shared codelists, by virtue of their XSD encoding. Therefore, in order to describe codelists for the TNM model through a WFS DescribeFeatureType response, the same approach used by the TDS would need to be adopted.

Another difference between enumerations for TNM and TDS feature properties is that enumerations for TNM properties are made up of a codelist value that is typically numeric and a label in natural language. In contrast, the enumerations for TDS properties are made up of labels in natural language only. This presents a challenge for any WFS attempting to present TNM codelists. Specifically, only the TNM codelist values would be presentable through the xsd:enumeration axiom.

7 CCI Architecture

In this section the service architecture adopted in the OWS-8 CCI thread is presented. As illustrated in Figure 5, the source data for the CCI thread was the NGA LTDS and USGS TNM products. Each source dataset was provided through a web feature service (WFS). The services are registered in a catalogue service where each service is associated with a domain ontology. These WFS services were invoked via the OGC mediation component. The mediation component not only is a WFS and CSW client, but also implements a WFS and WMS interface. The mediation component, as a CSW client, accesses a symbology registry to generate maps based on the feature type, the symbols, and rules registered for those feature types. The mediation component also translates between instances of domain models in GML and RDF, queries a knowledge base and integrates the results in a map with proper styles. The knowledge base contains the common model (Rosetta Mediation Model, RMM), ontologies representing each data model, mappings from each data model to RMM, and rules.

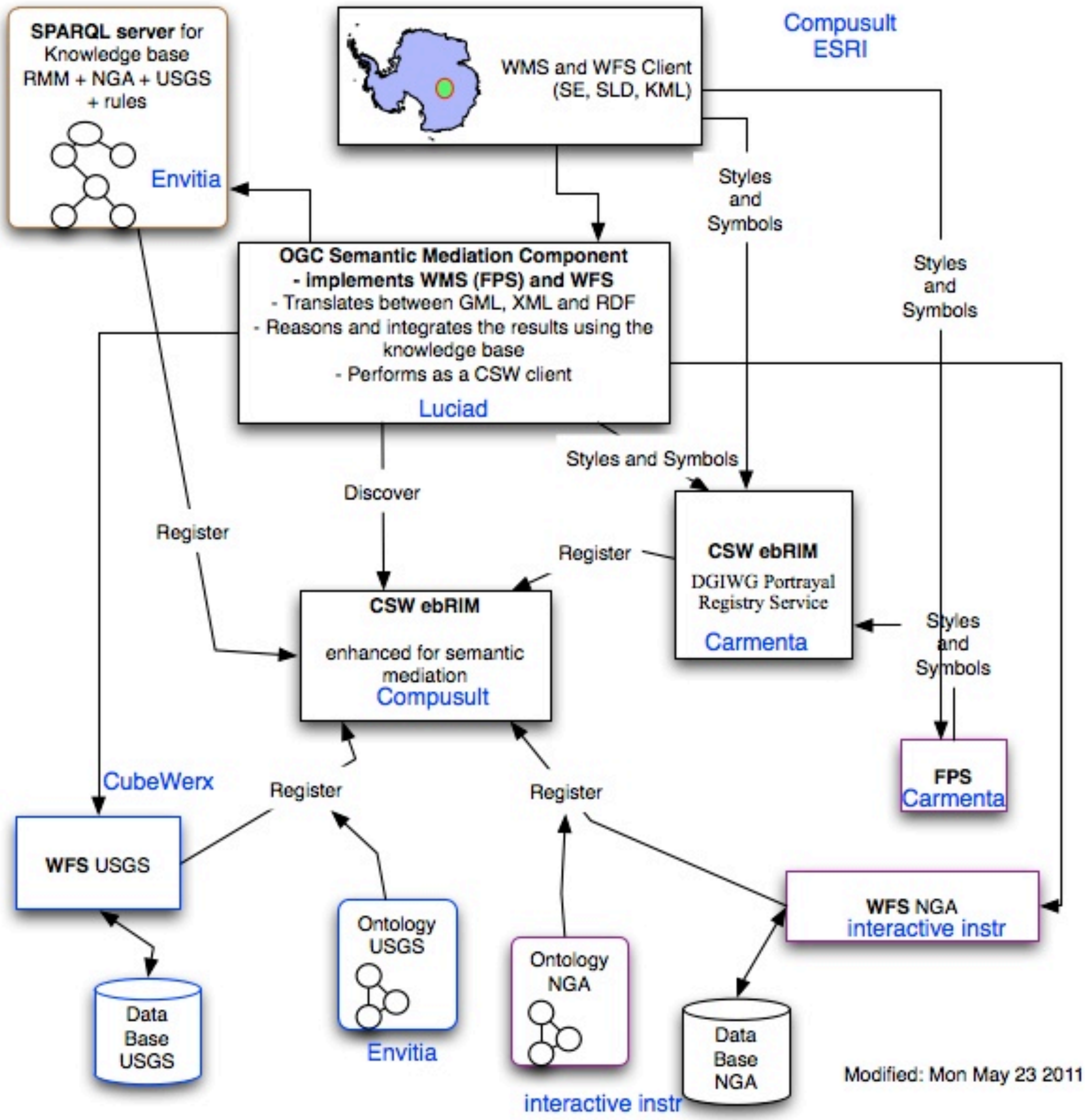

Figure 5. CCI Architecture

Within this architecture, the Web GUI Clients provide the interface between the user and the rest of the system. The CSW is the principal component for resource discovery. The WFS provide vector data supplied by the NGA and USGS. The semantic mediator harmonizes data retrieved from the WFS. The FPS renders data obtained from the semantic mediator and the WFS services. The knowledge base is the collection of RDF-encoded documents generated from mapping specifications and the RMM, TNM and TDS data models. The SPARQL server is the web application that offers a SPARQL interface for the querying of the RDF-encoded knowledge base.

7.1 Component Implementations

Participants of the CCI thread provided the following components in order to implement the CCI architecture:

- Web GUI Clients provided by Compusult and ESRI
- CSW Services provided by Compusult
- CSW-ebRIM Portrayal Registry provided by Carmenta
- FPS provided by Carmenta
- WFS USGS provided by CubeWerx
- WFS NGA provided by Interactive Instruments GmbH
- Semantic mediator provided by Luciad
- Knowledge Base and SPARQL server provided by Envitia Ltd.

8 Semantic Mediation Approach

As illustrated in the architecture diagram, there are at least two source datasets and each dataset has an ontology associated with its data model. The semantic mediator uses a common model to mediate between the two (or more source models). The following approaches were identified as candidate formulae for defining the common model:

Option 1: Create a new common model

- A new RMM would be created as the intersection of the NGA and USGS models
- NGA model would map to the RMM.
- USGS model would map to the RMM.

Option 2: Use NGA Model as the Common Model

- The common model for this thread would be the NGA model, since it has portrayal symbols and rules already defined for some feature types
- The USGS model would be mapped to the common model, i.e. the NGA model in this case.
- Two WFS services would be available from the semantic mediator services
  - WFS based on NGA model serving NGA data
  - WFS based on NGA model serving USGS data

Option 3: Use NGA Model as starting point for the Common Model augmented as needed from USGS model.

- The common model for this thread, referred to as the Surrogate RMM (RMM-S), would be based on the TDS model
- Mapping to be done are as follows:
  - TNM -> RMM-S
  - TDS -> RMM-S
  - RMM-S -> TNM
  - RMM-S-> TDS
- Four WFS services would be available:
  1. WFS based on NGA TDS model serving NGA data - provided directly by NGA
  2. WFS based on NGA TDS model serving USGS data - provided by semantic mediator
  3. WFS based on USGS TNM model serving NGA data - provided by semantic mediator
  4. WFS based on USGS TNM model serving USGS data - provided directly by USGS

The CCI thread selected Option 3, which defines a common model as the NGA TDS model augmented by that subset of the USGS TNM model that is not covered by the TDS.

8.1 Semantic Description of Data Models

In GML, feature types are defined in XML schema definition files (XSD) as complex types that extend the gml:AbstractFeatureType complex type. The properties of the feature type are represented as elements within the complex type. Geometry properties are represented in GML as elements of a type that is substitutable by the type gml:AbstractGeometry. As illustrated by the NGA TDS GML model, codelists may be implemented as enumeration restrictions on each element within a complex type. It was therefore necessary to adopt an RDF/OWL approach that could also describe feature types, properties and enumerations. With the development of the GeoSPARQL standard ongoing within the OGC, it was decided to adopt an RDF-based approach that remains consistent with the GeoSPARQL work as much as possible. The GeoSPARQL specification requires feature instances to subtype the GeoSPARQL class 'ogc:Feature'. Further, the draft GeoSPARQL standard offers representations of geometry classes.

A key difference between the representation of properties in GML schema compared to representation in RDF is that feature properties in GML are defined within the scope of a feature type through xsd:element declarations. In contrast, properties in OWL are independent of an owl:Class. An association between a property and a class in OWL is made through the rdfs:domain axiom. For a single property one can have declare multiple rdfs:domain axioms. However, multiple properties may have the same name but different restrictions if the properties belong to different feature types. Therefore it was necessary to qualify each property in OWL with the name of the feature type to which it belongs. Following the recommendation from the ISO TC 211 project 19150 that “the property name shall be composed of the class name and the attribute name using the dot (.) notation”, a convention was adopted where each xsd:element entry is represented by an individual owl:DatatypeProperty [ISO TC 211 N 2705].

Consequently, the following formula was designed:

- Each ontology should be described using an owl:Ontology axiom with minimal metadata provided by dc:title, owl:versionInfo, dc:date and dc:description properties.

- Each feature type should have the 'Feature' rdfs:Class from GeoSPARQL as a superclass.

- Each feature type should reference its parent ontology through an rdfs:isDefinedBy property that references the resource identifier(URI) of the parent ontology.

- For each feature type, create an owl:Class representing the concept abstracted by the feature type.

- For each feature property, create an owl:DatatypeProperty for relations between instances of classes and XML Schema datatypes. Alternatively create an owl:ObjectProperty for relations between instances of classes. The owl:DatatypeProperty or owl:ObjectProperty should have a name with a dot-separated concatenation of the feature type name and the property name. For example, FeatureTypeName.PropertyName.

- For each feature property, use rdfs:domain and rdfs:range to associate the property to a feature type.

- For each feature type or property, use dc:source for the qualified logical name with the convention {BaseURI}#FeatureTypeName.PropertyName.

- For each feature type or property, use dc:description to encode the description in natural language(i.e. Dublin Core).

- For each feature type include a hasGeometry or its sub-property, defaultGeometry, inherited from GeoSPARQL.

- For each codelist or enumeration, a SKOS representation of the codelist should be provided.

- Mappings should be specified in a separate file from any of the models.

The following extract presents an example declaration of an ontology.

```
<owl:Ontology rdf:about="">
     <owl:versionInfo
     rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
     >0.3</owl:versionInfo>
       <dc:date
          rdf:datatype="http://www.w3.org/2001/XMLSchema#da
          te">2011-08-04</dc:date>
       <dc:description
          rdf:datatype="http://www.w3.org/2001/XMLSchema#str
          ing">Topographic Data Store Geodatabase Schema: This
          geodatabase schema defines the logical content for
          Topographic Feature Data in the U.S. National System for
          Geospatial-Intelligence (NSG). </dc:description>
       <dc:title>NGA TDS ontology</dc:title>
      <owl:imports
          rdf:resource="http://env032011.appspot.com/geosparql.
          rdf" />
    </owl:Ontology>
  </rdf:RDF>
```

The following extract presents an example representation of a feature type.

```
<owl:Class about="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#RailwayGeocurve">
   <rdfs:label xml:lang="en">Railway Geospatial Curve</rdfs:label>
   <rdfs:subClassOf resource="http://env032011.appspot.com/geosparql.rdf#Feature" />
   <rdfs:subClassOf>
   <owl:Restriction>
   <owl:onProperty
      resource="http://env032011.appspot.com/geosparql.rdf#defaultGeometry" />
   <owl:allValuesFrom resource="http://env032011.appspot.com/geosparql.rdf#Curve" />
   </owl:Restriction>
   </rdfs:subClassOf>
   <dc:description datatype="http://www.w3.org/2001/XMLSchema#string">Railway
      Geospatial Curve: One or more railway tracks comprising a network that is operated
      for the conveyance of passengers and/or goods</dc:description>
   <dc:relation datatype="http://www.w3.org/2001/XMLSchema#anyURI">
urn:ogc:ows8:cci:topicCategory:Transportation</dc:relation>
   <rdfs:isDefinedBy rdf:resource=" "/>
</owl:Class>
```

