Open Geospatial Consortium Inc.

Date: 2009-07-27

Reference number of this OGC® project document: **OGC 09-010**

Version: **0.3.0**

Category: OGC® Discussion Paper

Editor: Kristin Stock

OGC® Catalogue Services – OWL Application Profile of CSW

Copyright notice

Copyright © 2009 Open Geospatial Consortium, Inc.
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type: OGC® Discussion Paper
Document subtype: Application Profile
Document stage: Approved for public release
Document language: English

Contents

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

i. Preface

This document describes an Application Profile for the Web Ontology Language (OWL) [W3C OWL] for CSW. It is intended to define a specification for how ontologies built using RDF and OWL may be included within an OGC CSW catalogue to semantically-enable the catalogue.

The OGC Catalogue Services Specification [OGC 07-006r1] establishes a general framework for implementing catalogue services that can be applied to meet the needs of stakeholders in a wide variety of domains. This Application Profile is based on the HTTP protocol binding described in Clause 10 of the Catalogue Services Specification (referred to as CSW).

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

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

ii. Submitting organizations

The following organizations submitted this Implementation Specification to the Open Geospatial Consortium Inc.:

a) Allworlds Geothinking.

b) University of Nottingham.

c) EDINA, University of Edinburgh.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

iii. Submission contact points

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

CONTACT	COMPANY
Kristin Stock (Editor)	Allworlds Geothinking and University of Nottingham.
Mark Small (Contributor)	EDINA, University of Edinburgh
Yang Ou (Contributor)	University of Edinburgh
Femke Reitsma (Contributor)	University of Edinburgh

iv. Revision history

Date	Release	Author	Paragraph modified	Description
05/03/2009	1.0	Kristin Stock	All	First draft.
07/10/2009	0.3.0	Carl Reed	Various	Prepare for publication as DP

v. Changes to the OGC® Abstract Specification

The OGC® Abstract Specification does not require changes to accommodate this OGC® Application Profile.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

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 specification set forth in this document, and to provide supporting documentation.

This document assumes familiarity with W3C Resource Description Framework [W3C RDF], W3C Resource Description Framework Schema Specification [W3C RDFS] and W3C Web Ontology Language [W3C OWL].

This document describes the Application Profile in the following sections (conforming to the requirements of the Catalogue Services Specification [07-006r1] guidelines for structure and format):

- A description of the system context.
- The information models that must be used to interact with the contents of the catalogue and specifically describing the queryable and returnable properties from the CSW specification and their format.
- A specification of each of the external interfaces that are to be used under this application profile, and their request and response formats, including query language.
- Annex A, containing the abstract test suite (normative) for the Application Profile.
- Annex B, containing use cases (informative).

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Introduction

Catalogues based on the CSW standard [OGC 07-006r1] are used to store information to manage geospatial resources in an interoperable environment. This information may include metadata about resources, including data sets and web services, as well as other related information to manage the resources. The existing CSW application profiles based on ebRIM [OASIS ebRIM] and ISO 19115/119 [ISO 19115/119] do not address the systematic inclusion of semantic information about the resources in catalogues beyond the inclusion of basic metadata, so up until now, ad hoc approaches have been required in order to include such formalised semantics for geospatial resources.

OWL is a language for expressing ontologies that store semantic information about concepts using a description logic language that provides opportunities for various kinds of reasoning. Furthermore, particular ontologies have been developed to allow web services to be semantically described in a way that can support automated orchestration in an interoperable environment (for example, OWL-S [W3C OWL-S]).

The inclusion of OWL in a geospatial catalogue has a number of benefits, including the representation of richer semantic information to describe resources, to assist discovery and to provide opportunities for web services orchestration. The addition of the catalogue specification to the usual description logic ontology standards provides for interoperability among semantic resource descriptions within a spatial data infrastructure.

This document specifies an Application Profile for CSW based on OWL. It provides a profile for how RDF [W3C RDF] (on which OWL is based) and OWL [W3C OWL] can be included in a CSW [OGC 07-006r1] catalogue, and can be used to support semantic interoperability of geospatial catalogues.

This application profile was developed under the auspices of the COMPASS[1] Project, funded by the UK Joint Information Systems Committee[2]. However, the specification is generic and provides a specification that can be used to include any OWL ontology in a CSW catalogue.

[1] http://compass.edina.ac.uk/

[2] http://www.jisc.ac.uk/

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

OGC® Catalogue Services — OWL application profile of CSW

1 Scope

This application profile specifies how OWL ontologies can be included in a catalogue that is compliant with the OGC Catalogue Services Specification HTTP protocol binding (referred to as CSW) [OGC 07-006r1]. It includes mappings from RDF [W3C RDF], RDF Schema [W3C RDFS] and OWL [W3C OWL], since OWL is an extension to RDF and relies on many of its components.

This specification defines how RDF, RDF Schema and OWL documents may be represented in CSW compliant catalogues using the defined interfaces and their request and response formats.

The specification may be applied to any OWL ontology, RDF document or RDF Schema document.

This specification is applicable to any type of OWL [W3C OWL] content, but is best suited to the storage of information about resources that may be exchanged through a catalogue. Any type of resources may be included and described, and generic concepts that are described using OWL and provide semantic information about the resources in the catalogue may also be included. It is also applicable to OWL-S files [W3C OWL-S].

2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing [OGC 19105].

In addition to satisfying the requirements stipulated in all normative clauses and Annex A, a catalogue implementation must also satisfy all relevant requirements in the following base specifications:

- OGC Catalogue Specification, Clause 10 [OGC 07-006r1]
- OGC Web Services Common Specification 1.0 [OGC 05-008]
- OGC Filter Encoding Implementation Specification 1.1 [OGC 04-095]
- W3C Resource Description Framework [W3C RDF]
- W3C Resource Description Framework Schema Specification [W3C RDFS]

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

- Web Ontology Language (OWL) Specification [W3C OWL]
- W3C SPARQL Specifications [W3C SPARQL, W3C SPARQL WSDL, W3C SPARQL XML]

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this specification are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

DCMI DC, Dublin Core Metadata Element Set, Version 1.1, http://dublincore.org/documents/dces/

DCMI DCT, Dublin Core Metadata Terms, http://dublincore.org/documents/dcmi-terms/

ISO 19105, *Geographic information — Conformance and Testing*, 2005.

OGC 04-095, *Filter Encoding Implementation Specification*, version 1.1.0 (3 May 2005), http://portal.opengeospatial.org/files/?artifact_id=8340

OGC 06-103r3, *OpenGIS Implementation Specification for Geographic Information – Simple Feature Access – Part 1: Common Architecture.*

OGC 07-006r1, *OGC™ Catalogue Services Specification*, version 2.0.2. (February 2007, includes Corrigendum 2).

OGC 06-121r3, *OGC Web Services Common Specification*, version 1.0 (May 2005).

OMG UML, *Object Management Group Unified Modeling Language Specification* Version 1.5, March 2003. http://www.omg.org/docs/formal/03-03-01.pdf

W3C RDF, *RDF Primer, RDF Concepts and Abstract Syntax, RDF Syntax Specification, RDF Semantics* (10 February 2004), http://www.w3.org/RDF/.

W3C RDFS, *RDF Vocabulary Description Language 1.0: RDF Schema* (10 February 2004), http://www.w3.org/TR/rdf-schema/.

W3C OWL, *Web Ontology Language* (10 February 2004), http://www.w3.org/2004/OWL/.

W3C OWL-S, Semantic Markup for Web Services (22 November 2004), http://www.w3.org/Submission/OWL-S/

W3C SPARQL, *SPARQL Query Language for RDF* (15 January 2008), http://www.w3.org/TR/rdf-sparql-query/

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

W3C SPARQL WSDL, *SPARQL Protocol for RDF* (15 January 2008), http://www.w3.org/TR/rdf-sparql-protocol/

W3C SPARQL XML, *SPARQL Query Results XML Format* (15 January 2008), http://www.w3.org/TR/2008/REC-rdf-sparql-XMLres-20080115/

W3C WSDL, Web Services Description Language (WSDL) Version 2.0 (26 June 2007), http://www.w3.org/TR/wsdl20/

W3C XML, *Extensible Markup Language (XML)* 1.0 (Fifth Edition) (26 November 2008), http://www.w3.org/TR/REC-xml/

W3C XMLS, *XML Schema*, http://www.w3.org/XML/Schema

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
ontology
A formal specification of a shared conceptualisation. In this case, the word ontology refers to a formal specification using OWL [W3C OWL]. An ontology formally describes the semantics of a set of concepts, often covering a particular domain.

4.2
registry
A synonym for catalogue.

4.3
reasoning
Inference over an ontology, usually to derive new information or to test the validity or consistency of the ontology. An explanation of reasoning and an explanation of a range of types of reasoning are contained in [2].

5 Conventions

5.1 Abbreviated terms

The following abbreviated terms apply in this document:

OGC	Open Geospatial Consortium
OWL	Web Ontology Language [W3C RDF]
RDF	Resource Description Framework [W3C RDF]

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

RDFS	Resource Description Framework Schema [W3C RDFS]
UML	Unified Modeling Language [OMG UML]
XML	eXtended Markup Language [W3C XML]
SPARQL	SPARQL Query Language for RDF [W3C SPARQL, W3C SPARQL RDF, W3C SPARQL XML]
CSW	Clause 10 of the OGC Catalogue Services Specification (HTTP Binding) [OGC 07-006r1]

5.2 UML Notation

The diagrams that appear in this standard are presented using the Unified Modeling Language (UML) [OMG UML] static structure diagram. The UML notations used in this standard are described in the diagram below.