The following extract presents an example representation of a feature property.

```
<owl:DatatypeProperty about="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#
RailwayGeocurve.facilityOperationalStatus">
 <rdfs:label xml:lang="en">
Railway Geospatial Curve : Facility Operational Status</rdfs:label>
 <dc:description datatype="http://www.w3.org/2001/XMLSchema#string">
Railway Geospatial Curve : Facility Operational Status:
The status of operation of a facility, as a whole.</dc:description>
```



```
 <skos:definition resource="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#
RailwayFacilityOperationalStatusType" />
 <rdfs:domain resource="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#
RailwayGeocurve" />
 <rdfs:range resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:DatatypeProperty>
```

The following extract presents an example representation of a codelist.

```
<skos:ConceptScheme about="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#
RailwayFacilityOperationalStatusType">
 <type resource="http://www.w3.org/2002/07/owl#Thing" />
 <skos:prefLabel datatype="http://www.w3.org/2001/XMLSchema#string">
Railway Facility Operational Status Type</skos:prefLabel>
 <skos:definition datatype="http://www.w3.org/2001/XMLSchema#string">
Railway Facility Operational Status Type: A coded domain value denoting the facility
      operational status type of a railway.</skos:definition>
</skos:ConceptScheme>
```

The following extract presents an example representation of a codelist value.

```
<skos:Concept about="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#
RailwayFacilityOperationalStatusType_continuous">
 <type resource="http://www.w3.org/2002/07/owl#Thing" />
 <skos:prefLabel datatype="http://www.w3.org/2001/XMLSchema#string">
Continuous</skos:prefLabel>
 <skos:definition datatype="http://www.w3.org/2001/XMLSchema#string">
Continuous: Operating without interruption.</skos:definition>
 <skos:altLabel datatype="http://www.w3.org/2001/XMLSchema#string">
continuous</skos:altLabel>
 <skos:inScheme resource="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#
RailwayFacilityOperationalStatusType" />
</skos:Concept>
```

Clients querying the SPARQL server were then able to compose queries based on the following example:

```
SELECT ?property ?codelist  ?codelistValue ?codelistValueLabel
WHERE  { ?property <http://www.w3.org/2000/01/rdf-schema#domain>
<http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#RailwayGeocurve>
. ?property <http://www.w3.org/2004/02/skos/core#definition>
?codelist . ?codelistValue
<http://www.w3.org/2004/02/skos/core#inScheme> ?codelist .
?codelistValue <http://www.w3.org/2004/02/skos/core#prefLabel>
?codelistValueLabel}
```

9 Semantic Mappings

Upon reviewing the feature types from the source datasets, it was observed that there potentially would be a variety of mapping scenarios for feature types. These included:

- No mapping: source feature type has no suitable match in target
- One-to-one: source feature type has one suitable match in target; all instances of the source feature type should be mapped as instances of the target feature type
- One-to-many: source feature type matches with more than one target feature type. In this case, the value of one or more properties determines which the target feature type is for a certain source instance
- Many-to-One: source feature type matches with one target feature type. However, there are other source feature types that also map to that target feature type. When transforming data from source to target, one or more properties should get a fixed value.

Similarly it was observed that feature properties would exhibit similar mapping scenarios. These included:

- Direct mapping: the property value of the source feature is directly used as the property value of the target feature
- No mapping: the source property has no suitable match in the target feature type
- Codelist mapping; Between codelists it was also observed that there could be:
  - Direct value mapping: one source code list value maps directly on one target code list value
  - No mapping: source code value has no suitable value in the target code list. In order not to lose information, the recommendation seems to be to put the value of the source code in some free TXT attribute (if available) of the target type
  - Many-to-one : multiple source code list values are mapped on a single target code list value

9.1 Ontology Mapping versus Transformation Rules

Semantic mediation can be conceptually broken into components for semantic matching and data transformation. Although the Resource Description Framework(RDF) and Web Ontology Language (OWL) are acknowledged as being capable of representing data models (specifically, feature types and properties) enough to support semantic matching, it was observed that RDF and OWL may not be expressive enough to describe data transformation rules. An example situation is where two properties from two different feature types are semantically matched to a property in a third feature type. However, the two source properties might require different transformations in order to convert them into the target property. For example, although all referring to distance, the following source properties would require different conversion factors to transform them into the target property:

Source Property A	→	Target Property C
(Distance in Metres)	(Conversion Factor 1)	(Distance in Miles)

Source Property B	→	Target Property D
(Distance in Yards)	(Conversion Factor 2)	(Distance in Miles)

A more complex transformation could be where conditional mappings (If...Then...Else) apply. The question for the CCI thread was how much of the data transformation mappings should be standardized and how much should be left to implementations. One option is to have the semantic mediator ‘decide' on the appropriate transformation to use for specific semantic matches. Another option is to use a ‘rule' language to define the appropriate transformation. There are multiple rule languages available that could be used to standardize transformation mappings, some of the examples include RIF and SPIN.

There are multiple possible approaches for describing mappings and rules within semantic mediation. For OWS-8, the CCI thread considered ontology mapping and transformation rules. This section describes these approaches.

9.1.1 Ontology Mapping

In this approach, the knowledge base provides (next to the ontologies) semantic information on how an ontology is mapped onto another ontology. The mediator uses this information from the knowledge base to create a set of object transformations that are able to transform GML features from one schema to another. In addition, the mediator also creates a set of filter transformations that can transform the OGC filter instances that are used to limit the results of WFS feature requests.

In essence, this approach simply names the semantic match between classes. The semantic mediator then applies an appropriate transformation based on the semantic match that is defined in the ontology. Some of the RDF, OWL and SKOS properties identified as applicable within this approach include skos:exactMatch, skos:closeMatch, owl:equivalentProperty and owl:equivalentClass.

It was observed however that there is no standard definition of a mapping language that is sufficiently expressive to describe all possible transformations. RDF, OWL and SKOS provide some constructs (like owl:equivalentProperty or skos:exactMatch) but these are not enough. Therefore a number of additional custom constructs need to be defined to cover transformations that are not supported by existing semantic web languages.

9.1.2 Transformation Rules

In this approach, the knowledge base provides a set of rules (next to the ontologies) that can be used to transform instances. The mediator enters these rules in an inference engine. To transform data, the mediator transforms the source GML features into RDF and feeds that into the inference engine. The transformation rules in the inference engine then ensure that a new set RDF rules is created that represents the instance in the target model. The mediator extracts that instance, converts it into GML and puts it in the resulting feature collection.

A rule is defined as a combination of an antecedent (body) and a consequent (head). Typically, a rule implies that if the antecedent is true then the consequent has to be true. Both the antecedent and the consequent may contain zero or more atomic statements. In first order logic, an atomic statement (also called an 'atom') is a statement that cannot be broken down. The presence of multiple atoms in an antecedent or consequent is interpreted as a conjunction of the atoms. Each atom is defined as a statement consisting of a triple. For example, where the symbol ● represents a conjunction between two atoms; and → represents the implication of the antecedent, the following is implied: if an entity A is the capital of entity B and the entity A has an airport called "Heathrow", then the entity B should have the name "United Kingdom".

(A capitalOf B ● A hasAirport "Heathrow") → (B hasName "United Kingdom")

RDF, OWL and SKOS are collections of triples, without the explicit definition of implications. Consequently, several rule languages have been developed for the semantic web. Examples include the JenaRules, Rule Interchange Format (RIF), N3 Logic and the SPARQL Inferencing Notation (SPIN). Across all the aforementioned rule languages, the basic form of antecedent and consequent is consistent. What differs is the expressivity and syntax adopted by each language for integrating statements.

For semantic mediation, a rule-based mechanism would need to support the use of the ontologies describing the feature types of the source datasets. Moreover, such an approach would also need to specify transformations to apply to specific objects from the source data models. Several options for the encoding of the rules were considered. These included:

- Rule Interchange Framework (RIF)
- N3 Logic Rules
- JenaRules
- SPARQL CONTRUCT queries
- SPARQL Inferencing Notation (SPIN)

9.1.2.1 Rule Interchange Format

The Rule Interchange Format (RIF) is a W3C Recommendation. RIF is a family of rule languages, called dialects, with rigorously specified syntax and semantics. RIF dialects can be classified into two groups: logic-based dialects and dialects for rules with actions. The logic-based dialects include languages that employ some kind of logic, such as first-order logic (often restricted to Horn logic). The rules-with-actions dialects include production rule systems, such as Drools, as well as event-condition-action rules, such as Reaction RuleML. Currently the only two logic dialects within RIF are the Basic Logic Dialect (RIF-BLD) and the RIF Core Dialect. The only rules-with-actions dialect defined within RIF is the Production Rule Dialect (RIF-PRD).

RIF dialects include:

- Basic Logic Dialect (BLD): BLD is designed to be simple. A BLD document consists of a number of rules. Each rule contains a set of Horn rules; existential qualification is supported, negation is not. RIF-BLD has a number of syntactical extensions to support features such as objects and frames. The main semantic extensions include datatypes and externally defined predicates.

- Production Rule Dialect (PRD): This is one of the major dialects of RIF, influenced by production rule technology that has been demonstrated by major software vendors. Production rules, as they are currently practiced in main-stream systems like Jess or JRules, are defined using ad-hoc computational mechanisms, which are not based on logic. For this reason, RIF-PRD is not part of the suite of logical RIF dialects and stands apart from them. However, significant effort has been extended to ensure interoperability with the other dialects where possible.

- Core Dialect (Core): This dialect is a subset of both RIF-BLD and RIF-PRD based on RIF-DTB 1.0, thus enabling limited rule exchange between logic rule dialects and production rules. RIF-Core corresponds to Horn logic without function symbols (often called 'Datalog') with a number of extensions to support features such as objects and frames as in F-logic, internationalized resource identifiers for concepts, and XML Schema datatypes.

- Datatypes and Built-Ins (DTB): This specification lists the datatypes, built-in functions and built-in predicates expected to be supported by RIF dialects such as the RIF-Core, RIF BLD and RIF PRD.

The following are examples of the use of RIF within the geospatial community.

- EC INSPIRE: The INSPIRE report on the state of the art in Schema Transformation services observed that there is no commonly-used interchange format for mapping definitions. The INSPIRE study on Schema Transformation Services recommended that RIF be adopted as the interchange format for mappings. The report identified RIF PRD (Production Rule Dialect) as the most suited to the combination of rule conditions and consequent actions. The INSPIRE report provides a RIF template rule that is applied to features imported from a GML document.

- OGC GeoSPARQL: In the latest revision of the GeoSPARQL draft standard, dated April 13th 2011, the OGC adopts RIF for describing query transformation rules. The GeoSPARQL draft standard provides a RIF template rule that is applied to features imported from an RDF document. The template example describes a rule for testing if two features are related.

9.1.2.2 N3 Logic

N3 Logic offers a notation for encoding triples and the inference rules [Berners-Lee, 1998]. It is designed to be compact and more readable than RDF's XML syntax. It is also designed to be more expressive than RDF. N3 has several subsets including NTriples, Turtle, N3 RDF and N3 Rules. N-Triples is a line-based, plain text format for representing correct answers to RDF/XML test cases. Turtle allows writing down an RDF graph in a compact textual form. N3 RDF offers a constrained language such that only correct RDF graphs can be defined. N3 Rules offers a language for defining rules with both an antecedent and consequent. Of the Rule languages under examination in OWS-8, N3 is one of the most mature and hence there are free and open source implementations available. One such implementation is Euler which is available in Java, C#, Python and other languages.

9.1.2.3 JenaRules

An alternative to the aforementioned approaches is to use JenaRules , supported by the Jena API [HP]. Jena is a Java library for handling RDF and provides the foundation for several Java-based semantic web applications such as ARQ, Pellet and others. Jena allows for querying, writing and reasoning over an RDF graph. The syntax and grammar of JenaRules is based on N3.

9.1.2.4 SPARQL CONSTRUCT

SPARQL is a language developed for querying RDF-encoded triple stores. It is a Recommendation of the W3C. It supports querying of triple patterns with conjunctions, disjunctions, value filters, and optional patterns. A separate W3C recommendation describes an XML format for the results returned by a SPARQL query. SPARQL results can be returned, alternatively, encoded in RDF. SPARQL offers a SELECT form which can be used to return a subset of the triple store. SPARQL also offers a CONSTRUCT query which returns an RDF graph formatted according to a graph template. The returned graph is formed by substituting the variables in the graph template with the subject, predicate or objects from the returned triples. The CONSTRUCT form is functionally equivalent to logical rules in that the construct template is equivalent to a consequent(head) and the WHERE clause equivalent to the antecedent(body). The following screenshots show the SPARQL queries equivalent to those presented in the N3 rule examples above. The only difference is that the N3 rules are provided in a single document, whereas the SPARQL queries are documented as separately.