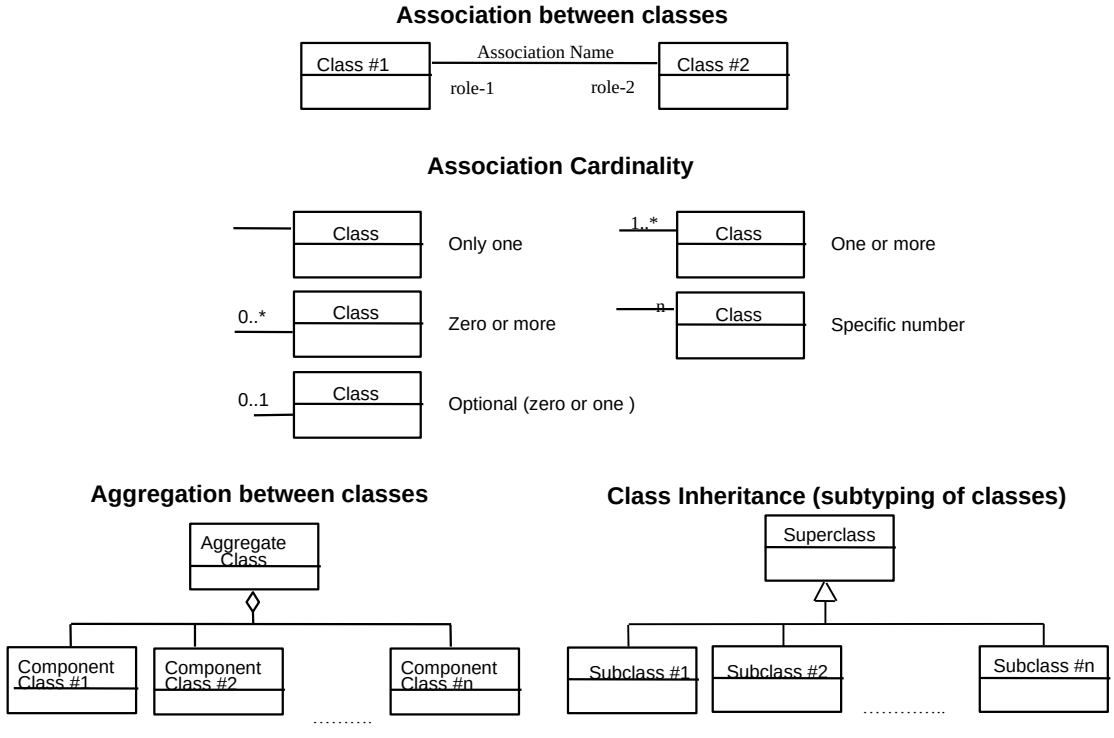


Figure 1 — UML notation

In this standard, the following standard data types are used:

a) String – A sequence of characters

b) Integer – An integer number

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

5.3 Namespaces

The following namespaces are used in this document:

Abbreviation	Namespace
dc	http://purl.org/dc/elements/1.1/ [DCMI DC]
dct	http://purl.org/dc/terms/ [DCMI DCT]
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns [W3C RDF[
rdfs	http://www.w3.org/2000/01/rdf-schema [W3C RDFS]
xsd	http://www.w3.org/2001/XMLSchema [W3C XSD]
owl	http://www.w3.org/2002/07/owl [W3C OWL]
st	http://www.w3.org/2005/09/sparql-protocol-types [W3C SPARQL WSDL]
sr	http://www.w3.org/2005/sparql-results [W3C SPARQL XML]
ows	http://www.opengis.net/ows [OGC 06-121r3]
ogc	http://www.opengis.net/ogc [OGC 06-103r3]
csw	http://www.opengis.net/cat/csw [OGC 07-006r1]
owlcsw	http://www.opengis.net/cat/owlcsw (defined in this specification)

Table 1: Namespaces

6 System Context

The goal of this Application Profile is to provide a specification for a semantic catalogue based on OGC standards. Such a catalogue provides semantic descriptions for the resources in the catalogue, and allows reasoning to be performed over the content of the registry.

The existing Application Profiles for CSW (for example, the ebRIM Profile) describe information models for access to registry content. However, if these Application Profiles are to be used with ontologies, they require the ontology content to be loaded into a structure that is consistent with the information model of the chosen Application Profile. Software that implements these existing Application Profiles normally uses a data storage structure that closely reflects the information model of the Application Profile. This requirement to use a different storage structure means firstly that some of the ontology content (detailed semantic specifications) may not be retained in full, and secondly that reasoning tools cannot be directly used on the content of the catalogue. In this case, the catalogue content must be either duplicated (in a format suited to the information model of the Application Profile and in a format that can be used by reasoning tools), or stored in either form by default and exported into the other form as required. [3] provides more details about the issues associated with including ontologies in ebRIM registries.

This Application Profile is designed with a view to supporting a registry architecture that stores the registry contents in the native OWL XML text files, a format that is generated by most ontology software and can be used by reasoning tools. In this way, the registry itself is an ontology (or series of ontologies), and the ontologies become the registry. The

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

content is stored as OWL, can be reasoned over, and the interface defined in this Application Profile operates directly on the OWL files. The content need not be duplicated or exported.

6.1 Application Domain

This Application Profile is not specific to a particular domain, and may be applied across all domains.

This Application Profile for a semantic registry is useful for any purpose involving enhanced use of geographic information, including discovery, manipulation, web services execution or orchestration. The Application Profile was initially developed for the purpose of creating a knowledge infrastructure that could describe resources in the form of publications, web services and data sets, and allowing intelligent discovery and reasoning over such resources. However, it could be used for a much wider range of purposes.

There is also scope to include web service ontologies in a registry that complies with this Application Profile, and to use these ontologies together with the inference tools that are made available through this Profile to perform automated discovery and orchestration of web services.

Figure 1 shows an example architecture for a system using this profile that can perform reasoning (and provide the results through this profile) and thereby make use of the semantic content of the catalogue. This particular architecture also combines results from digital library standards (Z39.50 and OAI-PMH).

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

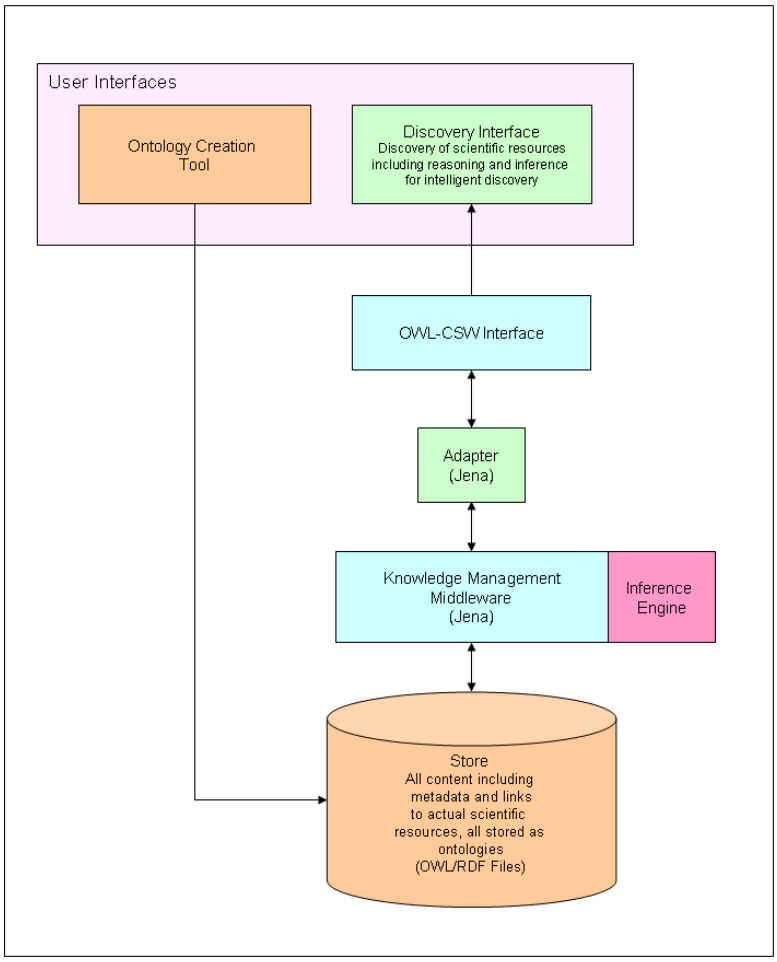


Figure 2 – Example Architecture for a Semantic Registry

6.2 Essential Use Cases

Annex B contains some example use cases for the kind of system that might use this Application Profile to store registry content as ontologies. However, these are not intended to encompass all of the possible uses of such a registry.

7 Information Models

7.1 Capability Classes

The capabilities provided by this Application Profile are as for CSW (OGC 07-006r1), and the same set of interfaces is supported. If anything is not explicitly described in this Application Profile, CSW should be assumed to apply.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

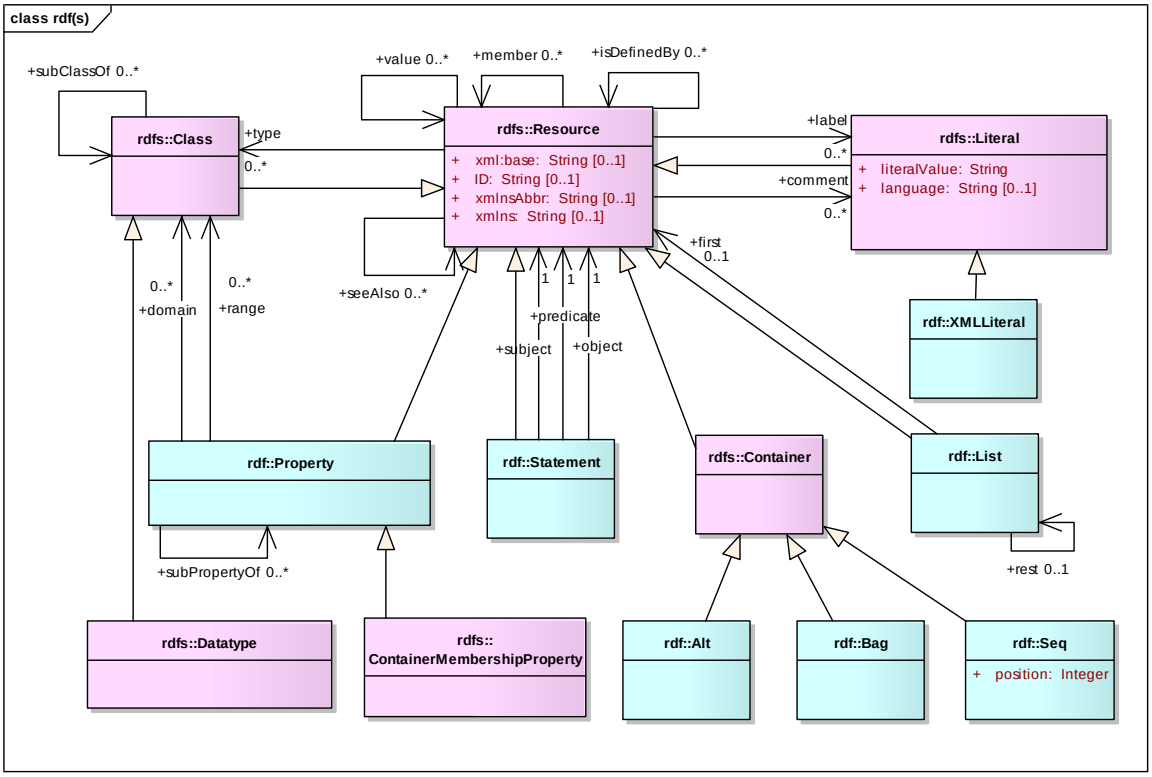
7.2 Catalogue Information Model

The information model for this Application Profile consists of two parts, corresponding to the standards that OWL incorporates. The OWL specification itself extends and depends on two other specifications: the Resource Description Framework (RDF) [W3C RDF] and RDF Schema [W3C RDFS]. Accordingly, the information model for this application profile has two components, one modelling RDF and RDF Schema (which are closely inter-related), and another modelling the OWL [W3C OWL] information model components that extend and build upon RDF and RDF Schema

7.2.1 Information Model RDF and RDF Schema Elements

The RDF and RDF Schema components of the application profile information model are shown in Figure 3, accompanied by descriptions of the mapping from RDF and RDF(S) to the information model in Table 2 and descriptions of the information model elements in Table 3.