9.1.2.5 SPARQL Inferencing Notation (SPIN)

The SPIN Modeling Vocabulary is a light-weight collection of RDF properties and classes to support the use of SPARQL to specify rules and logical constraints. Based on an RDF representation of SPARQL queries, SPIN defines three class description properties:

spin:constraint can be used to define conditions that all members of a class must fulfill. spin:rule can be used to specify inference rules using SPARQL CONSTRUCTs and DELETE/INSERTs. A spin:constructor can be used to initialize new instances with default values. In addition to these class description properties, SPIN provides a meta-modeling capability that can be used to build a modeling language and SPARQL extensions. The meta-modeling capabilities provide the ability to encapsulate reusable SPARQL queries into templates, and to derive new SPARQL functions as well as magic properties from other SPARQL queries and functions. TopQuadrant offers an open source API for SPIN.

SPIN inherits it rule capabilities from SPARQL. The key benefit of SPIN is from its ability to persist rules within the data the rules apply to. Therefore for the purpose of OWS-8, the compact nature of alternative rule languages such as N3 and JenaRules outweighs this benefit of SPIN.

9.2 Selected Approach for Semantic Mappings

After careful consideration of the various approaches for transformation rules and the ontology mappings approach, the ontology mappings approach was selected as most appropriate for the OWS-8 CCI work. The main reasoning behind this was because the ontology approach would allow the semantic mediator to create inverse filter mappings for queries to the source WFS. Filtered queries to the source WFS result in better performance as the source WFS has to return less feature instances during GetFeature requests.

In order to specify the ontology mappings, a structure for Microsoft Excel spreadsheets and an RDF documents was designed. This section describes the spreadsheet and RDF encoding for mappings.

Each worksheet in the mappings spreadsheet contains the following columns:

- sourceFeatureType: This is the natural name of the source feature type.
- sourceProperty: This is the natural name of a source feature property.
- sourceEnumerant: This is a value in a source codelist or enumeration.
- action1: Describes a type of mapping between the source and target objects.
- action2: Describes another type of mapping between the source and target objects.
- targetFeatureType: This is the natural name of the target feature type.
- targetProperty1 : This is the natural name of the target feature property.
- targetEnumerant1: This is a value in a target codelist or enumeration.
- targetLiteralValue1: This is a literal value to be assigned to a target feature property.
- targetProperty2 : This is the natural name of another target feature property.
- targetEnumerant2: This is a value in another target codelist or enumeration.

- targetLiteralValue2: This is a literal value to be assigned to another target feature property.

Based on the values in these columns, an RDF document can be encoded with a single row in the spreadsheet decomposed into two mapping specifications. One mapping specification describes how the source property relates to the first target property; the other mapping specification describes how the source property relates to the second target property. This ‘decomposition’ allows for any number of target properties to be assigned to a single source property. For rapid conversion from Excel to RDF, it is recommended that the columns in the spreadsheet be duplicated, with the duplicates offering resource identifiers (URLs) from the USGS and NGA ontologies. The following extracts show an RDF encoding of a single row from the Excel spreadsheet. It should be noted that the target enumerant and literal value are represented in the same RDF property, namely ows8:targetValue.

```
<ows8:Mapping rdf:about="http://env032011.appspot.com/ows8.rdf#mapping_206_1">
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
 <ows8:mappingType rdf:resource="http://env032011.appspot.com/ows8.rdf#
sameAsEnumerant" />
 <ows8:targetFeatureType rdf:resource="http://env032011.appspot.com/rmm.rdf#
InlandWaterbodyGeosurface" />
 <ows8:targetProperty rdf:resource="http://env032011.appspot.com/rmm.rdf#
InlandWaterbodyGeosurface.waterResourceInfo.hydrologicPersistence" />
 <ows8:targetValue rdf:resource="http://env032011.appspot.com/rmm.rdf#
InlandWaterbodyWaterResourceInfoHydrologicPersistenceType_perennial" />
 <ows8:sourceFeatureType rdf:resource="http://www.usgs.gov/projects/ows8#
LakePondGeosurface" />
 <ows8:sourceProperty rdf:resource="http://www.usgs.gov/projects/ows8#
LakePondGeosurface.featureCode" />
 <ows8:sourceValue rdf:resource="http://www.usgs.gov/projects/ows8#
NhdWaterbodyFeatureCodeType_lakePondIntermDateOfPhoto" />
</ows8:Mapping>
```

```
<ows8:Mapping rdf:about="http://env032011.appspot.com/ows8.rdf#mapping_206_2">
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
 <ows8:targetValue>NHD: Lake/Pond: Hydrographic Category = Perennial; Stage = Date of
     Photography</ows8:targetValue>
 <ows8:mappingType rdf:resource="http://env032011.appspot.com/ows8.rdf#
sameAsEnumerant" />
 <ows8:targetFeatureType rdf:resource="http://env032011.appspot.com/rmm.rdf#
InlandWaterbodyGeosurface" />
 <ows8:targetProperty rdf:resource="http://env032011.appspot.com/rmm.rdf#
InlandWaterbodyGeosurface.note.memorandum" />
 <ows8:sourceFeatureType rdf:resource="http://www.usgs.gov/projects/ows8#
LakePondGeosurface" />
 <ows8:sourceProperty rdf:resource="http://www.usgs.gov/projects/ows8#
LakePondGeosurface.featureCode" />
 <ows8:sourceValue rdf:resource="http://www.usgs.gov/projects/ows8#
NhdWaterbodyFeatureCodeType_lakePondIntermDateOfPhoto" />
</ows8:Mapping>
```


The CCI thread also considered whether RDF, OWL and SKOS could enable the specification of semantic matches and the required data transformation function. It was observed that RDF, OWL and SKOS have a limited vocabulary for describing semantic relationships between objects. For example, SKOS offers 14 semantic relation types, i.e. those that inherit directly or indirectly from skos:semanticRelation. The semantic relation types describe the degree to which the meanings of two objects are similar. For discovery purposes, such semantic relationships are adequate. However, for semantic mediation involving data remodeling, additional information is required to enable a mediator to apply an appropriate conversion to input data. It was concluded that RDF, OWL and SKOS did not offer a vocabulary expressive enough to describe data transformation.

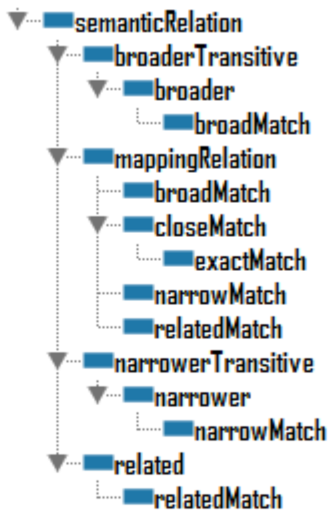

Figure 6. Semantic relation hierarchy of SKOS

A vocabulary of actions to apply to source properties was provided in the OWS-8 CCI mappings spreadsheet. The vocabulary addresses limitations of the properties offered by OWL and SKOS. Specifically, the vocabulary of action types offers a mixture of semantic relations and transformation types. The following values are offered by the vocabulary:

- convertKMtoM: Specifies conversion from kilometers to meters.
- convertKM2toM2: Specifies conversion from square kilometers to square meters.
- firstComponentOf: Specifies that the value of a property should be the first in a collection.
- secondComponentOf: Specifies that the value of a property should be the second in a collection.
- thirdComponentOf: Specifies that the value of a property should be the third in a collection.

- fourthComponentOf: Specifies that the value of a property should be the fourth in a collection.

- fifthComponentOf: Specifies that the value of a property should be the fifth in a collection.

- generalizeEnumerant: Specifies that the domain of a property is a generalization of another.

- sameAsEnumerant: Specifies that an enumerant (i.e. codelist value) is the same as another enumerant (from a different codelist). This mapping type is different from the owl:sameAs and skos:exactMatch constructs because it can be used to associate enumerants declared as skos:Concept constructs to literal values that have not been declared in the ontology.

- sameAsProperty: Specifies that two properties from different feature types refer to the same characteristic, although the properties may hold different values. This mapping type is different from the owl:equivalentProperty construct which is used to state that two properties have the same property extension (i.e. they hold the same "values"). This mapping type applies to both geometric and alphanumeric properties.

- replace: Specifies that a target property or enumerant is a replacement of a source property or enumerant respectively. This relation may only be used if the source and target constructs do not mean the same thing although serving a similar purpose. For example, the source enumerant "unknown" may relate to the target enumerant "not applicable" through a "replace" mapping; however, source enumerant "NA" would relate to the target enumerant "not applicable" through a "sameAsEnumerant" mapping.

In a scenario where a user requests an NGA RailwayGeocurve, the semantic mediator would send the following request to the knowledge base:

```
DESCRIBE  ?a  WHERE

{

   ?a            <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://env032011.appspot.com/ows8.rdf#Mapping> .

   ?a    <http://env032011.appspot.com/ows8.rdf#targetFeatureType>
<http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#RailwayGeocurve>

}
```

The SPARQL query above requests a description of all mappings for USGS feature type LightRailGeocurve. The response from the request specifies that the NGA RailwayGeocurve feature type maps to the USGS LightRailGeocurve feature type. Based on the response, the semantic mediator can then apply the appropriate transformation.

9.3 Coordinate Reference Systems

An instance of a feature may include a geometric property that uses a named coordinate reference system (CRS). Both the GML and GeoSPARQL specifications support the referencing of CRS through URI and URN. The most popular registry for CRS is the EPSG database, developed by the European Petroleum Survey Group. The EPSG organization has now been superseded by the International Association of Oil & Gas producers (OGP) however the registry of CRS continues to be referred to through its historic EPSG name. The following extract is taken from the OGC policy on names [OGC 09-048r3]:

"http URI form:

http://www.opengis.net/def/objectType/EPSG/0/code

URN form:

urn:ogc:def:objectType:EPSG::code

In this case, the authority part of the URI is 'EPSG'. The code part of the URI is the EPSG code unique identifier of the referenced definition. Alternately, the code part of the URI can be the EPSG name unique identifier. In this case, omission of the version number is recommended, as this is not required to identify a referenced record in the EPSG dataset and may even lead to confusion if a version number is provided."

9.3.1 CRS Encoding in RDF

Within the CCI thread, the TNM data supplied by the USGS adopted the NAD83 CRS whereas the LTDS data supplied by the NGA adopted the WGS84 CRS. An RDF/OWL encoding was developed for describing these CRS. For each CRS, an owl:Class describing a URI, natural name and additional identifiers is provided. Each CRS description may include a text-based definition of its geodetic parameters in OGC Well Known Text (WKT). Alternatively, a URI to a file containing a GML-encoded description of the CRS may be included. The following extract describes the WGS84 CRS using a multiple URI, URN and WKT:

```
<ows8:CRS rdf:about="http://www.opengis.net/def/crs/EPSG/0/4326">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
<ows8:otherIdentifier
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EPSG:4
    326</ows8:otherIdentifier>
<ows8:asWKT
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">GEOGC
    S["WGS 84",DATUM["WGS_1984",SPHEROID["WGS
    84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORI
    TY["EPSG","6326"]],PRIMEM["Greenwich",0,AUTHORITY["EPSG","89
    01"]],UNIT["degree",0.017453292519943328,AUTHORITY["EPSG","91
    22"]],AUTHORITY["EPSG","4326"]]</ows8:asWKT>
<ows8:name
```



```
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">WGS84
    </ows8:name>
<ows8:otherIdentifier
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">urn:o
    gc:def:crs:EPSG::4326</ows8:otherIdentifier>
<ows8:otherIdentifier
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http:
    //metadata.ces.mil/mdr/ns/GSIP/crs/WGS84E_2D</ows8:otherIdenti
    fier>
</ows8:CRS>
```

The following extract describes the WGS84 longitude-latitude CRS using a URI, URN and a reference to a remotely hosted GML file:

```
<ows8:CRS
    rdf:about="http://www.opengis.net/def/crs/OGC/1.3/CRS84">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
<ows8:name
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CRS8
</ows8:name>
<ows8:otherName
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">WGS
    84 longitude-latitude</ows8:otherName>
<ows8:asGML
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http:
    //www.opengis.net/def/crs/OGC/1.3/CRS84.gml</ows8:asGML>
<ows8:otherIdentifier
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">urn:o
    gc:def:crs:OGC:1.3:CRS84</ows8:otherIdentifier>
</ows8:CRS>
```

9.3.2 Lessons Learnt

It was observed that within the OWS-8 CCI architecture it is necessary for a semantic mediator to provide metadata on whether it supports coordinate transformations or whether it relies on coordinate transformations by the WFS. Such metadata could be provided as part of the GetCapabilities response document. A typical scenario would be for the coordinate reference system parameter received by the semantic mediator to be forwarded to the WFS services. However, it is possible that the source WFS may not always support the same coordinate reference systems. Therefore, some implementations of the semantic mediator may apply coordinate transformations on-the-fly. It is therefore necessary for the GetCapabilities response of the semantic mediator to be able to include information on its coordinate transformation capabilities.