The information model of RDF and RDF Schema elements is based on the Ontology Definition Metamodel (ODM) published by the Object Management Group [1] and adopts that model largely as it is, but with the exception that new objects included in the ODM that are not in RDF or RDFS are not included, as the information model for this application profile aims to accurately model RDF and RDF Schema without additional elements to allow easy mapping from actual ontology files into the information model for this application profile.



Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Figure 3 - Information Model: RDF and RDF Schema Elements

Table 2 - Mapping of RDF and RDF Schema to the Information Model

RDF and RDF Schema Element	Description	Information Model
rdfs:Resource	The class of all things. All things in RDF are resources.	rdfs:Resource class.
rdfs:Class	A resource with a rdfs:Resource.hasType association value of 'rdfs:Class'. A class represents a category of resources with similar characteristics.	rdfs:Class class.
rdfs:Literal	A literal value, used as the object of a property for example.	rdfs:Literal class
rdfs:Datatype	The class of datatypes, used to specify datatypes for a literal value.	rdfs:Datatype class.
rdf:XMLLiteral	A particular subset of string literals that meet a number of constraints and contain valid XML.	rdf:XMLLiteral class.
rdf:Property	A relation between subject and object resources and the classes to which they belong.	rdf:Property class.
rdfs:range	The classes of resource to which the property applies (the classes that are objects of the property).	rdf:Property.range association.
rdfs:domain	The classes of resource to which the property belongs (the classes that are subjects of the property).	rdf:Property.domain association.
rdf:type	A grouping of resources in a particular category (members of a class).	rdfs:Resource.type association.
rdfs:subClassOf	A relationship between a class and a more general class.	rdfs:Class.subClassOf association.
rdfs:subPropertyOf	A relationship between two properties, one of which is a specialisation of the other.	rdf:Property.subPropertyOf association
rdfs:label	A human-readable version of a resource's name.	rdfs:Resource.label association.
rdfs:comment	A human-readable description of a resource.	rdfs:Resource.comment association.
rdfs:Container	A grouping of resources that are members of a set (either unordered, ordered or a set of alternatives). A superclass of the possible types of container (Bag, Seq and Alt).	rdfs:Container class
rdf:Bag	An unordered grouping of resources that are members of a set.	rdf:Bag class
rdf:Seq	An ordered grouping of resources	rdf:Seq class

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

RDF and RDF Schema Element	Description	Information Model
	that are members of a set.	
rdf:Alt	A grouping of resources that are alternative members of a set.	rdf:Alt class
rdfs:ContainerMembershipProperty	A particular type of property that indicates that a resource is a member of a container.	rdfs:ContainerMembershipProperty class
rdfs:member	A relationship between two resources, one of which is a member of the other, container resource.	rdfs:Resource.member association.
rdf:List	A grouping of resources that are members of a list.	rdf:List class
rdf:first	A relationship between a list and the resource that is the first element in the list.	rdf:List.first association.
rdf:rest	A relationship between a list and the rest of the list.	rdf:List.rest association.
rdf:nil	A flag indicating the end of a list.	Not explicitly represented in the model. This would appear as an instance of the rdf:List class.
rdf:Statement	A statement linking resources that are the subject, object and predicate.	rdf:Statement class.
rdf:subject	A relationship indicating the subject of a statement.	rdf:subject association.
rdf:predicate	A relationship indicating the predicate of a statement.	rdf:predicate association.
rdf:object	A relationship indicating the object of a statement.	rdf:object association.
rdfs:seeAlso	Another resource that might provide additional information about a resource.	rdfs:Resource.seeAlso association.
rdfs:isDefinedBy	Another resource that defines the resource.	rdfs:Resource.isDefinedBy association.
rdf:value	A piece of vocabulary that may be used to describe structured values (for example, the units used by a literal value of a property. It is generic and may be applied in a number of circumstances.	rdfs:Resource.value association.

Table 3 - Description of Information Model Elements from RDF

Information Model Object	Description
rdfs:Resource	A generic superclass containing attributes that may be applied to any other class in the information model. The rdfs:Resource class has attributes:

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Information Model Object	Description
	- xml:base: the namespace of the resource. - ID: the local identifier of the resource (combined with the value in xml:base, this gives a unique identifier). - xmlnsAbbr: a namespace abbreviation that may be declared for use within the resource and its children, the actual namespace location being stored in xmlns. - xmlns: the actual namespace. The Resource class has associations: - type: an instance of rdf:Property indicating the type of resource, either indicating membership of a class (if the association is with rdfs:Class, or the data type (if the association is inherited and is with rdfs:Datatype). - label: an instance of rdf:Property providing a human-readable version of the resource's name. - comment: an instance of rdf:Property providing a human-readable description of a resource. - seeAlso: an instance of rdf:Property indicating another resource that provides additional information about the resource. - isDefinedBy: an instance of rdf:Property indicating another resource that provides a definition for the resource. - value: an instance of rdf:Property indicating a resource that is part of a structured value (usually attached to a literal). - member: an instance of rdf:Property indicating a resource that is a member of another resource that is a container. - first: an instance of rdf:Property indicating that a resource is the first item in a list (collection). - subject: an instance of rdf:Property indicating that the resource is the subject of an RDF triple. - object: an instance of rdf:Property indicating that the resource is the object of an RDF triple. - predicate: an instance of rdf:Property indicating that the resource is the predicate of an RDF triple.
rdfs:Class	A category of resources with similar characteristics. The Class class is used in conjunction with rdfs:Resource.hasType. The type attribute assigns specific resources to the class that is represented in the Class class. RDF Schema allows a Class to be created and then resource descriptions assigned to it. Class specialises rdfs:Resource with a type value of 'rdfs:Class'. The rdfs:Class class has associations: - subClassOf: an instance of rdf:Property indicating the parent (more general) class of the current class.
rdfs:Datatype	The datatype of the literal value, referencing an existing data type specified in another place (for example, some XML Schema types are valid). RDF Schema does not allow new data types to be defined. This class inherits attributes from rdfs:Resource, all of which are optional in the rdfs:Resource class but which are compulsory in the

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Information Model Object	Description
	Datatype class.
rdfs:Literal	A literal value, usually used as an object of a property. Literal values may be plain or typed. The Property class has attributes: - literalValue: an instance of rdf:Property indicating the actual literal value, possibly text strings or integers. - language: an instance of rdf:Property indicating the language of the literalValue attribute value. This class inherits attributes and associations from rdfs:Resource, including the type association with rdfs:Class, which is inherited by rdfs:Datatype. Typed literals must have a value for the inherited type association with the rdfs:Datatype class.
rdf:XMLLiteral	An XML literal, containing valid XML encoded text.
rdf:Property	A relationship between a resource and classes that are related to the resource as subject or object. This is a generic structure that may be used to describe a number of relationships including: - classes of objects of a predicate; - classes of subjects of a predicate; - classes of a resource; - types of a literal. The Property class has associations: - range: an instance of rdf:Property indicating the classes (or data types) that are the object of a property, allowing none (indicating nothing, so any classes are objects), one or multiple classes. - domain: an instance of rdf:Property indicating the classes that are subjects of the property, allowing none (indicating nothing, so any classes are subjects), one or multiple classes. - subPropertyOf: an instance of rdf:Property indicating a parent, or more general property of the current property, meaning that pairs of instances that are the subject and object of a particular property are also the subject and object of the parent property, and that all domain and range associations that apply to a property also apply to all its subproperties. The information contained in the range and domain associations is only descriptive (as per the RDF Schema specification), and it's actual use and semantic meaning is decided by the application: - it may be interpreted as a constraint, dictating which classes may be subjects or objects of a property and thus used for validation or control of input or - it may be interpreted as an inference, identifying that an instance with the property is a member of the class (or if there are multiple classes, then a member of all of the classes).
rdfs:Container	A set of resources that are members of a collection. Collections of various types are possible, depending on the subclass to which the resources belong.
rdf:Bag	An unordered collection of resources.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Information Model Object	Description
rdf:Alt	An unordered collection of resources in which it is intended that one of the members be selected, the default being the first member.
rdf:Seq	An ordered collection of resources, in which the numerical ordering of the members is intended to be significant. The Seq class has attributes: position: the position of the element in the sequence.
rdfs:ContainerMembershipProperty	A class of resources containing properties that indicate that a resource is a member of a container resource. Instances of this class of properties are subproperties of the rdfs:Resource.member association.
rdf:List	A class representing a collection or list of resources. The List class has associations: - first: an instance of rdf:Property indicating the resource that is the first element in the list. - rest: an instance of rdf:Property indicating the rest of the list. The rdf:Nil element that signals the end of a list is represented as an instance of the rdf:List class. Lists differ from Containers in that their membership is exhaustive. A list shows all members of the list, while a Container may always have other members in the same or different documents.
rdf:Statement	A class representing a statement linking subject, object and predicate. The Statement class has associations: - subject: an instance of rdf:Property indicating the resource that is the subject of the statement. - predicate: an instance of rdf:Property indicating the resource that is the predicate of the statement. - object: an instance of rdf:Property indicating the resource that is the object of the statement. Statements form the core of RDF, and allow resources to be related to each other. Resources can be the subject (a URI) or object (a URI or literal) of a statement and also a property that represents the predicate linking the subject and object.

The information model described in Tables 2 and 3 provides a generic and flexible information model containing the elements in RDF and RDF Schema. RDF also includes an XML syntax for RDF, showing specifically how these elements should be represented in an XML format. Table 4 explains how each of these RDF/XML elements shall be represented in the information model for the OWL application profile for CSW.