10 SPARQL Server

The SPARQL server is a web application that allows for querying RDF documents through SPARQL. The SPARQL server relies on a knowledge base stored as RDF triples in a database. The knowledge base can store multiple RDF documents even if they describe different ontologies. Due to this flexibility, it is necessary for the ontologies offered through the SPARQL server to also capture a minimal set of metadata such as the unique identifier (URI), title, version and date. Such metadata provides a mechanism for distinguishing between ontologies stored in the knowledge base.

10.1.1 Interface

Client applications send requests to the SPARQL server through the HTTP Get method. Each HTTP Get request contains a SPARQL query in URL-safe UTF-8 text. Responses to SPARQL SELECT queries are encoded in the SPARQL Results XML format, whereas responses to CONSTRUCT and DESCRIBE queries are encoded in RDF/XML. An illustration of the SPARQL Results XML format is presented in the following listing:

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="a" />
  </head>
  <results>
   <result>
     <binding name="a">
       <uri>http://../tds/3.0#RailwayGeocurve.trackInfo.railwayGaugeClass</uri>
     </binding>
   </result>
   <result>
     <binding name="a">
       <uri>http://../tds/3.0#RailwayGeocurve.trackOrLaneCount</uri>
     </binding>
   </result>
  </results>
</sparql>
```

10.1.2 Lessons Learnt

There are immediate opportunities to define OWS-based requests/responses for the SPARQL server. Such requests/responses should be considered in the context of both the semantic mediation work and the GeoSPARQL work. Following requirements from the common specification for OWS, the following capabilities could be applied in the SPARQL server:

- GetCapabilities: This operation could offer the identifiers of the ontologies offered by the SPARQL server (i.e. as defined by the <owl:Ontology> element). Based on the RDF model adopted for OWS-8, the following SPARQL query could be used to generate a GetCapabilities document that lists all ontologies offered by a SPARQL server.

```
SELECT ?ontologyURI ?ontologyTitle WHERE  {
?ontologyURI <http://purl.org/dc/elements/1.1/title> ?ontologyTitle .
}
```

- DescribeOntology: This operation could present all OWL axioms that have the property rdfs:isDefinedBy referencing the owl:Ontology being described. Based on the RDF model adopted for OWS-8, the following SPARQL query could be used to generate a listing of all OWL axioms that reference the owl:Ontology entries returned by the GetCapabilities response.

```
SELECT ?object ?ontologyURI ?ontologyTitle WHERE  {
?ontologyURI <http://purl.org/dc/elements/1.1/title> ?ontologyTitle .
?object
<http://www.w3.org/2000/01/rdf-schema#isDefinedBy> ?ontologyURI .
}
```

- Error reporting: The return of an ows:ExceptionReport, in the event of an error, could be supported.

- Although not supported in the current implementation of the SPARQL server, the SPARQL specification offers an optional SOAP binding. The SPARQL request and response elements contained within the SOAP body have names with hyphens. Although OGC does not explicitly prohibit the use of hyphens in element names, it is not a typical naming convention. Therefore, future OGC standardisation of a SPARQL server would need to provide element definitions that are more consistent with OGC naming conventions.

  The following listing presents an example SOAP request based on the SPARQL specification:

```xml
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <query-request xmlns="http://www.w3.org/2005/09/sparql-protocol-types/#">
      <query>SELECT ?z {?x ?y ?z . FILTER regex(?z, 'Harry')}</query>
    </query-request>
  </soapenv:Body>
</soapenv:Envelope>
```

  The following listing presents an example SOAP response based on the SPARQL specification:

```xml
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
```

```xml
<query-result xmlns="http://www.w3.org/2005/09/sparql-protocol-types/#">
<ns1:sparql xmlns:ns1="http://www.w3.org/2005/sparql-results#">
<ns1:head>
<ns1:variable name="z" />
</ns1:head>
<ns1:results distinct="false" ordered="false">
<ns1:result>
<ns1:binding name="z">
<ns1:literal>Harry Potter and the Chamber of Secrets</ns1:literal>
</ns1:binding>
</ns1:result>
<ns1:result>
<ns1:binding name="z">
<ns1:literal>Harry Potter and the Half-Blood Prince</ns1:literal>
</ns1:binding>
</ns1:result>
<ns1:result>
<ns1:binding name="z">
<ns1:literal>Harry Potter and the Goblet of Fire</ns1:literal>
</ns1:binding>
</ns1:result>
<ns1:result>
<ns1:binding name="z">
<ns1:literal>Harry Potter and the Philosopher's Stone</ns1:literal>
</ns1:binding>
</ns1:result>
<ns1:result>
<ns1:binding name="z">
<ns1:literal>Harry Potter and the Order of the Phoenix</ns1:literal>
</ns1:binding>
</ns1:result>
<ns1:result>
<ns1:binding name="z">
<ns1:literal>Harry Potter and the Prisoner Of Azkaban</ns1:literal>
</ns1:binding>
</ns1:result>
</ns1:results>
</ns1:sparql>
</query-result>
</soapenv:Body>
</soapenv:Envelope>
```


11 Semantic Mediator

The Semantic Mediator is a web service capable of mediating between the NGA TDS and USGS TNM data models and providing access to the data through OGC web services.

Using a common data model (RMM-S) as described in Chapter 8 and a knowledge base describing the mapping to this data model, the Semantic Mediator enables the transformation between data models without having the need to provide one-to-one mappings. Within the CCI thread, this approach is used to transform NGA TDS data to USGS TNM data and vice versa.

To allow existing NGA TDS or USGS TNM clients work transparently with both data sets, two OGC Web Feature Service (WFS) interfaces are defined:

- An OGC WFS publishing the USGS TNM source data according to the NGA TDS data model;
- An OGC WFS publishing the NGA TDS source data according to the USGS TNM data model.

Additionally, the Semantic Mediator offers an OGC Feature Portrayal Service (FPS), enabling clients to access the data as images. Users of the FPS can supply an OGC Styled Layer Descriptor (SLD) to select a WFS data source (NGA or USGS) and choose a rendering style. The FPS loads and renders the data accordingly, and returns the results as an image.

To better explain the mediation approach applied by the Semantic Mediator, Figure 7 illustrates the work flow between an incoming WFS request and an outgoing response with GML data. It should be noted that Figure 7 presents a subset of the CCI workflow. The complete CCI semantic mediation workflow is presented as use cases in Section 13.1.

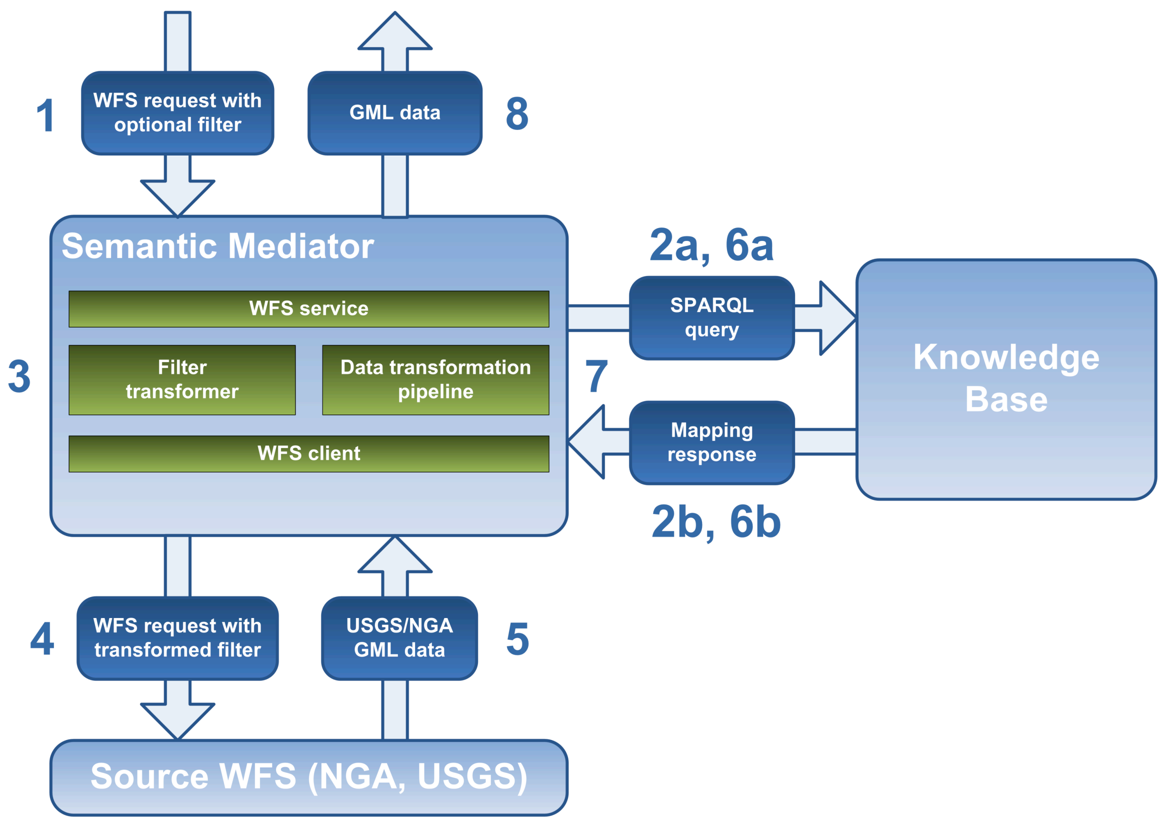

Figure 7. Semantic Mediator architecture (subset)

In the preceding diagram, the following steps are defined:

1. A WFS client connects with the Semantic Mediator and performs a WFS GetFeature request, optionally associated with an OGC Filter.

2. Communication with the Knowledge Base (SPARQL Server) to transform the OGC Filter.

   a. The Semantic Mediator converts the WFS/OGC Filter request into a SPARQL query and sends it to the Knowledge Base

The Semantic Mediator sends a SPARQL query to the Knowledge Base to retrieve the relevant mapping information for transforming the given client WFS filter to a source WFS filter.

   b. The Knowledge Base replies with a mapping in RDF.

3. The Semantic Mediator converts the OGC Filter based upon the mapping in RDF.

4. The GetFeature request with the (optional) transformed OGC Filter is sent to one of the source WFS servers (NGA or USGS).

5. The WFS server replies with USGS or NGA data in GML.

6. Communication with the Knowledge Base (SPARQL Server) to transform the response.

   a. The Semantic Mediator generates a SPARQL query and sends it to the Knowledge Base.

The Semantic Mediator sends a SPARQL query to the Knowledge Base to retrieve the relevant mapping information for transforming the GML data from the source WFS into GML data that is compatible with the client's data model.

   b. The Knowledge Base replies with a mapping in RDF.

7. The Semantic Mediator transforms the GML data based on the mapping.

8. The transformed GML data is sent to the client.

11.1 Transforming client WFS requests into source WFS requests

Each semantic mediator server is initialized with a preprocessing step that analyses the mappings found on the knowledge base server at startup time. This is needed because the semantic mediator builds a local cache of which source feature types map to which target feature types. This local cache avoids querying the knowledge base more than once during a single feature instance retrieval.

At this point it is important to note the difference between source and target feature types: A target feature type is a feature type that is requested by a client of the semantic mediator, a source feature type is a feature type that appears on the source WFS. While all transformations happen from source to feature type, it is also necessary to have mappings from target to source as this is the only way to query the correct feature types on the source WFS.

While all data transformations happen from source to target feature type, mappings from target to source feature types are also needed to transform client WFS requests (step 1) into corresponding source WFS requests (step 4).

11.2 Handling Feature Mappings

As discussed in chapter 9, there are several possible feature mapping types. This subsection describes how each type of feature mapping is handled by the semantic mediator.

- No mapping: If no mapping is present, the semantic mediator will not add the target feature type in the GetCapabilities of the WFS server. GetFeature requests to these features will not return anything.

- One-to-one mapping: This case is handled by mapping the target requested feature type to the source feature type, and propagating the request to the source WFS server.

- Many-to-one mapping: This case can be handled by mapping the requested target feature type to its corresponding source feature types, and sending the WFS server a request for all these source types. Since all source types map to the requested target type, the union of all these source type instances will correspond to the requested target feature type instances.

- One-to-many mapping: This case is handled as described in section 11.2.1.

11.2.1 Special Handling for One-to-Many Mappings

The semantic mediator receives target feature type requests from clients. The target feature types in these requests are transformed to source feature types before the requests are sent to the source WFS server. For the case when a single source feature type maps to several target feature types, it is important to select the right source feature instances to map. The knowledge base offers a collection of mappings from source to target feature types, but the semantic mediator still needs to infer which mappings apply to which feature types.

To clarify the problem, let us define the following feature types:

- A = tds:BuildingGeopoint

- B = usgs:PublicAttractLandmarkBldgGeopoint

- C = usgs:WaterSupplyTreatStructureGeopoint

In this example, A maps to (among others) B and C. Now assume a client that requests all instances of B. The semantic mediator can map this type to source type A and request all instances of A from the source WFS server using a WFS GetFeature request. The problem here is identifying which returned instances of A are in fact B and which are in fact C. (In the assumption that they are mutually exclusive, and therefor can't be both at the same time).

This problem was solved by making the assumption that when all mappings for a given source and target feature type are analyzed, certain properties could be identified as being key to deciding whether A maps to B or C. This is done by requesting all mappings with A as a source, using the following SPARQL query on the knowledge base:

```
  DESCRIBE ?a  WHERE {

  ?a <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://env032011.appspot.com/ows8.rdf#Mapping> .