Generally speaking, RDF/XML is mapped straight into the UML information model, but some simplifications are made. An RDF document normally uses various abbreviations so that text need not be repeated throughout an RDF document. However, this Application Profile requires that objects be self-contained and externally referable, since a query to the catalogue through the application profile may return only a fragment of the entire catalogue content.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 4 - Representation of RDF/XML Elements in the Information Model

RDF/XML Element	Information Model Representation
RDF	The RDF element is simply a container for a particular RDF document, indicating that RDF is contained therein. This is not required in the information model because the entire contents of any catalogue built using this application profile will contain RDF (and its OWL extensions).
Description element	The contents of a description element are instances of rdf:Statement. The about attribute is rdf:Statement.object, the first child is rdf:Statement.predicate and the child within the predicate is rdf:Statement.object. The type attribute is rdf:Resource.type. As per the information model, the classes at the end of all of these associations are resources. - In the case of the subject (the about attribute), the resource must be an instance of rdfs:Resource, not one of its children. - In the case of the predicate, the resource must be an instance of rdf:Property. - In the case of the object, the resource may be an instance of any subclass of rdfs:Resource (including rdf:Statement) except rdf:Property. RDF/XML allows the Description element to have multiple nested children to represent different properties of the same resource (different predicate and object pairs for a single subject). This is simply represented using multiple rdf:Statement instances with the same value for rdf:Statement.subject. RDF/XML allows Description elements with no about attribute, effectively creating an empty subject that is used to create a grouping of predicates and objects that refer to a single, unnamed subject. These are referred to in RDF/XML as blank nodes. In RDF/XML, blank nodes may optionally be given a nodeID attribute (an internal identifier) so that they can be referenced from within the document. For the purposes of the information model, all blank nodes are to be assigned an arbitrary identifier using the namespace of the source document in the rdfs:Resource class (that is, unnamed resources are not permitted). This identifier shall be stored in the rdfs:Resource.ID attribute with the namespace of the local document from which the Description originates. Some abbreviations are allowed within the Description element that must be expanded for their full form for representation in the information model. These are: - Property attributes, including the predicate as a property of the Description element; - Subjects of properties as attributes of the property.
XML DOCTYPE	The XML element is simply a container for a particular XML document, indicating that XML is contained therein. This is not required in the information model because the entire contents of any catalogue built using this application profile will contain XML (and its RDF and OWL extensions).

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

RDF/XML Element	Information Model Representation
xml:lang	rdfs:Literal.language
parseType = “Literal”	Items with an parseType literal are instances of the rdf:XMLLiteral class.
datatype	Literal objects of a statement may have a datatype attached. This is implemented in the information model using rdfs:Resource.type, inherited by all resources and in this case pointing to the rdfs:Datatype subclass of rdfs:Class.
nodeID	Blank nodes (without a resource attribute) may have a nodeID assigned to provide a way of internally referencing them. This is implemented in the information model using rdfs:Resource.ID with the namespace of the source document. No distinction is made between blank nodes and other nodes. In the case of blank nodes, nodeID is assigned to ID, while in the case of nodes (Description elements) that have resource attributes, resource is split and assigned to the ID and namespace.
resource	The resource attribute is simply an abbreviation that allows the Description element to be omitted for blank nodes. In the information model the full form as mapped to the information model shall be used.
rdf:li and rdf:_n	These values are used to list the sequence of elements in a Container. These elements are only meaningful if the Container is of type Seq. This information is stored in the rdf:Seq:position attribute. Values for this attribute may be set to the value of n if rdf:_n is used, otherwise allocated in the order that each element appears in the Seq in the source document.
parseType = “Collection”	A grouping of Descriptions that form an ordered list with a finite and completely described membership. A collection usually refers to a group of objects of a predicate. In the information model such a group is represented using rdf:List and its associated associations to indicate the start, order and end of the list (as distinct from a Container of type Seq which is not exhaustive).

7.2.2 Information Model OWL Components

OWL (W3C OWL) is a semantic markup language for describing ontologies, and is an extension to RDF and RDF Schema. OWL provides three increasingly expressive sublanguages designed for particular purposes:

- OWL Lite is a very simple structure describing a classification hierarchy.
- OWL DL is a description logic language, allowing expressivity without losing computational completeness and decidability for reasoning.
- OWL Full has maximum expressiveness but does not guarantee computational completeness or decidability.

This application profile uses OWL-DL to ensure computational resolvability, but the changes required to use OWL-Full would be minimal, since OWL-Full simply applies additional constraints. The information model is presented in Figure 4.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

The mapping from OWL into the information model is simple and involves further extending the information model for RDF in a similar way. Generally, OWL elements map to an element with the same name in the information model (OWL classes map to classes, OWL properties map to instances of rdf:Property, shown as associations, so the mapping is not fully listed as it was with RDF and RDF Schema except in cases that are unusual or where explanation is required (Table 5), and Table 6 provides a description of the information model elements.

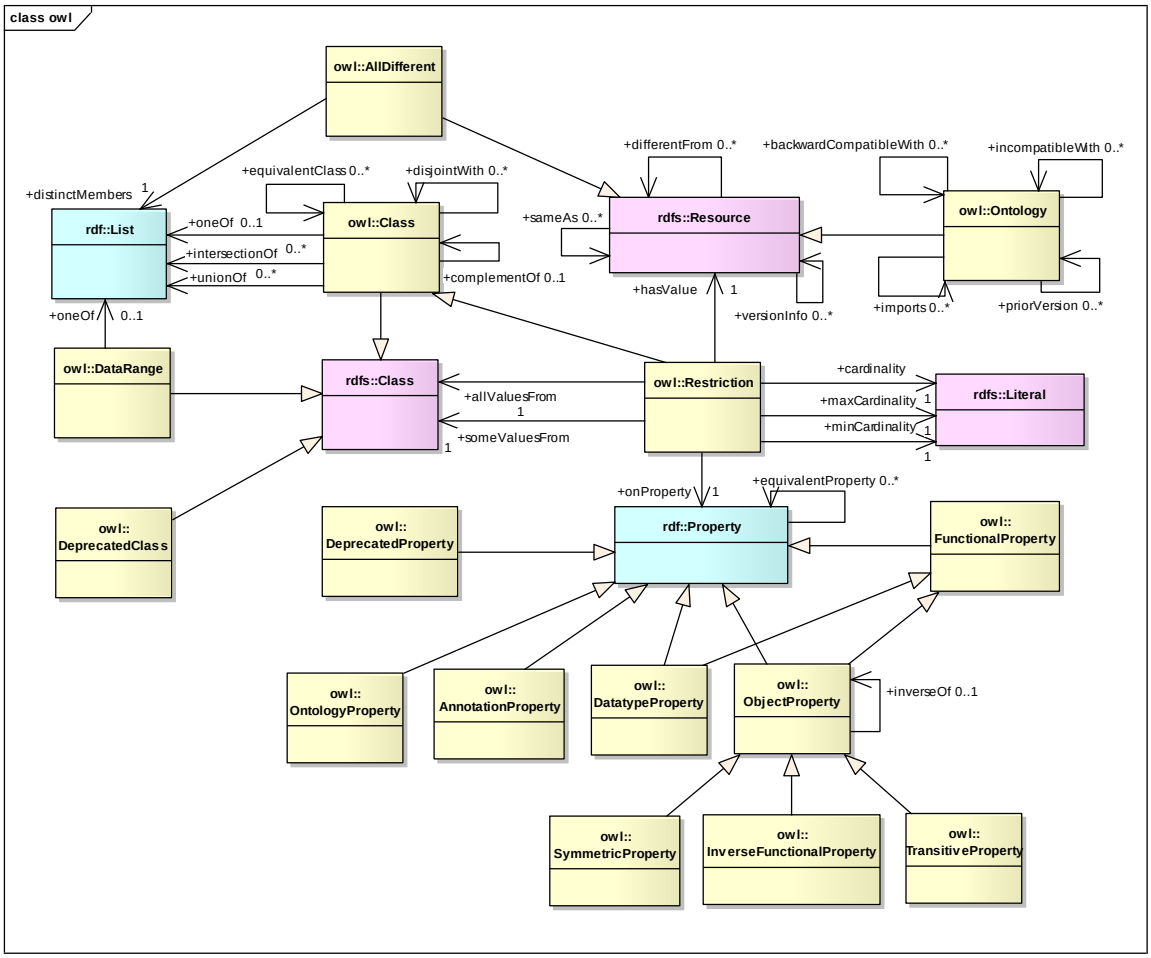


Figure 4 - Information Model: OWL Elements

The Ontology Definition Metamodel Specification [1] was not closely followed for this part of the information model, mainly because the ODM model includes a number of additional classes that are not part of the OWL specification, and are not required for this Application Profile. Also, the ODM provides classes for individuals or instances, and these are not represented in the information model because they would appear as instances of the information model objects. For example, an instance of a Literal would appear as instances of the rdfs:Literal class in the information model for the application profile.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 5 – Unusual Mappings from OWL Elements into the Information Model

OWL Schema Element	Comment
versionInfo	versionInfo is mapped to an association on rdfs:Resource, since it may be attached to any OWL resource.
Thing and Nothing	These classes are not required and are not included, since all elements are members of Thing and none are members of Nothing.
differentFrom and SameAs	These two properties are listed as being on Thing, which is equivalent in principal to rdfs:Resource, so the associations are attached to rdfs:Resource.

Table 6 - Description of Information Model Elements from OWL

Information Model Object	Description
owl:Ontology	A class representing a grouping of resources that form a coherent description within a domain, sometimes conceptual and sometimes defined for business purposes. This object is a container for elements within the ontology and for header information. This header information is represented using the associations of the class and those inherited from rdfs:Resource. The Ontology class has associations: - backwardCompatibleWith:.an association with a prior version of the ontology with which it is backward compatible, meaning that identifiers from the previous version have the same interpretation in the new ontology. - incompatibleWith: an association with a prior version of the ontology with which it is not backward compatible. - priorVersion: an association with a prior version of the ontology with no additional semantic meaning. - imports: an association with another OWL ontology containing definitions, whose meaning is considered to be part of the meaning of the importing ontology.
owl:Class	A class representing a category of resources. The various associations that are attached to this class usually define necessary or sufficient characteristics of a class. The Class class has associations: - equivalentClass: an association between two classes indicating that they have the same extension (membership). - disjointWith:.an association between two classes indicating that their extensions have no individuals in common. - complementOf: an association between two classes indicating that the members of one class are exactly those individuals that are not members of the other. - oneOf: an association between a class and a list indicating that the individuals that are instances of the class must be one of those belonging to the list. - intersectionWith: an association between a class and a list indicating that the extensions of the class and the list are the same.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	- unionOf: an anonymous class containing all of the members of a List (combining possible multiple descriptions of list membership). owl:Class is similar to rdfs:Class but a separate class is needed in order to define additional restrictions necessary for a Class to meet OWL DL requirements, which are stricter than those for rdfs. However, OWL Full does not have these requirements so the two classes could merge together in an OWL Full representation.
owl:DeprecatedClass	A class that is retained for backward compatibility but should not be included in a new ontology.
owl:AllDifferent	A class that links any type of resource to a list, specifying that all of the members of that list are different from each other. This is used as a way of avoiding having to make a large number of pairwise statements using the differentFrom property. The AllDifferent class has associations: - distinctMembers: an association between the set of different items and an individual, distinct item.
owl:DataRange	A class containing an enumerated set of literals. The DataRange class has associations: - oneOf: an association with a list containing the literal members of the data range.
owl:Restriction	A class containing restrictions on properties of various types, some of which are value constraints and some of which are cardinality constraints. Value constraints limit the possible range of the property, while cardinality constraints limit the number of values a property can take in the context of a particular class description. The different types of constraints are represented using associations. The Restriction class has associations: - onProperty: an association indicating the property on which the restriction applies. - hasValue: an association to the actual value that the property range must have, which may be an individual or a data value (literal). - allValuesFrom: an association to a class specifying that all property range values must be members of the class (or the data range). - someValuesFrom: an association to a class specifying that at least one of the property range values must be a member of the class (or the data range). - cardinality: an association to a literal value from the XML Schema data type nonNegativeInteger describing how many instances of the property members of the class must have. - maxCardinality: an association to a literal value from the XML Schema data type nonNegativeInteger describing the maximum number of instances of the property members of the class must have. - minCardinality: an association to a literal value from the XML Schema data type nonNegativeInteger describing the minimum number of instances of the property members of the class must

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	have.
owl:DeprecatedProperty	A property that is retained for backward compatibility but should not be included in a new ontology.
owl:OntologyProperty	A class of properties that apply to ontologies, including imports, backwardCompatibleWith, incompatibleWith and priorVersion. The domain and range of these properties must be a member of the owl:Ontology class.
owl:AnnotationProperty	A class of properties that are considered annotation, including. versionInfo, label, comment, seeAlso, isDefinedBy. Restrictions on the domain and range of these properties are applied by OWL.
owl:FunctionalProperty	A class of properties for which each member of the domain can be associated with no more than one member of the range by that property (that is, each subject can have only one object).
owl:DatatypeProperty	A set of properties that link individuals to data values. This class is not necessarily mutually exclusive with other property classes.
owl:ObjectProperty	A set of properties that link individuals to other individuals. This class is not necessarily mutually exclusive with other property classes. The ObjectProperty class has associations: - inverseOf: an association indicating that the property is an inverse of another property that reverses the domain and range but has a different meaning.
owl:SymmetricProperty	A class of properties for which if the property connects a domain and range, the same property can connect the range and domain if they are swapped over (the property is true in both directions).
owl:TransitiveProperty	A class of properties for which if the range of one instance of the property is also the domain of another instance of the property, the domain of the first property and the range of the second property are also related via the same property.
owl:InverseFunctionalProperty	A class of properties for which each member of the range can be associated with no more than one member of the domain by that property (that is, each object can have only one subject).
owl:versionInfo on rdfs:Resource	An association applying version information to any OWL resource. This is a string giving information, possibly combined with keywords.
owl:differentFrom on rdfs:Resource	An association indicating that two resources refer to different individuals.
owl:sameAs on rdfs:Resource	An association indicating that two resources refer to the same individual.
owl:equivalentProperty on rdf:Property	An association indicating that two properties have the same property extension (but possibly different intensional meaning).

7.3 Supported Data Bindings

The OGC Catalogue Services Specification [OGC 07-006r1] abstractly describes a set of core metadata properties that are to be made available for query and response by

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

catalogue services that are realised by binding protocols. These are based on Dublin Core [DCMI DC] properties and terms.

This specification is an OWL Application Profile for CSW and is thus compatible with Dublin Core, which uses RDF and RDFS. Therefore, all of the Dublin Core properties from CSW are included in this specification in their Dublin Core form. In addition, the owlcsw:BoundingBox, and rdfs:Resource elements are added, the former to cater for spatial searches (defined in the owlcsw namespace) and the latter to allow all elements within an ontology to be returned. The AnyText queryable item must also be supported by implementations under this Application Profile. This item does not map to any specific element but allows a free text search across the names and contents of all elements.

Table 7 summarises the queryable and returnable properties that must be supported by implementations of this Application Profile. The properties that are marked as queryable in Table 7 may be included in a query element as part of a request using some operations under this Application Profile (see Section 8.9). Section 8 provides more detail about how these attributes are to be used in queries and how they are to be returned in response to queries.

Ontologies within a registry that implements this Application Profile will have their own structure that is domain dependent, and this set of attributes is not intended to constraint that structure. However, all of these properties must be supported and either queryable or returnable as specified (depending on the attribute). Implementations of this Application Profile must therefore map their own ontology representation to these queryable and returnable attributes if they are not explicitly represented in the ontologies. The response must return the parent element to which the property applies (for example, if the 'title' property is queried, the returned element must be the resource that has the specified title).

Table 7: OWL Application Profile Queryable and Returnable Properties

Dublin Core Element (Cat Services)	OGC Queryable Item (Cat Services)	XML Element Name (CSW)	OWL Application Profile Element	Queryable?[3]
title	Title	dc:title	http://purl.org/dc/elements/1.1/title	Yes
creator		dc:creator	http://purl.org/dc/elements/1.1/creator	No
subject	Subject	dc:subject	http://purl.org/dc/elements/1.1/subject	Yes
description	Abstract	dct:abstract	http://purl.org/dc/terms/abstract	Yes
publisher		dc:publisher	http://purl.org/dc/elements/1.1/publisher	No
contributor		dc:contributor	http://purl.org/dc/elements/1.1/contributor	No
date	Modified	dct:modified	http://purl.org/dc/terms/modified	Yes
type	Type	dc:type	http://purl.org/dc/elements/1.1/type	Yes
format	Format	dc:format	http://purl.org/dc/elements/1.1/format	Yes

[3] This means that the property may be included in a csw:Query element as part of an operation request under this Application Profile.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

identifier	Identifier	dc:identifier	http://purl.org/dc/elements/1.1/identifier	Yes
source	Source	dc:source	http://purl.org/dc/elements/1.1/source	Yes
language		dc:language	http://purl.org/dc/elements/1.1/language	No
relation	Association	dc:relation	http://purl.org/dc/elements/1.1/relation	Yes
coverage	BoundingBox	owlcsw:BoundingBox	http://www.opengis.net/cat/owlcsw/has_bbox	Yes
rights		dc:rights	http://purl.org/dc/elements/1.1/rights	No
	AnyText		Does not map to any specific element, but requires all elements to be searched in the entire catalogue (not just the queryable and returnable properties).	Yes
		rdfs:Resource	Any resource in the ontology may be returned using the format described in Section 7.2.	No

The following describes the format that shall be used for bounding boxes in queries and responses under this specification (in the owlcsw namespace). This may be imported into other ontologies and the BoundingBox object referenced by classes and properties in those ontologies.

Code 1 – Definition of Bounding Box in the owlcsw Namespace:

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://compass.edina.ac.uk/ontologies/owlcsw.owl#"
 xml:base="http://compass.edina.ac.uk/ontologies/owlcsw.owl">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Point">
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty>
         <owl:DatatypeProperty rdf:ID="crs"/>
       </owl:onProperty>
       <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
       >1</owl:maxCardinality>
     </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
 </owl:Class>
 <owl:Class rdf:ID="BoundingBox">
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty>
         <owl:DatatypeProperty rdf:about="#crs"/>
       </owl:onProperty>
       <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
       >1</owl:maxCardinality>
     </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
 </owl:Class>
 <owl:ObjectProperty rdf:ID="lowerCorner">
   <rdfs:domain rdf:resource="#BoundingBox"/>
   <rdfs:range rdf:resource="#Point"/>
 </owl:ObjectProperty>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
 <owl:ObjectProperty rdf:ID="upperCorner">
   <rdfs:range rdf:resource="#Point"/>
   <rdfs:domain rdf:resource="#BoundingBox"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:about="#crs">
   <rdfs:domain rdf:resource="#BoundingBox"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="x">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
   <rdfs:domain rdf:resource="#Point"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="y">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
   <rdfs:domain rdf:resource="#Point"/>
 </owl:DatatypeProperty>
</rdf:RDF>
```

Supported element sets that contain subsets of the core queryable and returnable properties are specified in Section 8.9.2.

7.4 Service Information Model

The services offered by implementations of this Application Profile shall be the same as for the CSW, with the modifications described in Section 8.

7.5 Native Language Support

The RDFS [W3C RDFS] specification provides support for multiple languages and character encodings using the language property of the Literal class. Implementations shall use this property to specify the language of catalogue content.

8 External Interfaces

8.1 Imported Protocol Bindings

The general requirements for operation request and response encoding specified in CSW [OGC 07-006r1] shall be complied with by implementations of this Application Profile, including requirements for HTTP method bindings, KVP and XML encoding, SOAP messaging and exception reporting. This Application Profile does not specify any further constraints on these aspects of CSW.

This Application Profile also supports the same set of operations as the CSW specification, derived from the general catalogue interface model specified in the Catalogue Services Specification [OGC 07-006r1]. However, in some cases the parameters and payload of the operations are modified slightly to suit the requirements of this Application Profile. This Section specifies the requirements for each of the operations under this Application Profile.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

8.2 GetCapabilities

8.2.1 Introduction

The mandatory GetCapabilities operation allows CSW clients to retrieve service metadata from a server. The response to a GetCapabilities request shall be an XML document containing service metadata about the server. This subclause specifies the XML document that a CSW server that conforms to this Application Profile shall return to describe its capabilities.

8.2.2 Operation Request

General requirements for CSW GetCapabilities requests are described in the CSW specification [OGC 07-006r1] which references the OGC Web Services Common Specification [OGC 06-121r3], and these shall be conformed to by implementations of this Application Profile.

The value of the *service* parameter for services conforming to this Application Profile shall be "OWLCSW", and the version shall be "1.0.0".

The CSW specification [07-006r1] extends the set of sections offered by the basic Catalogue Services specification by adding a Filter_Capabilities section that refers to the OGC Filter Encoding Specification [04-095]. This specification adds two further sections as shown in Table 8, and these can be included in the list of Sections in a GetCapabilities request to implementations of this Application Profile.

Table 8: Additional Sections for OWLCSW Catalogue Services

Section Name	Meaning
Sparql_Capabilities	A Sparql_Capabilities section shall be included in the service metadata to describe which (if any) elements of SPARQL are supported. This specifically includes the query forms supported by the service (SELECT, CONSTRUCT, DESCRIBE and ASK).
Reasoning_Capabilities	A Reasoning_Capabilities section shall be included in the service metadata to describe which (if any) reasoning capabilities are supported. Services are not required to support any reasoning capabilities, but if they do, they must be described in the GetCapabilities document. A number of common types of reasoning may be supported as described in Table 9.

Table 9 describes in more detail the types of reasoning that may be supported by implementations of this Application Profile. The table also identifies the operation under which these reasoning methods are made available. More details about how the reasoning method may be invoked are included under the Section describing the relevant operation.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 9: Reasoning Types Available in the OWL Application Profile for CSW

Reasoning Method	Parameters[4]	Returns	Operation
Satisfiability	An OWL Class.	True or False, indicating whether the class specification is satisfiable.	DescribeRecord
Subsumption	A collection of two or more classes.	The subsumption hierarchy. If none of the classes are subsumed by others, then the empty set is returned.	GetRecords
Equivalence	A pair of OWL classes.	The set of equivalent classes. If the classes are not equivalent, then the empty set is returned (either both are returned or none).	GetRecords
Disjointness	A pair of OWL classes.	The set of disjoint classes. If the classes are not disjoint, then the empty set is returned (either both are returned or none).	GetRecords
Consistency	A set of individuals in an OWL ontology.	The set of consistent individuals. If the individuals are not consistent, then the empty set is returned (either all are returned or none).	GetRecords
InstanceChecking	A single individual and a set of individuals.	The single individual, if it is entailed by the set of individuals, otherwise the empty set.	GetRecords
Retrieval	An OWL description and a set of individuals.	The members of the set of individuals that fulfil the OWL description. If none of the individuals fulfil the description, then the empty set is returned.	GetRecords
Realization	An individual and a set of OWL Classes.	The most specific OWL classes that the individual is a member of. If the individual is not a member of any of the OWL classes, the empty set is returned.	GetRecords

8.2.3 Operation Response

The GetCapabilities response shall conform to the requirements of the CSW specification [07-006r1]. In addition, the Sparql_Capabilities and Reasoning_Capabilities sections shall be included if the catalogue service supports either SPARQL or reasoning capabilities. Code 2 contains the definition of these sections in the GetCapabilities Response.

Code 2 – Definition of Extensions to GetCapabilities Response