  ?a <http://env032011.appspot.com/ows8.rdf#sourceFeatureType>
<http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#BuildingGeopoint
>

  }
```

For each returned sourceProperty that has a given sourceValue, a list is created for all targetFeatureType possibilities. For some sourceProperties, this will be both B and C, for others it will be only B or C. Any property that only maps to a single feature (or less than the amount of targetFeatureType possibilities) is used during the transformation step to determine what the actual target feature is for any source feature instance. These properties are referred to as "*critical properties*": These are properties that can distinguish which target feature type is mapped to when multiple candidates are present.

To increase the performance of the semantic mediation, the critical properties, along with their values are used to create OGC filters in step 4 of Figure 7 to ensure the source WFS only returns source feature types that actually map to the requested target feature type.

To increase the performance of the semantic mediator, the transformed OGC filter in step 4 of Figure 7 is extended with an additional ruleset that filters on the values of the critical properties, to ensure the source WFS only returns source feature types that actually map to the requested target feature type.

For the specific example in this section, a critical property of A is "featureFunction-1". If the value of this property is "sewage", or "waterTreatment", A maps to C. If the value of this property is "amusement", "recreation", "theatre" … then A maps to B. An example of a non-critical property would be "controllingAuthority", which can be either "private" or "public" and is applicable for both B and C. The table below gives an overview of the source properties, their values and their possible target feature types.

Source Property	Source Value	Possible Target Feature Types
featureFunction-1	sewage	C
featureFunction-1	waterTreatment	C
featureFunction-1	amusement	B
featureFunction-1	recreation	B
featureFunction-1	theatre	B
controllingAuthority	private	B,C
controllingAuthority	public	B,C

The problem with this heuristic is that it might fail for more complex data types that require multiple decisions in order to arrive at an appropriate transformation. It did however not pose any problems for the context of this OWS thread. Rule-based languages, such as RIF, that are able to model multiple decisions may be more appropriate for handling more complex data types.

11.3 Using the RMM-S as a mediating model

Up until now, this chapter has not yet covered the additional transformation that is performed when transforming source feature type to target feature type. In general, the following steps should be performed:

1. Client sends request for target feature type to semantic mediator
2. The semantic mediator transforms target feature type to one or more feature types in the mediating model (RMM-S).
3. The semantic mediator transforms each feature type of the previous step to one or more source feature types.
4. The semantic mediator sends a request to the source WFS using the collection of source feature types.
5. The semantic mediator receives a list of feature instances, and transforms them to feature instances of the mediating model (RMM-S)

6. The semantic mediator takes the feature instances found in the previous step, and transforms them to feature instances of the target feature types.

7. The semantic mediator sends the data to the client.

Note that when performing steps 2 and 3, attention should be paid to the special case when multiple source feature type map to one target feature type, as described in section 0. The problem does have increased complexity due to the fact that the input for step 3 will include the critical properties (along with their values) found in step 2.

11.4 Lessons Learnt

The Semantic Mediator queries the Knowledge Base to learn about the mapping between feature types of the USGS and NGA models. Such a mapping is defined in RDF and should be applied by the Semantic Mediator on the GML data. The feature types are identified in the RDF data with a URI, while in GML they are defined by a localname and a namespace. The Semantic Mediator should therefore need to be able to detect this difference to interpret and apply the mapping. To solve this, it was decided to use GMLnamespace#localname in the URI of a feature type in RDF, following the best practices from W3C.

The mappings offered by the SPARQL server do not change frequently. Therefore, it is reasonable for the semantic mediator to harvest and cache the mappings at scheduled times. Such a caching approach can improve performance as evidenced in several CSW and WMS implementations. To enable the semantic mediator to detect when an ontology has been revised, metadata is added to the owl:versionInfo and owl:priorVersion properties of the owl:Ontology axiom. The value of owl:versionInfo is a literal, whereas the value of owl:priorVersion is a URI reference. This is because the owl:versionInfo axiom refers to the URI described by the enclosing OWL axiom, such as owl:Ontology. In order to be able to distinguish between different versions of the same ontology using owl:priorVersion, it is therefore necessary for the URI identifying an ontology to include versioning information. Such an approach would also be consistent with OGC naming of GML since version 3.2, the namespace of which includes the version number.

12 Resource Discovery in CCI

12.1.1 Registry Catalogue Service

The Catalogue service standard offers the primary interface for resource discovery in the OGC baseline. In addition to offering an HTTP binding, the catalogue service standard also offers CORBA and Z39.50 bindings. The HTTP binding of the Catalogue service standard is also referred to as the Catalogue Service for the Web (CSW). The OGC has developed a profile for the CSW based on the ebXML Registry Information Model (ebRIM) developed by the Organisation for the Advancement of Structured Information Standards (OASIS). The OGC and OASIS have a mutual relationship where the OGC is a member of OASIS and conversely OASIS is a member of the OGC.

The CSW was used within the CCI Thread as a means of resource discovery. All participating components were published to this registry allowing other components to discover service or data information for interoperability. In addition, the catalogue provided the primary means for user discovery of service and data components used for the CCI scenario. What follows is a brief overview of the Registry and how the primary catalogued entities are stored and retrieved.

The CSW interface was based upon the OGC Catalogue Services Specification Version 2.0.2. The interaction between a client and a server was accomplished using a standard request-response model of the HTTP protocol. That is, a client sends a request to the server using HTTP, and expects to receive a response to the request or an exception message. An XML HTTP **POST** operation is performed.

The service repository resides in an Oracle database and has been implemented using version 3 of the ebRIM standard by OASIS [OASIS]. Repository service access is based upon the HTTP protocol with messaging between the client and server using XML. Client applications can use this interface for executing service repository queries and receiving service repository metadata results. Basically, the essential purpose of a catalogue service is to enable a user to locate, access, and make use of resources in an open, distributed system by providing facilities for retrieving, storing, and managing many kinds of resource descriptions that conform to any standard Internet media type, such as:

- XML schemas;
- thumbnail representations of remotely sensed images;
- audio annotations;
- specification documents;
- a simple map of a meteorological sensor network; and,
- style sheets for generating detailed topographic maps.

Within the CCI thread, feature types were catalogued and annotated with themes based on the ISO 19115 topic categories. Such cataloguing and annotation of feature types was intended to enable clients to discovery which feature types are supported within the OWS-8 CCI infrastructure. Furthermore, arbitrary relationships among cataloged items can be expressed by creating links between any two resource descriptions. For example, a service offer may be associated with the data sets that can be acquired using the service. Relationships between feature types offered by the WFS and concepts described in the ontologies were also catalogued in order to facilitate the querying of the SPARQL server by the Semantic Mediator.

A catalogue can function as a stand-alone service or it can interact with other affiliated catalogs within a federation that spans multiple administrative domains. The federation then effectively enlarges the total search space, within which resource descriptions may be discovered.

When a catalogue is linked to a peer catalogue, it makes the resource descriptions managed by the peer implicitly available to its own clients. Each catalogue client connects to a single catalogue service as its main point of contact within the federation. This is the agent node; the propagation of request messages to neighboring nodes is invisible to the client. It is not necessary to know where the metadata repositories are located or how they are accessed. The CSW standard is intended to provide a flexible, general-purpose catalogue service that can be adapted to meet the needs of diverse communities of practice within the geospatial domain.

12.1.2 Service Metadata

Within the CCI architecture, service metadata was stored in and accessed from the Service Manager Service Registry. Service metadata was automatically ingested from OGC service capabilities documents using the GetCapabilities operation on the specified service. In addition, service metadata could be manually published in the system to represent interfaces to Legacy/Heritage systems or other applications that do not support OGC interfaces e.g. the SPARQL server used to serve semantic mappings.

Service metadata was captured in ISO 19119 and is defined by the following major categories:

- Service: Distinct part of the functionality that is provided by an entity through interfaces.
- Interface: Named set of operations that characterize the behavior of an entity.
- Operation: Specification of a transformation or query that an object may be called to execute. It has a name and a list of parameters.

Services are accessible through a set of interfaces that provide a set of operations. The aggregation of interfaces in a service defines the functionality provided by the service. The combination of services to achieve results specific to a task is enabled within ISO 19119 through service chaining and enables users to combine data and services in ways that are not pre-defined by the data or service providers.

Through the Service Manager, a user can search the service metadata catalogue to find instances of a specific, well-known service type. For example, WCS, WMS, WFS or others. The service metadata is presented in XML documents.

Within the CCI thread, the following service types were catalogued:

- WFS 2.0
- WFS 1.1
- SPARQL Server
- Portrayal Registry Service (CSW)

- FPS (WMS)
- Semantic Mediator

Within the semantic mediation use cases, client components may query the CSW for metadata about the semantic mediator, FPS, SPARQL server, NGA or USGS WFS. The CSW returns service metadata that includes the service endpoint thereby enabling the clients to bind to the service being described by the metadata. The following sample request returns all services registered in the CSW:

```
<csw:GetRecords xmlns:csw="...">
<csw:Query typeNames="csw:Record rim:Service">
  <csw:ElementSetName
typeNames="rim:Service">full</csw:ElementSetName>
  </csw:Query>
</csw:GetRecords>
```

Lacking a complete capabilities document, the SPARQL server does not provide a publishing consumer such as the CSW publishing mechanism with enough metadata to indicate what rules it has available. It would be of some benefit as the SPARQL standards evolve inside of OGC for the SPARQL server definition to have support for an OWS capabilities document that indicates what sort of mappings it provides.

Rule sets could be considered data elements associated with the SPARQL server and be catalogued as objects in the catalogue.

If this was the case, after publication of multiple SPARQL servers, any Registry client could query not only for SPARQL servers, but also by what rules/dialects etc. they provided.

12.1.3 Feature Catalogue Metadata

Dataset metadata describes data holdings. Metadata elements and schema are used by data producers to characterize their geographic data. Metadata enables the use of geographic data in the most efficient way by knowing its basic characteristics. Metadata facilitates data discovery, retrieval and reuse. Metadata enable users to determine whether geographic data in a holding will be of use to them. Metadata is applicable to independent datasets, aggregations of datasets, individual geographic features, and the various classes of objects that comprise a feature.

Metadata Standards supported by the CCI CSW catalogue components included:

- ISO 19115 - This international standard provides an abstract structure for describing digital geographic data by defining metadata elements and establishing a common set of metadata terminology, definitions and extension procedures. This specification is meant to enhance interoperability by providing a common specification for describing, validating and exchanging metadata about geographic datasets.

- FGDC GEO Profile Metadata Standard - This Geospatial Metadata Profile is based on ANSI/NISO Z39.50-1995 Information Retrieval (Z39.50): Application Service

Definition and Protocol Specification. The GEO Profile includes not only the specifications for Z39.50 in the application but also other aspects of a GEO-conformant server that are outside the scope of Z39.50. The Content Standard for Digital Geospatial Metadata was developed by the Federal Geographic Data Committee (FGDC) and the ASTM Section D18.01.05 on Mapping and GIS to provide a standard set of data element references for digital, geo-referenced spatial (or “geospatial”) information.

- DDMS - The Department of Defense Discovery Metadata Standard (DDMS) defines discovery metadata elements for resources posted to community and organizational shared spaces. Discovery is the ability to locate data assets through a consistent and flexible search. The DDMS specifies a set of information fields that are to be used to describe any data or service asset that is made known to the Enterprise. The DDMS is employed consistently across the Department’s disciplines, domains and data formats.

When a WFS is registered with the Catalogue, in addition to the service level metadata, the feature type metadata is stored. The Feature metadata stored with the Registry included the xml namespace, name and topic category. This is explicitly done so that the mediation components can search for source data by these elements. Results from these queries allowed the mediation component to retrieve enough information about these feature elements so that it could query the correct WFS and feature type for feature information to translate. An example query for retrieving all feature types belonging to a specific classification (e.g. urn:ogc:ows8:cci:topicCategory:Transportation) is shown in the following listing:

```
  <?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<csw:GetRecords xmlns….>
  <csw:Query typeNames="csw:Record rim:Classification">
      <csw:ElementSetName
         typeNames="csw:Record">full</csw:ElementSetName>
     <csw:Constraint version="2.0.0">
       <ogc:Filter>
         <ogc:And>
           <ogc:PropertyIsEqualTo>
           <ogc:PropertyName>
           dc:identifier
           </ogc:PropertyName>
              <ogc:PropertyName>
           rim:Classification/@classifiedObject
              </ogc:PropertyName>
             </ogc:PropertyIsEqualTo>
             <ogc:PropertyIsEqualTo>
             <ogc:PropertyName>
                 rim:Classification/@classificationNode
              </ogc:PropertyName>
              <ogc:Literal>
        urn:ogc:ows8:cci:topicCategory:Transportation
```



```
            </ogc:Literal>
           </ogc:PropertyIsEqualTo>
         </ogc:And>
       </ogc:Filter>
     </csw:Constraint>
   </csw:Query>
 </csw:GetRecords>
```

For example, when the NGA WFS is published, the Feature Type “tds:AircraftHangarGeopoint” is stored as the Feature Type name. The associated namespace for the name - http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0 is also stored. The last item is stored as a “Slot” in the ebRim data model. The mediator, when required to retrieve data that conforms to this name and namespace, can query the registry for services that have associated feature types that conform to these criteria.

13 Scenario

The OWS-8 initiative adopted an emergency management scenario based on the aftermath of an earthquake. This chapter describes the semantic mediation aspect of the scenario.

Within this scenario, a Command Centre (CC) has been set up to coordinate emergency services. A Very Important Person (VIP) is evaluating the area affected by the earthquake. The VIP would like to visit the hospitals receiving residents affected by the earthquake. A CC operator has been tasked with preparing a ‘Situation Report’ that identifies hospitals, transportation facilities and utilities.

Data can be portrayed in different views, depending on the preference (profile) of the user. A view is composed of a model and set of styles and rules applied to that model. This scenario includes two views: a USGS view and an NGA view. This scenario also includes two data sources: USGS and NGA. An operator will want to select one view to view both datasets and get a harmonized Shared Situational Awareness (SSA).

13.1 Use Cases

This section describes use cases adopted for implementing the OWS-8 CCI scenario.

13.1.1 Initial Setup for Emergency Management Preparedness

The following use case describes initial setup and preparation for emergency management. This particular use case is only illustrative and has not been built into the OWS-8 demo.

Title	Initial Setup for Emergency Management Preparedness

Description	This use case describes a process for preparing resources required for semantic mediation. The use case focuses on the preparation of the RDF-encoded ontologies, mappings and their associated metadata.
Actors:	CC operator (Primary), SPARQL server, CSW, Portal
Trigger:	A CC operator is tasked with preparing data and web services required for emergency management.
Preconditions:	• TNM data is available through the USGS WFS • TDS data is available through the NGA WFS
Post conditions:	• RDF-encoded schemas and mappings are available from a SPARQL server • Metadata about the SPARQL server and feature types is available from the CSW
Basic Flow (Steps):	1. The CC operator converts TDS XSD from the NGA WFS to RDF. 2. The CC operator converts TNM XSD from the USGS WFS to RDF. 3. The CC operator generates an RDF-encoded RMM model to mediate between the TDS and TNM data models. 4. The CC operator uploads the generated RMM, TDS and TNM RDF into the SPARQL server. 5. The CC operator designs mappings from the TDS schema to the RMM, and vice versa, in a spreadsheet format. 6. The CC operator designs mappings from the TNM schema to the RMM, and vice versa, in a spreadsheet format. 7. The CC operator converts the mappings to RDF. 8. The CC operator uploads the RDF-encoded mappings into the SPARQL server. 9. The CC operator queries the SPARQL server for information about its service endpoint, the TDS and TNM feature types and generates the associated metadata. 10. The CC operator uploads the metadata into the CSW. 11. The CC operator configures the Portal to query the FPS. 12. The CC operator configures the FPS to query the Semantic Mediator, NGA WFS and USGS WFS. 13. The CC operator configures the Semantic Mediator to query the SPARQL server.

13.1.2 NGA Viewpoint

The following use case describes the semantic mediation workflow with an NGA viewpoint as the output. This use case was implemented and demonstrated during the OWS-8 demo.

Title	Semantic Mediation Towards an NGA Viewpoint
Description	This use case describes the process of semantic mediation as implemented for OWS-8. It focuses on the generation of an NGA viewpoint through a portrayal of TDS and harmonized TNM data.
Actors:	CC operator (Primary), Portal, Semantic Mediator, USGS WFS, NGA WFS, SPARQL server, CSW, Portrayal Registry
Trigger:	An earthquake occurs and a CC operator is tasked with preparing a situation report identifying the best health centers and roads in the affected area.
Preconditions:	- TNM data is available through the USGS WFS - TDS data is available through the NGA WFS - The semantic mediator has access to the USGS WFS and NGA WFS. - RDF-encoded schemas and mappings are available from a SPARQL server - Metadata about the SPARQL server and feature types is available from the CSW - TDS and TNM Symbology is registered on the Portrayal Registry
Post conditions:	- A portrayal of TDS data is presented through the portal. - A portrayal of harmonized TNM data is presented through the portal.
Basic Flow (Steps):	1. The CC operator launches the OWS-8 CCI client portal. 2. The CC operator selects NGA viewpoint (profile). 3. The Portal requests a base map of the area affected by the earthquake from the XYZ WMS. 4. The Portal requests a list of classification categories from the CSW. 5. The CSW returns the list of classification categories to the Portal. 6. The CC operator selects categories for Health centers and Roads. 7. The Portal retrieves SLDs and Symbology Encoding from the Portrayal Registry. 8. The Portal sends requests for TDS data and harmonized TNM data to the FPS. 9. The FPS sends a request for TDS data to the NGA WFS. 10. The NGA WFS returns TDS data to the FPS. 11. The FPS sends a request for harmonized TNM data to the Semantic Mediator. 12. The Semantic Mediator retrieves mappings from the SPARQL server. 13. The Semantic Mediator retrieves TNM data from the USGS WFS. 14. The Semantic Mediator harmonizes the TNM data according to mappings obtained from the SPARQL server.

	15. The Semantic Mediator returns harmonized TNM data to the FPS. 16. The FPS returns portrayals of the TDS data and harmonized TNM data to the Portal. 17. The Portal presents the list of feature types that match the categories of Health centers and others as Roads from the two data sources, but both symbolized following NGA TDS symbology.
Notes	The XYZ WMS is any web map service selected by a participating vendor to provide a base map.

A sequence diagram showing the requests and responses sent between components is presented in Figure 8.

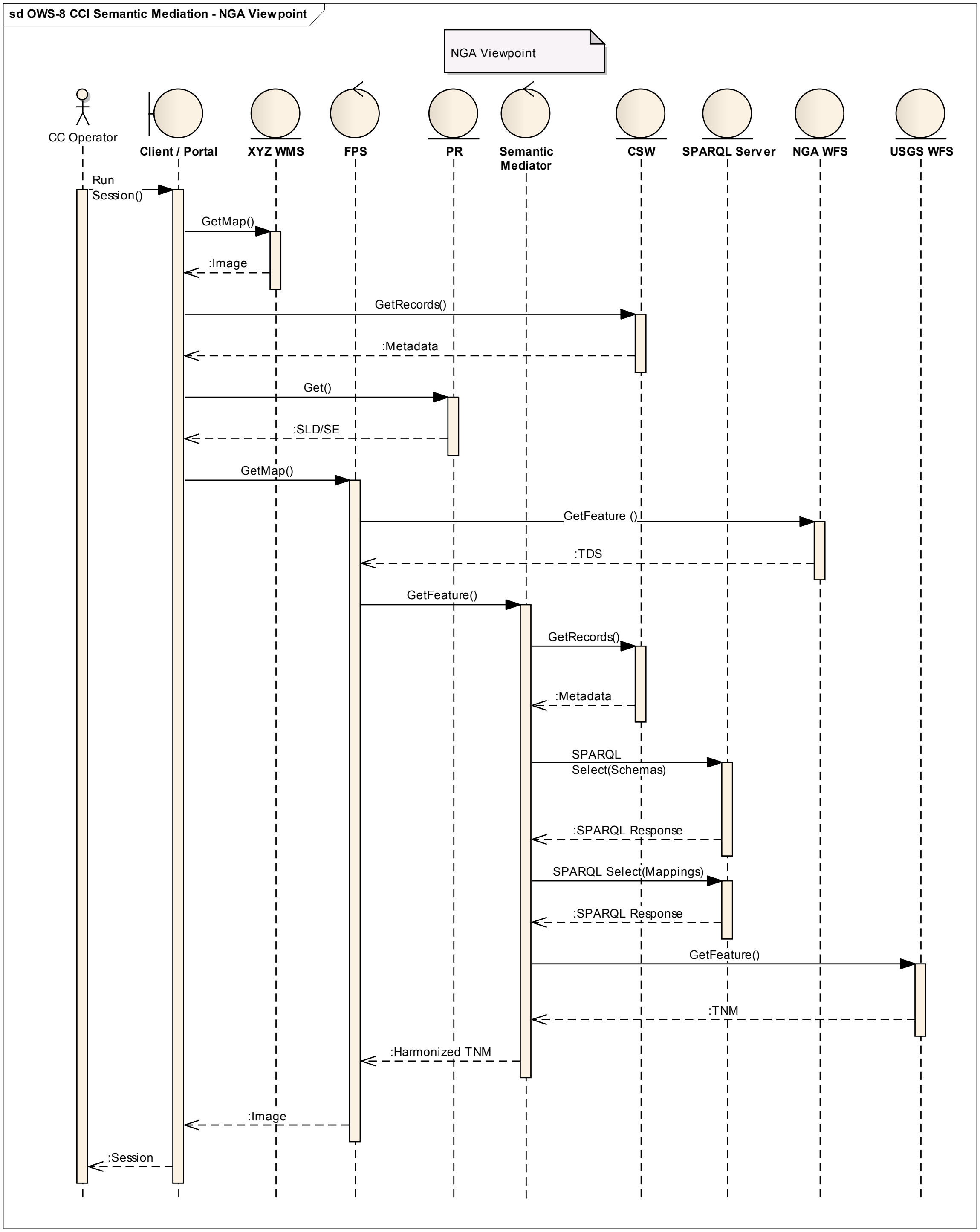

Figure 8. Sequence diagram for generating the NGA viewpoint

13.1.3 USGS Viewpoint

The following use case describes the semantic mediation workflow with a USGS viewpoint as the output. This use case was implemented and demonstrated during the OWS-8 demo.

Title	Semantic Mediation Towards a USGS Viewpoint
Description	This use case describes the process of semantic mediation as implemented for OWS-8. It focuses on the generation of a USGS viewpoint through a portrayal of TNM and harmonized TDS data.
Actors:	CC operator (Primary), Portal, Semantic Mediator, USGS WFS, NGA WFS, SPARQL server, CSW, Portrayal Registry
Trigger:	An earthquake occurs and a CC operator is tasked with preparing a situation report identifying the best health centers and roads in the affected area.
Preconditions:	• TNM data is available through the USGS WFS • TDS data is available through the NGA WFS • The semantic mediator has access to the USGS WFS and NGA WFS. • RDF-encoded schemas and mappings are available from a SPARQL server • Metadata about the SPARQL server and feature types is available from the CSW • TDS and TNM Symbology is registered on the Portrayal Registry
Post conditions:	• A portrayal of TNM data is presented through the portal. • A portrayal of harmonized TDS data is presented through the portal.
Basic Flow (Steps):	1. The CC operator launches the OWS-8 CCI client portal. 2. The CC operator selects USGS viewpoint (profile). 3. The Portal requests a base map of the area affected by the earthquake from the XYZ WMS. 4. The Portal requests a list of classification categories from the CSW. 5. The CSW returns the list of classification categories to the Portal. 6. The CC operator selects categories for Health centers and Roads. 7. The Portal retrieves SLDs and Symbology Encoding from the Portrayal Registry. 8. The Portal sends requests for TNM data and harmonized TDS data to the FPS. 9. The FPS sends a request for TNM data to the USGS WFS. 10. The USFS WFS returns TNM data to the FPS. 11. The FPS sends a request for harmonized TDS data to the Semantic Mediator.

	12. The Semantic Mediator retrieves mappings from the SPARQL server. 13. The Semantic Mediator retrieves TDS data from the NGA WFS. 14. The Semantic Mediator harmonizes the TDS data according to mappings obtained from the SPARQL server. 15. The Semantic Mediator returns harmonized TDS data to the FPS. 16. The FPS returns portrayals of the TNM data and the harmonized TDS data to the Portal. 17. The Portal presents the list of feature types that match the categories of Health centers and others as Roads from the two data sources, but both symbolized following USGS TNM symbology.
Notes	The XYZ WMS is any web map service selected by a participating vendor to provide a base map.

A sequence diagram showing the requests and responses sent between components is presented in Figure 9. Considering that the TNM and TDS data models are significantly different, the similarity of the processes presented in the sequence diagrams suggests that the architecture adopted for the OWS-8 CCI work is repeatable and independent of the source data models.

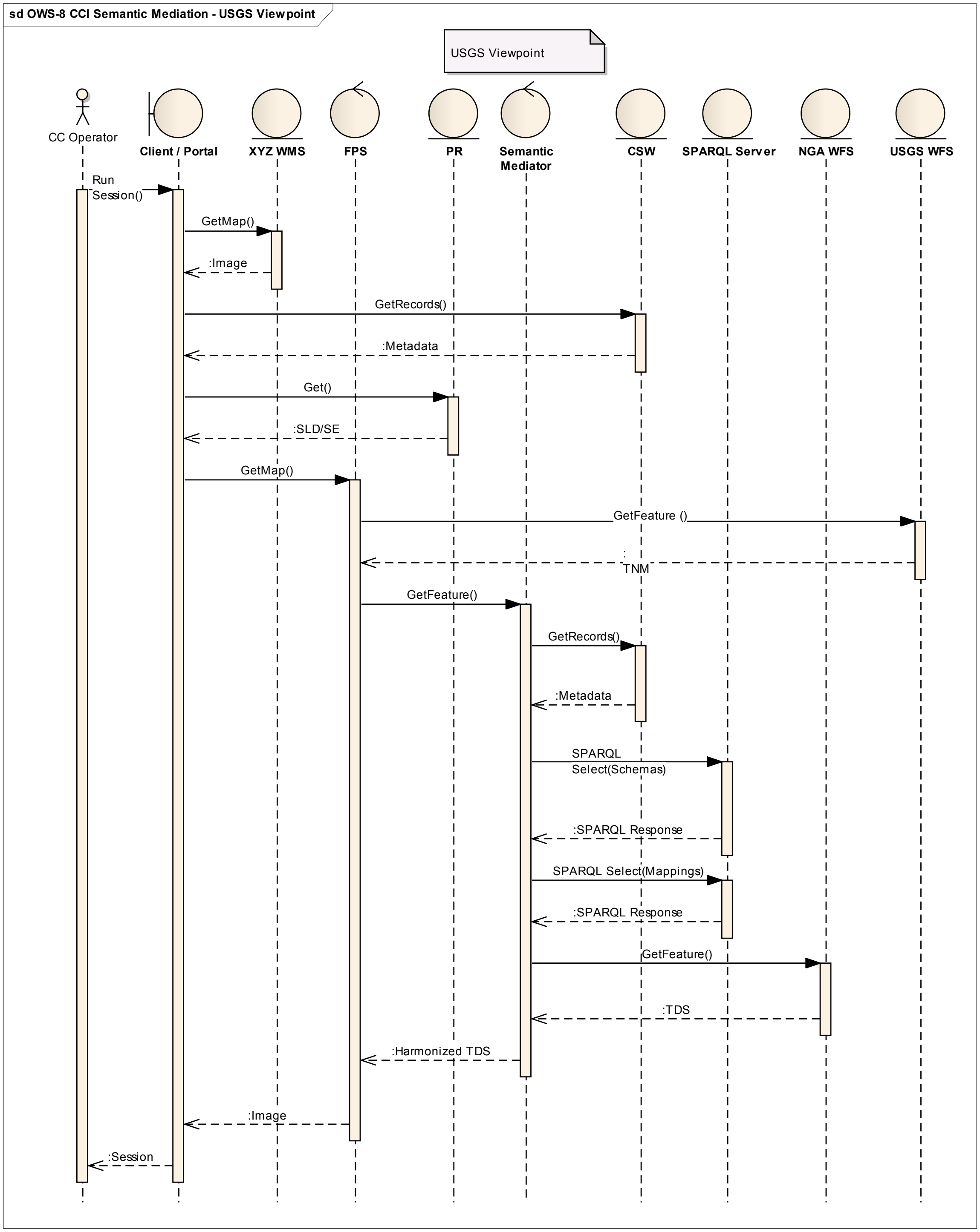

Figure 9. Sequence diagram for generating the USGS viewpoint

13.2 Portrayal of the Harmonized Data

The outputs of the semantic mediator, USGS and NGA WFS services were rendered by the FPS and presented through a browser-based client. The following figures show a portrayal of some of the harmonized data as presented by different the clients.

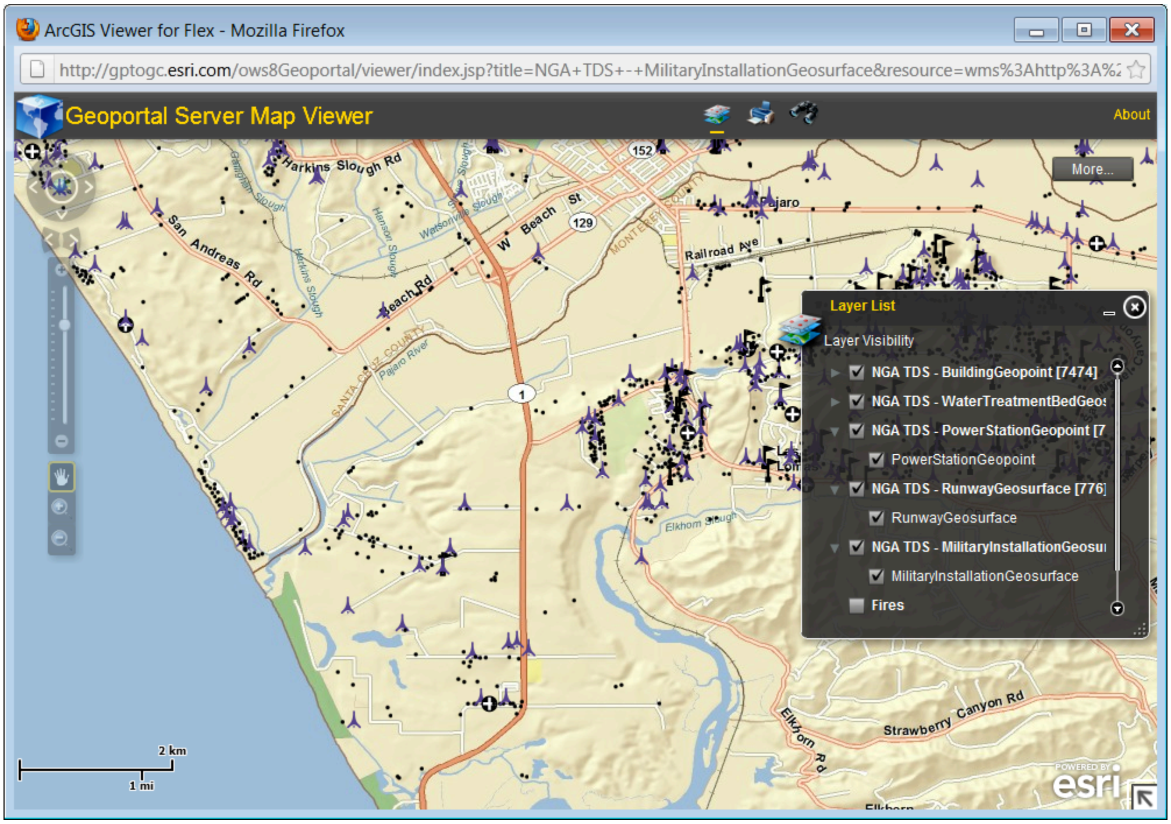

Figure 10. A portrayal of some of the harmonized data through the ESRI client

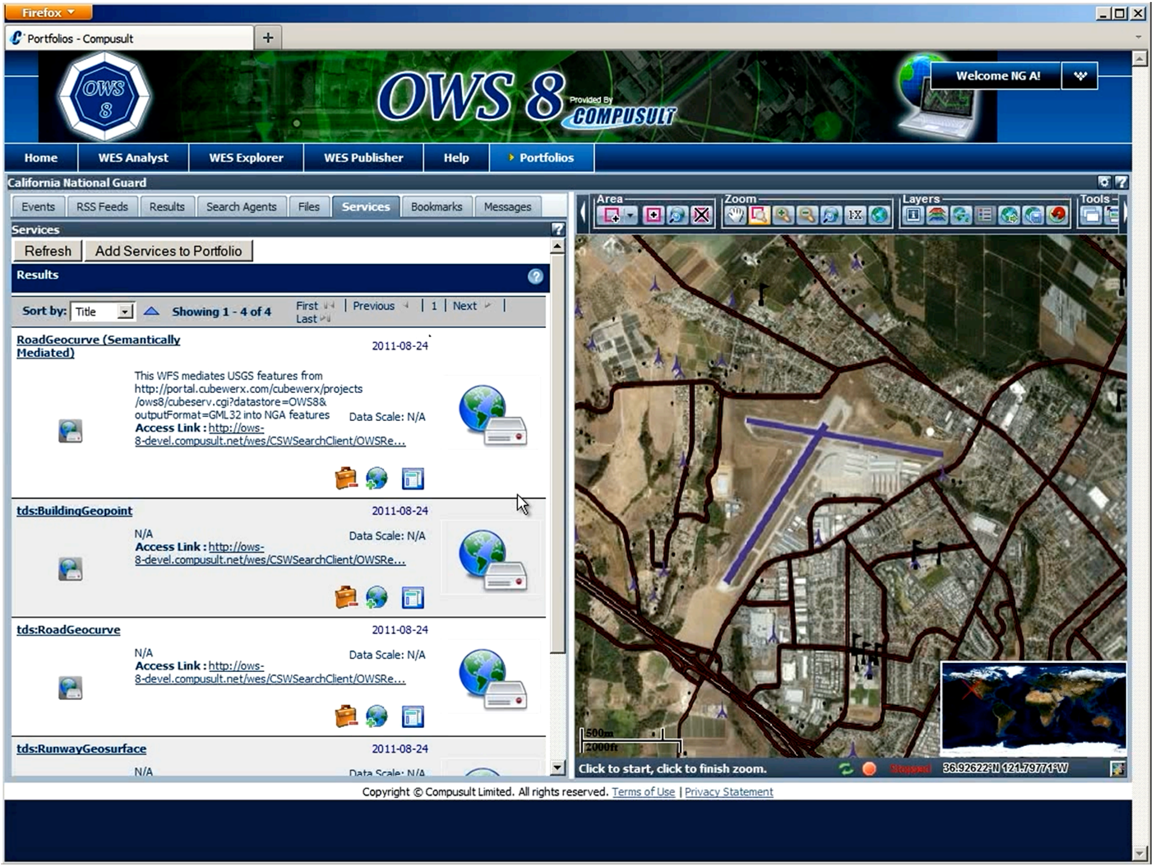

Figure 11. A combination of unmodified and harmonized data presented through the Compusult client

A more detailed discussion of portrayal issues is presented in the OWS-8 Portrayal Registries Engineering Report (OGC 11-062).

14 Conclusions and Recommendations

The OWS-8 CCI thread was able to develop RDF-encoded ontologies of the NGA TDS and USGS TNM data models and a mediating model called the RMM-S. A semantic mediation component that allows a client to query multiple data sources was implemented and offered through an interface conformant to the OGC Web Feature Service standard. Mappings from TNM-to-RMM-to-TDS and others from TDS-to-RMM-to-TNM were developed. Mappings between different representations of the same coordinate reference systems were defined. Finally, the mappings were applied in a demonstrator based on the OWS-8 CCI architecture. It can therefore be concluded that OGC standards can successfully support semantic mediation through the architecture proposed in the OWS-8 CFP.

Based on the experiences and implementations of the CCI thread, the following recommendations are made for future OGC activities:

- Development of a standard for a SPARQL server

- Development of a standard for an OGC semantic mediation service, possibly through extension of the WFS standard
- Development of a standard for encoding mappings based on the OWS-8 approach

Appendix A: Interoperability with the Content Model Mapper format

This appendix describes an experiment conducted by Envitia to show that the RDF-encoded mappings developed by the OWS-8 CCI thread are interoperable with the Content Model Mapper (CMM), a file format used by the UK MoD in the GI2RA project. An extract of a CMM file is shown in the following listing.