```
<xsd:complexType name="CapabilitiesType" id="CapabilitiesType">
 <xsd:annotation>
  <xsd:documentation>This type extends csw:CapabilitiesType defined in OGC 07-006r1 to
include information about SPARQL and reasoning.</xsd:documentation>
 </xsd:annotation>
```

[4] Where an OWL class included in the set of parameters, it includes the full OWL description for the class, as this is required to perform reasoning.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
 <xsd:complexContent>
  <xsd:extension base="csw:CapabilitiesType">
   <xsd:sequence>
    <xsd:element ref="Sparql Capabilities"/>
    <xsd:element ref="Reasoning_Capabilities"/>
   </xsd:sequence>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
<xsd:element name="Sparql Capabilities">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:enumeration value="SELECT"/>
   <xsd:enumeration value="CONSTRUCT"/>
   <xsd:enumeration value="DESCRIBE"/>
   <xsd:enumeration value="ASK"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
<xsd:element name="Reasoning Capabilities">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:enumeration value="Satisfiability"/>
   <xsd:enumeration value="Subsumption"/>
   <xsd:enumeration value="Equivalence"/>
   <xsd:enumeration value="Disjointness"/>
   <xsd:enumeration value="Consistency"/>
   <xsd:enumeration value="InstanceChecking"/>
   <xsd:enumeration value="Retrieval"/>
   <xsd:enumeration value="Realization"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

8.3 DescribeRecord

8.3.1 Introduction

The mandatory DescribeRecord operation allows a client to discover elements of the information model supported by the target catalogue service. This operation is particularly relevant for this Application Profile because although the information model is specified by OWL and RDF(S), there are many cases in which the user (machine or human) will need to discover the details of the actual ontology that is used for a particular catalogue implementation. Such ontologies are application specific, but as has been discussed in other Sections, the level of nesting used in an ontology may be important for the useful interpretation of the content.

8.3.2 Operation Request

The request parameters are the same as for the CSW specification [OGC 07-006r1], with some restrictions and additions as shown in Table 10.

Table 10: Parameters for KVP Encoding for DescribeRecord Operation Request

Parameter	Data type and value	Optionality and use

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

REQUEST[5]	Character String. Fixed value of *DescribeRecord*, case insensitive.	One (mandatory)
service	Character String. Fixed value of *OWLCSW*	One (mandatory)
version	Character String. Fixed value of *1.0.0*	One (mandatory)
NAMESPACE[6]	List of Character String, comma separated. The namespaces for any elements (TypeNames) that are to be searched. This may include the generic OWL and RDF namespaces if those TypeNames are of interest, or may contain the namespaces relating to specific ontologies (for example, the "is" ontology in Example 5). Format is xmlns([prefix=]namespace-url). If prefix is not specified, then it is the default namespace.	Zero or one (optional). Include declarations for each namespace used in a TypeName.
TypeName	List of Character String, comma separated. The element in the ontology that is of interest. This may be a generic element in OWL or RDF (although this structure is already known), but is more likely to be an instance of an OWL element (e.g. an OWL Class) or RDF Resource that is of interest. Namespace qualification is required.	Zero or one (optional). Include declarations for each namespace used in a TypeName. Default action is to describe all types known to the server.
outputFormat	Character String. Fixed value of *application/rdf+xml*	Not required.
schemaLanguage	Character String. Fixed value of *OWL*	Not required.
reasoningType	Character String. Fixed value of *Satisfiability* A flag specifying whether or not the response should indicate whether the TypeName (OWL or RDF element) is satisfiable.	Zero or one (optional).

Note that the CSW outputFormat and schemaLanguage attributes are not required. They have fixed values and any values provided for those attributes shall be ignored.

Code 3 contains the XML encoding for the DescribeRecord operation request.

[5] The REQUEST parameter contains the same information as the name of the <DescribeRecord> element in the XML encoding.

[6] The NAMESPACE parameter contains the same information as the xmlns attributes which may be used to define and bind namespaces in XML encoding.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Code 3 – Parameters for XML Encoding for DescribeRecord Operation Request.

```
<xsd:element name="DescribeRecord" type="owlcsw:DescribeRecordType"/>
<xsd:complexType name="DescribeRecordType">
 <xsd:complexContent>
  <xsd:extension base="csw:RequestBaseType">
   <xsd:sequence>
    <xsd:element name="TypeName" type="xsd:QName" minOccurs="0" maxOccurs="unbounded" />
   </xsd:sequence>
   <xsd:attribute name="reasoningType" type="owlcsw:DescribeRecordReasoningType"
use="optional"/>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
<xsd:element name="DescribeRecordReasoningType">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:enumeration value="Satisfiability"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

8.3.3 Operation Response

The following XML Schema fragment defines the response to a DescribeRecord operation under this Application Profile. ParentSchema and schemaLanguage from CSW are not required, the former because all OWL elements exist in their own right in some namespace and can thus be referenced to other OWL elements, and the latter because the schemaLanguage is always OWL.

Code 4 – Parameters for XML Encoding for DescribeRecord Operation Response.

```
<xsd:element name="DescribeRecordResponse" id="DescribeRecordResponse"
type="owlcsw:DescribeRecordResponseType" />
<xsd:complexType name="DescribeRecordResponseType" id="DescribeRecordResponseType">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">The response contains a list of matching OWL or RDF
element.</xsd:documentation>
 </xsd:annotation>
 <xsd:sequence>
  <xsd:element name="SchemaComponent" type="owlcsw:SchemaComponentType" minOccurs="0"
maxOccurs="unbounded" />
 </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SchemaComponentType" mixed="true" id="SchemaComponentType">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">An OWL or RDF element includes the OWL or RDF
description as well as all children of the component.  All referenced namespaces must be
included within the relevant OWL or RDF element.</xsd:documentation>
 </xsd:annotation>
 <xsd:sequence>
  <xsd:any namespace="##any" processContents="lax" />
 </xsd:sequence>
 <xsd:attribute name="targetNamespace" type="xsd:anyURI" use="required" />
 <xsd:attribute name="satisfiability" type="xsd:boolean" use="optional" />
</xsd:complexType>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

The elements in the DescribeRecord response here are generally similar to the elements of the CSW DescribeRecord operation, with the addition of the satisfiability attribute. This is a true/false value that indicates simply whether or not the TypeName (OWL or RDF element) is satisfiable. This is determined by inference over the element.

8.4 GetDomain

8.4.1 Introduction

The GetDomain operation is used to return information about the domains of a specified property or parameter. The operation under this Application Profile shall conform to that described in the CSW specification [OGC 07-006r1], with the modifications described in this Section.

8.4.2 Operation Request

The request parameters are as described in Section 10.7.2 and 10.7.3 of the CSW specification [OGC 07-006r1]. The value of the *service* parameter for services conforming to this Application Profile (including this one) shall be "OWLCSW", and the version shall be "1.0.0".

8.4.2.1 PropertyName Parameter

The PropertyName parameter under this Application Profile may take the value of any of the queryable and returnable properties listed in Table 7 in Section 7.3, using the XML property name (for example, dc:title; owlcsw:BoundingBox). In addition, the PropertyName may specify a particular construct and attribute from the information model to gain the domain for that construct. For example, rdf:Property.ID could be requested to gain a list of all of the property names in an ontology.

8.4.2.2 ParameterName Parameter

The ParameterName parameter does not change in this Application Profile. The set of parameters listed in Section 10.7.4.2 of CSW [OGC 07-006r1] are permitted.

8.4.3 Operation Response

The response format is as described in Section 10.7.5 of the CSW specification [OGC 07-006r1] and is not changed.

8.5 GetRecords

8.5.1 Introduction

The GetRecords operation is used to query and retrieve content from the ontologies that make up the registry. It is based on the same operation in the CSW specification [OGC 07-006r1] with additions and modifications to handle the customised reasoning and querying in this Application Profile.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

8.5.2 Operation Request

Table 11 shows the parameters for the KVP encoding of the GetRecords request, followed by Code 5, which contains the XML encoding. The latter is adapted from the CSW definition with the addition of the reasoning capabilities and the replacement of the CQL_TEXT constraint language with the SPARQL constraint language.

Many of the parameter values are from the CSW specification and more details can be obtained in Section 10.8 of that document [OGC 07-006r1].

Table 11: Parameters for KVP Encoding for GetRecords Operation Request

Parameter	Data type and value	Optionality and use
REQUEST[7]	Character String. Fixed value of *GetRecords*, case insensitive.	One (mandatory)
service	Character String. Fixed value of *OWLCSW*	One (mandatory)
version	Character String. Fixed value of *1.0.0*	One (mandatory)
NAMESPACE[8]	List of Character String, comma separated. The namespaces for any qualified names used in the request. This may include the generic OWL and RDF namespaces if those TypeNames are of interest, or may contain the namespaces relating to specific ontologies (for example, the "is" ontology in Example 5). Format is xmlns([prefix=]namespace-url). If prefix is not specified, then it is the default namespace.	Zero or one (optional). Include declarations for each namespace used in the request.
resultType	Codelist with allowed values: "*hits*", "*results*", "*validate*". Respectively, these values specify return of: o The approximate number of records that meet the request. o The actual records that meet the request. o An acknowledgement that the request is valid and option to retrieve a response after asynchronous processing.	Zero or one (optional). Default value is "*hits*".
requestId	URI	Zero or one (optional).
outputFormat	Character String. Fixed value of *application/rdf+xml*	Not required.

[7] The REQUEST parameter contains the same information as the name of the <DescribeRecord> element in the XML encoding.

[8] The NAMESPACE parameter contains the same information as the xmlns attributes which may be used to define and bind namespaces in XML encoding.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	The CSW specification requires *application/rdf+xml* to be supported. Any document that conforms to *application/rdf+xml* also conforms to *application/rdf+xml.*	
outputSchema	One of the following two URIs: o http://www.opengis.net/cat/owlcsw o http://www.w3.org/2005/sparql-results If the former URI is specified, the result is returned using the OGC Filter record format, as described in Section 10.2.1 of this document. If the latter URI is specified, the result is returned using the SPARQL XML format, as described in Section 8.9.2 of this document. In both cases, embedded OWL may be included within the response.	Zero or one (optional). Default value is owlcsw.
startPosition	Non-zero Positive Integer.	Zero or one (optional). Default value is 1.
maxRecords	Positive Integer.	Zero or one (optional). Default value is 10.
typeNames	List of Character String, comma separated. The list of queryable entities. This may be owlcsw:Record (to return the record in the format described in Section 9) or any OWL or RDF class name from the information model in Section 8.	One (mandatory). Include declarations for each namespace used in a TypeName. Default action is to describe all types known to the server.
ElementName	List of Character String. A set of specific elements that the query should return. If typeName is owlcsw:Record, then the ElementName list should include values from the returnable attributes from owlcsw:Record specified in Section 10.2. If the typeName is a class from the information model, then the ElementName may include specific attributes or roles from the class using XPath notation. Note that the contents of the ontology that are not part of the core queryable and returnable properties can be accessed either through the owlcsw:Record element or directly as an ElementName.	Zero or more (mutually exclusive with ElementSetName)
ElementSetName	List of Character String. A subset of elements that the query should return. If typeName is owlcsw:Record, a subset of properties can be specified as either owlcsw:SummaryRecord or owlcsw:BriefRecord.	Zero or one (mutually exclusive with ElementName)
ConstraintLanguage	Codelist with allowed values: FILTER or SPARQL. This indicates the language of the query request, in constrast to the query response, which is specified in the	Zero or one (optional). Default value is FILTER.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	outputSchema parameter.	
Constraint	Character String. Predicate expression specified in the query language indicated by the ConstraintLanguage parameter.	Zero or one (optional). Default action is to execute an unconstrained query.
SortBy	List of Character String, comma separated. Ordered list of names of elements to sort the response by. Format of each list item is *element_name:A* (ascending order) or *element_name:B* (descending order). The element names should come from the list appearing in ElementName or be a member of the subset included in ElementSetName.	Zero or one (optional). Default action is to present the results in the order in which they are retrieved.
DistributedSearch	Boolean.	Zero or one (optional). Default value is FALSE.
hopCount	Integer.	Zero or one (optional). Include only if DistributedSearch parameter is included. Default value is 2.
ResponseHandler	Any URI.	Zero or one (optional). If not included, process request synchronously.
reasoningType	Codelist with allowed values: *"Subsumption", "Equivalence", "Disjointness", "Consistency", "InstanceChecking", "Retrieval", "Realization".* A non-null value for this parameter indicates that the query specifies a part of the ontology over which the reasoning must be performed, and the response returns the results of that reasoning. Thus a request with the same query value will return different results depending on the value of this parameter. If the value is null, then records that meet the query shall be returned. If the value is non-null then records that result from performance of inference over the records that meet the query shall be returned.	Zero or one (optional).
reasoningHeadQuery	Character String. In the case of "InstanceChecking", and "Realization", a predicate expression specified in the query language indicated by the ConastraintLanguage parameter. In the case of "Retrieval" an OWL fragment.	Zero or one (optional). Mandatory if reasoningType = "InstanceChecking", "Retrieval", "Realization", otherwise NULL.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Code 5 – Parameters for XML Encoding for GetRecords Operation Request.

```
<xsd:element name="GetRecords" type="owlcsw:GetRecordsType" id="GetRecords"/>
<xsd:complexType name="GetRecordsType" id="GetRecordsType">
 <xsd:complexContent>
  <xsd:extension base="csw:RequestBaseType">
   <xsd:sequence>
    <xsd:element ref="csw:DistributedSearch" minOccurs="0" />
    <xsd:element ref="csw:ResponseHandler" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:choice>
     <xsd:element ref="AbstractQuery"/>
     <xsd:any processContents="strict" namespace="##other" />
    </xsd:choice>
   </xsd:sequence>
   <xsd:attribute ref="csw:requestId" use="optional"/>
   <xsd:attribute ref="csw:resultType" use="optional"/>
   <xsd:attribute ref="csw:outputSchema" use="optional"/>
   <xsd:attributeGroup ref="csw:BasicRetrievalOptions"/>
   <xsd:attribute ref="csw:startPosition" use="optional"/>
   <xsd:attribute ref="csw:maxRecords" use="optional"/>
   <xsd:attribute name="reasoningType" type="owlcsw:GetRecordsReasoningType"
use="optional"/>
   <xsd:attribute name="reasoningHeadQuery" type="xsd:string" use="optional"/>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
<xsd:element name="AbstractQuery" type="owlcsw:AbstractQueryType" id="AbstractQuery"
abstract="true" />
<xsd:complexType name="AbstractQueryType" id="AbstractQueryType" abstract="true" />
<xsd:element name="Query" type="owlcsw:QueryType" id="Query"
substitutionGroup="csw:AbstractQuery" />
<xsd:complexType name="QueryType" id="QueryType">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">Specifies a query to execute against instances of one
or more object types. A set of ElementName elements may be included to specify an adhoc
view of the csw:Record instances in the result set. Otherwise, use ElementSetName to
specify a predefined view. The Constraint element contains a query filter expressed in a
supported query language. A sorting criterion that specifies a property to sort by may be
included. typeNames - a list of object types to query.</xsd:documentation>
 </xsd:annotation>
 <xsd:complexContent>
  <xsd:extension base="csw:AbstractQueryType">
   <xsd:sequence>
    <xsd:choice>
     <xsd:element ref="csw:ElementSetName"/>
     <xsd:element ref="csw:ElementName" minOccurs="0" maxOccurs="1"/>
    </xsd:choice>
    <xsd:element ref="owlcsw:Constraint" minOccurs="0" maxOccurs="1" />
    <xsd:element ref="ogc:SortBy" minOccurs="0" maxOccurs="1" />
   </xsd:sequence>
   <xsd:attribute ref="csw:typeNames" use="required" />
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
<xsd:element name="Constraint" type="owlcsw:QueryConstraintType" id="Constraint" />
<xsd:complexType name="QueryConstraintType" id="QueryConstraintType">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">A search constraint that adheres to one of the
following syntaxes: Filter - OGC filter expression CqlText - OGC CQL
predicate</xsd:documentation>
 </xsd:annotation>
 <xsd:choice>
  <xsd:element ref="ogc:Filter"/>
  <xsd:element name="SPARQL" type="xsd:string" minOccurs="0" maxOccurs="1"/>
 </xsd:choice>
 <xsd:attribute name="version" type="xsd:string" use="required">
  <xsd:annotation>
   <xsd:documentation>Query language version</xsd:documentation>
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
   </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:element name="GetRecordsReasoningType">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:enumeration value="Subsumption"/>
   <xsd:enumeration value="Equivalence"/>
   <xsd:enumeration value="Disjointness"/>
   <xsd:enumeration value="Consistency"/>
   <xsd:enumeration value="InstanceChecking"/>
   <xsd:enumeration value="Retrieval"/>
   <xsd:enumeration value="Realization"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

8.5.3 Operation Response

If the reasoningType attribute is non-null, then some kind of reasoning (specified by the parameter value) is applied to the records that meet the query, and the operation response contains the results of the reasoning.

The following table specifies for each type of reasoning the additional constraints that apply in the form of requirements of the results of the query and the values returned after the reasoning has been applied.

Table 12: Query Contents and Responses for Reasoning Types in GetRecords

Reasoning Type	reasoningHeadQuery Parameter Contents	query Parameter Contents	Operation Response
Subsumption	Null	A query that returns a set of two or more classes.	The subsumption hierarchy is returned. If none of the classes are subsumed by others, then the empty set is returned.
Equivalence	Null	A query that returns a pair of classes.	If the two classes are equivalent, then their OWL description is returned. Otherwise the empty set is returned (either both are returned or none).
Disjointness	Null	A query that returns a pair of classes.	If the two classes are disjoint, then their OWL description is returned. Otherwise the empty set is returned (either both are returned or none).
Consistency	Null	A query that returns a set of individuals in an ontology.	If the two classes are consistent, then their OWL description is returned. Otherwise the empty set is returned (either both are returned or none).
InstanceChecking	A query that returns a single individual in an	A query that returns a set of individuals in an	If the individual in reasoningHeadQuery is entailed by

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	ontology.	ontology.	the set of individuals, then that individual is returned. Otherwise the empty set if returned.
Retrieval	A string containing an OWL description.	A query that returns a set of individuals in an ontology.	The members of the set of individuals that fulfil the OWL description in reasoningHeadQuery are returned. If none of the individuals fulfil the description, then the empty set is returned.
Realization	A query that returns a single individual in an ontology.	A query that returns a set of classes.	The most specific OWL classes that the individual in reasoningHeadQuery is a member of are returned. If the individual is not a member of any of the OWL classes, the empty set is returned.

The format of the GetRecordsResponse element under this Application Profile is the same as in Section 10.8.5 of the CSW specification [OGC 07-006r1]. The contents of the Record element within the GetRecordsResponse wrapper is as shown in Section 8.9.2, depending on the value of outputSchema.

8.6 GetRecordById

8.6.1 Introduction

The GetRecordById operation is used to retrieve particular records from the ontologies using an ID value that is already known. The GetRecordById operation in this Application Profile is extended relative to the GetRecordById operation in the CSW specification [OGC 07-006r1] to allow retrieval either of predetermined metadata subsets or of a specific element from the information model.

8.6.2 Operation Request

Table 13 describes the KVP encoding for the GetRecordById request, followed by the XML encoding in Code 6.

Table 13: Parameters for KVP Encoding for GetRecordById Operation Request

Parameter	Data type and value	Optionality and use
REQUEST[9]	Character String. Fixed value of *GetRecordById*, case insensitive.	One (mandatory)
service	Character String.	One (mandatory)

[9] The REQUEST parameter contains the same information as the name of the <DescribeRecord> element in the XML encoding.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	Fixed value of *OWLCSW*	
version	Character String. Fixed value of *1.0.0*	One (mandatory)
NAMESPACE[10]	List of Character String, comma separated. The namespaces for any qualified names used in the request. This may include the generic OWL and RDF namespaces if those TypeNames are of interest, or may contain the namespaces relating to specific ontologies (for example, the "is" ontology in Example 5). Format is xmlns([prefix=]namespace-url). If prefix is not specified, then it is the default namespace.	Zero or one (optional). Include declarations for each namespace used in the request.
ElementSetName	Codelist with allowed values: "*owlcsw:Record", "owlcsw:SummaryRecord", "owlcsw:BriefRecord*". A subset of elements that the query should return. Details of the content of each subset are contained in Section 10.2.	Zero or one (mutually exclusive with ElementName)
ElementName	Character String. The name of a specific element from the information model (e.g. owl:Class).	Zero or more (mutually exclusive with ElementSetName)
outputFormat	Character String. Fixed value of *application/rdf+xml* The CSW specification requires *application/rdf+xml* to be supported. Any document that conforms to *application/rdf+xml* also conforms to *application/rdf+xml*.	Not required.
outputSchema	One of the following two URIs: o http://www.opengis.net/cat/owlcsw o http://www.w3.org/2005/sparql-results If the former URI is specified, the result is returned using the OGC Filter record format, as described in Section 10.2.1 of this document. If the latter URI is specified, the result is returned using the SPARQL XML format, as described in Section 8.9.2 of this document. In both cases, embedded OWL may be included within the response.	Zero or one (optional). Default value is owlcsw.
Id	Comma separated list of anyURI If the element is requested using ElementSetName (and is thus one of the predefined subsets, then the Id value refers to dc:identifier from the set of queryable and returnable properties. This property is mandatory in all three of the predefined subsets.	One (mandatory)

[10] The NAMESPACE parameter contains the same information as the xmlns attributes which may be used to define and bind namespaces in XML encoding.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

	If the element is requested using ElementName and thus refers to a class in the information model, then the Id refers to the concatenation of the namespace and the ID from the rdfs:Resource class (inherited by all classes in the information model).	