```xml
<ContentModelMapping>
<name>Mapping from USGS_CM to NGA_CM</name>
<FeatureMappingItem srcID="RailGeocurve" destID="RailwayGeocurve"
fmiIndex="1">
<DirectMappingAttribute srcID="RailGeocurve.lengthKilometer"
destID="RailwayGeocurve.length" />
<DirectMappingAttribute srcID="RailGeocurve.permanentIdentifier"
destID="RailwayGeocurve.uniqueEntityIdentifier" />
<DirectMappingAttribute srcID="RailGeocurve.railClass"
destID="RailwayGeocurve.railwayClass">
<DirectMappingEnum srcKey="99" destKey="No Information">
<srcName>99</srcName>
<destName>No Information</destName>
</DirectMappingEnum>
<DirectMappingEnum srcKey="2" destKey="Branch-line">
<srcName>2</srcName>
<destName>Branch-line</destName>
</DirectMappingEnum>
<DirectMappingEnum srcKey="3" destKey="High Speed Rail">
<srcName>3</srcName>
<destName>High Speed Rail</destName>
</DirectMappingEnum>
<DirectMappingEnum srcKey="1" destKey="Main Line">
<srcName>1</srcName>
<destName>Main Line</destName>
</DirectMappingEnum>
</DirectMappingAttribute>
</FeatureMappingItem>
</ContentModelMapping>
```

A.1 Implementation

CMM mappings describe associations between content models, which are descriptions of feature types, properties and enumerations defined in a schema. The CMM mappings are therefore similar to the OWS-8 CCI mappings although encoded differently. To demonstrate repeatability of the harmonization, the RDF-encoded ontologies and mappings were harvested from the SPARQL server and converted into content models and CMM mappings. The CMM mappings were then applied to TNM RoadSegmentGeocurve data. An illustration of the harmonization process is shown in Figure 12.

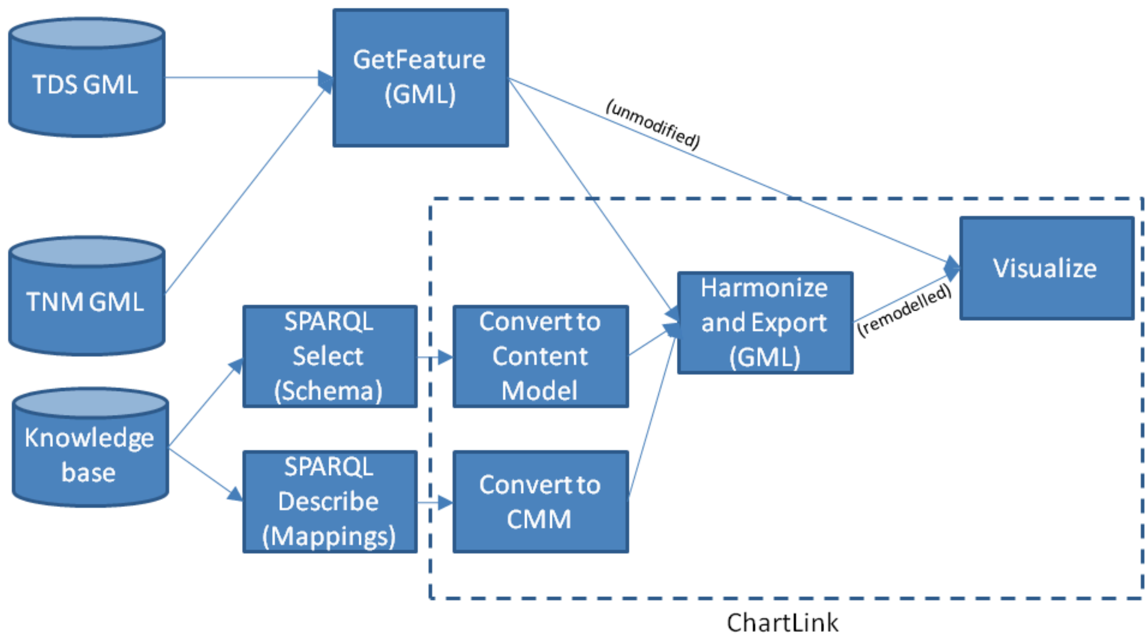

Figure 12. Harmonization process using the CMM

A.2 Results

A visualization of the output is presented in Figure 13. The figure shows TDS road networks (in black), unmodified TNM roads (in green) and remodelled TNM road networks (in red). To make it easier to spot overlapping networks, the TDS networks have been styled with thicker lines. It can be seen from the screenshot that there are roads available in the harmonized TNM data that are missing from the TDS data. This is because the supplied NGA dataset offered a subset of the roads offered by the USGS dataset. This shows that the TDS and harmonized TNM dataset can be complementary.

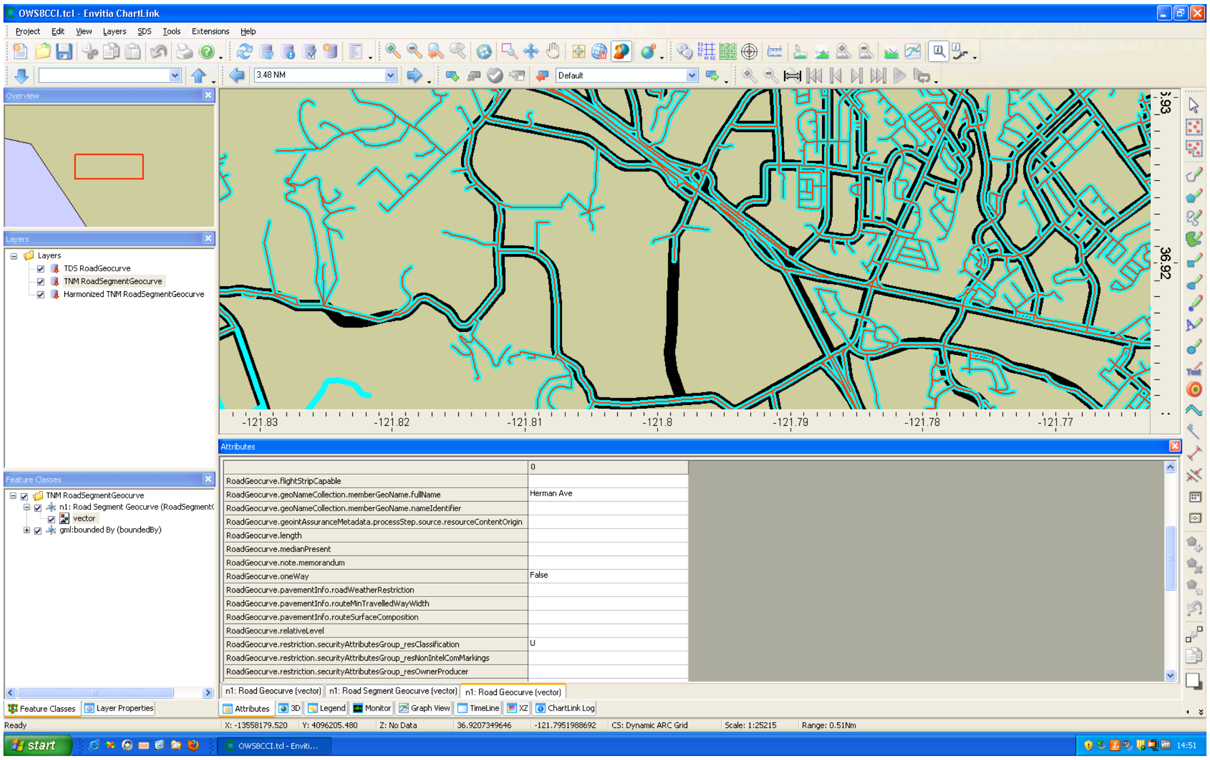

Figure 13. Visualization of TDS and Harmonized TNM Road networks using Envitia ChartLink

A presentation of attributes for a single TDS feature and the corresponding harmonized TNM feature, is presented in Figure 14. It can be seen from the screenshot that attributive information such as the road name and the type of road is available in the harmonized TNM data but not in the TDS data. Therefore in a situation where attributive information is not available in the TDS, the harmonized TNM data can provide additional attributive information about features. The reverse is also true for TNM and harmonized TDS data.

TDS:RoadGeocurve

	0
GML ID	TransportationCurves2462
area	-999999.00000000
centerlineSpacing	-999999.00000000
conditionOfFacility	fullyFunctional
containedInTunnel	false
controllingAuthority	noInformation
deckLevel	-999999
divided	false
featureConfiguration	nonDivided
flightStripCapable	false
geoNameCollection.memberGeoName.fullName	No Information
geoNameCollection.memberGeoName.nameIdentifier	-999999
geointAssuranceMetadata.processStep.source.resourceContentOrigin	noInformation
length	-999999.00000000
medianPresent	false
note.memorandum	No Information
oneWay	false
pavementInfo.roadWeatherRestriction	allWeather
pavementInfo.routeMinTravelledWayWidth	-999999.00000000
pavementInfo.routeSurfaceComposition	asphalt
relativeLevel	level
restriction.securityAttributesGroup_resClassification	U
restriction.securityAttributesGroup_resNonIntelComMarkings	No Information
restriction.securityAttributesGroup_resOwnerProducer	No Information

Harmonized TNM:RoadSegmentGeocurve

	0
GML ID	
RoadGeocurve.area	
RoadGeocurve.centerlineSpacing	
RoadGeocurve.conditionOfFacility	
RoadGeocurve.containedInTunnel	
RoadGeocurve.controllingAuthority	
RoadGeocurve.deckLevel	
RoadGeocurve.divided	
RoadGeocurve.featureConfiguration	
RoadGeocurve.flightStripCapable	
RoadGeocurve.geoNameCollection.memberGeoName.fullName	Dolan Rd
RoadGeocurve.geoNameCollection.memberGeoName.nameIdentifier	
RoadGeocurve.geointAssuranceMetadata.processStep.source.resourceContentOrigin	
RoadGeocurve.length	
RoadGeocurve.medianPresent	
RoadGeocurve.note.memorandum	A41: Local neighborhood rural road city street unseparated.
RoadGeocurve.oneWay	False
RoadGeocurve.pavementInfo.roadWeatherRestriction	
RoadGeocurve.pavementInfo.routeMinTravelledWayWidth	
RoadGeocurve.pavementInfo.routeSurfaceComposition	
RoadGeocurve.relativeLevel	
RoadGeocurve.restriction.securityAttributesGroup_resClassification	U
RoadGeocurve.restriction.securityAttributesGroup_resNonIntelComMarkings	
RoadGeocurve.restriction.securityAttributesGroup_resOwnerProducer	

Figure 14. Attributes for sample features of the TDS and harmonized TNM datasets

The attributes generated by the harmonization of the TNM:RoadSegmentGeocurve data are shown on the right hand side of Figure 14. It can be seen from the screenshot that the TNM road names were copied directly to the 'geoNameCollection.memberGeoName.fullName' attribute. Enumerations of the type of road were converted to TDS enumerants and stored in the 'note.memorandum' attribute. Similarly, TNM data security attribute values were converted to their TDS equivalents and stored in the restriction.securityAttributesGroup_resClassification attribute.

A.3 Recommendations on CMM-based Harmonization

One of the key capabilities of the CMM format is the ability to use OGC filters as constraints on feature attributes. The RDF-encoded mappings developed for OWS-8 did not consist of constructs similar to OGC filters, therefore this capability of the CMM was not tested. It is therefore recommended that future OWS testbeds explore the use of OGC filters in RDF-encoded mappings.