Code 6 – Parameters for XML Encoding for GetRecordById Operation Request.

```
<xsd:element name="GetRecordById" type="owlcsw:GetRecordByIdType" id="GetRecordById" />
<xsd:complexType name="GetRecordByIdType" id="GetRecordByIdType">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">Convenience operation to retrieve default record
representations and elements by identifier.</xsd:documentation>
 </xsd:annotation>
 <xsd:complexContent>
  <xsd:extension base="csw:RequestBaseType">
   <xsd:sequence>
    <xsd:element name="Id" type="xsd:anyURI" maxOccurs="unbounded" />
    <xsd:choice>
     <xsd:element ref="csw:ElementSetName"/>
     <xsd:element ref="csw:ElementName" minOccurs="0" maxOccurs="1"/>
    </xsd:choice>
   </xsd:sequence>
   <xsd:attribute ref="csw:outputSchema" use="optional" />
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

8.6.3 Operation Response

The format of the GetRecordByIdResponse element under this Application Profile is the same as in Section 10.9.5 of the CSW specification [OGC 07-006r1]. If the ElementSetName parameter is used, then the contents of the Record element within the GetRecordByIdResponse wrapper is as shown in Section 8.9.2, depending on the value of outputSchema. If the ElementName parameter is used, then the contents of the Record element is the OWL description for the selected element from the information model.

8.7 Record Locking

This Application Profile does not define a locking interface, instead relying on the underlying repository to mediate concurrent access to catalogue records.

8.7 Transaction

8.7.1 Introduction

The Transaction operation for this Application Profile conforms to the CSW specification [OGC 07-006r1] with minor additional constraints regarding payloads as described in this Section. The Transaction parameters are specified in the CSW specification, but include

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

the same service and version parameters as the other operations. For implementations that conform to this specification, "OWLCSW" and "1.0.0" must be used respectively.

8.7.2 Insert Action

Requests and responses that use the Insert action shall conform to the CSW specification [OGC 07-006r1]. In that specification, the Insert element is described, and it is a container for one or more records that may be inserted into the catalogue. For implementations that conform to this Application Profile, the records that appear within the Insert element container shall conform to the information model described in Section 7.2 and also to the OWL [W3C OWL] and/or RDF(S) [W3C RDF; W3C RDFS] specifications. That is, as immediate children of the Insert element, OWL and RDF elements as described in the information model must appear.

EXAMPLE 1 – Insert Request:

```
<Transaction service="OWLCSW" version="1.0.0">
 <Insert>
    <is:JournalArticle
rdf:ID="The determination of alkylphenols in aqueous samples from the Forth Estuary by SPE
-HPLC-fluorescence-JournalArticle"
    xmlns:is="http://compass.edina.ac.uk/ontologies/informationsource0-4.owl#"
    xml:base="http://compass.edina.ac.uk/ontologies/Smith01ChemPub17.owl">
   <is:has-author>
    <is:Author rdf:ID="Coffey M.">
     <is:has-lastname
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Coffey</is:has-lastname>
      <is:has-firstname rdf:datatype="http://www.w3.org/2001/XMLSchema#string">M</is:has-
firstname>
    </is:Author>
          </is:has-author>
  </is:JournalArticle>
 </Insert>
</Transaction>
```

8.7.3 Update Action

The update action is the same as for the CSW specification [OGC 07-006r1], except for the constraint format, which is modified to match the different query languages available in this Application Profile. The constraint element is defined in Section 8.5, and is used to identify the set of elements in the ontologies that the update action will affect. The constraint element is required to avoid accidental update of all records.

The <RecordProperty> element is used to define the actual updates that are to be made. The <Name> element shall refer to an element in the information model, and the corresponding <Value> element shall contain the new value for the element specified by <Name>. The <Value> element may contain an internally complex object (for example, an OWL class), but it must be valid OWL for the element type specified in <Name>.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Code 7 – Parameters for XML Encoding for Transaction Operation Request Update Action.

```
<xsd:complexType name="UpdateType" id="UpdateType">
 <xsd:sequence>
  <xsd:choice>
   <xsd:any processContents="strict" namespace="##other" />
   <xsd:sequence>
    <xsd:element ref="csw:RecordProperty" maxOccurs="unbounded" />
    <xsd:element ref="owlcsw:Constraint" />
   </xsd:sequence>
  </xsd:choice>
 </xsd:sequence>
 <xsd:attribute name="handle" type="xsd:ID" use="optional" />
</xsd:complexType>
```

8.7.4 Delete Action

The following XML Schema fragment defines the Delete action in this Application Profile:

Code 8 – Parameters for XML Encoding for Transaction Operation Request Delete Action.

```
<xsd:complexType name="DeleteType" id="DeleteType">
 <xsd:sequence>
  <xsd:element ref="owlcsw:Constraint" minOccurs="1" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="typeName" type="xsd:anyURI" use="optional" />
 <xsd:attribute name="handle" type="xsd:ID" use="optional" />
</xsd:complexType>
```

This is to be interpreted in the same was as for the CSW specification [OGC 07-006r1], the only modification being the change in query languages used in this Application Profile.

8.7.5 Operation Response

The operation response shall be as for the CSW specification [OGC 07-006r1].

8.8 Harvest

8.8.1 Introduction

Harvesting in this Application Profile shall be as for the CSW specification [OGC 07-006r1], with a minor modification involving the addition of the capability to specify topics for harvesting. The original intention for harvesting is that an entire repository might be harvested into the catalogue. However, sometimes only resources relating to a particular theme may be required (for example, if a registry of marine science resources is being created), so this Application Profile includes a very high level filtering mechanism.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Other than this, the Harvest operation under this Application Profile is the same as for CSW.

8.8.2 Operation Request

Table 14 lists the parameters required for a request for the Harvest operation.

Table 14: Parameters for KVP Encoding for Harvest Operation Request

Parameter	Data type and value	Optionality and use
REQUEST[11]	Character String. Fixed value of *Harvest*, case insensitive.	One (mandatory)
service	Character String. Fixed value of *OWLCSW*	One (mandatory)
version	Character String. Fixed value of *1.0.0*	One (mandatory)
NAMESPACE[12]	List of Character String, comma separated. The namespaces for any qualified names used in the request. This may include the generic OWL and RDF namespaces if those TypeNames are of interest, or may contain the namespaces relating to specific ontologies (for example, the "is" ontology in Example 5). Format is xmlns([prefix=]namespace-url). If prefix is not specified, then it is the default namespace.	Zero or one (optional). Include declarations for each namespace used in the request.
Source	URI. Reference to the source from which the resource is to be harvested.	One (mandatory)
ResourceType	Character String. Reference to the type of resource being harvested, as per the CSW.	One (mandatory)
Topic	List of Character String, comma separated. A set of topics about which resources are to be harvested. The topics specified must be from the taxonomy used by the specified Source, as there are multiple taxonomies and topic listings used for resources. The taxonomy is often determinable from the ResourceType of the Source.	Zero or one (optional). If not included, harvest all resources.
ResourceFormat	Character String. MIME type indicating the format of the resource being harvested.	Zero or one (optional). Default value of *application/rdf+xml*

[11] The REQUEST parameter contains the same information as the name of the <DescribeRecord> element in the XML encoding.

[12] The NAMESPACE parameter contains the same information as the xmlns attributes which may be used to define and bind namespaces in XML encoding.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

ResponseHandler	URL A reference to a person or entity that the CSW should respond to when it has completed processing Harvest request asynchronously.	Zero or one (optional). If not included, process request asynchronously.
HarvestInterval	Period Must conform to ISO8601 Period syntax.	Zero or one (optional). If not included, havest once.

Code 9 – Parameters for XML Encoding for Harvest Operation Request.

```
<xsd:element name="Harvest" type="owlcsw:HarvestType"/>
<xsd:complexType name="HarvestdType">
 <xsd:complexContent>
  <xsd:extension base="csw:HarvestType">
   <xsd:sequence>
    <xsd:element name="Topic" type="xsd:String" minOccurs="0"/>
   </xsd:sequence>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

The implementation of the Harvest operation is dependent on the structure of the Source, as identified in the ResourceType parameter. Therefore, particular implementations are likely to implement Harvest operations for only some ResourceTypes, and this must be advertised in the GetCapabilities document. In addition to those listed in the CSW specification [OGC 07-006r1], the following possibilities may be used. These are repository standards for digital libraries that may contain useful resources for a registry.

Table 15: URIs for Additional Resource Repository Standards

URI	Name
http://www.loc.gov/z3950/agency/	Z39.50
http://www.openarchives.org/OAI/openarchivesprotocol.html	OAI-PMH
http://www.loc.gov/standards/sru/	SRU-W

8.8.2 Operation Response

The operation response must be the same as for CSW [OGC 07-006r1], the only change being that it may only include resources that only fall into the specified topics, rather than all resources.

8.9 Query Facilities

This Application Profile requires that two query languages be supported for operations that include a query capability:

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

- FILTER. This is an XML encoding of the OGC Filter Encoding Specification [OGC 04-095], is used widely across OGC specifications and is required for all CSW implementations.

- SPARQL [W3C SPARQL]. This is a query language that is designed for use with RDF, and is used widely in the semantic web.

CSW [OGC 07-006r1] also offers the option to support the CQL_TEXT query language, but this Application Profile does not include that query language and it cannot be assumed to be supported by implementations.

Both of these languages are included to promote interoperability across different catalogues (particularly with other Application Profiles of CSW), as supported by FILTER; and to make available the query language of the semantic web, as provided by SPARQL. SPARQL is used by many reasoners, and implementations of this Application Profile may use reasoners to perform queries over the underlying OWL ontologies. Thus this approach allows a SPARQL query to be passed into a request under this Application Profile, and then relayed to reasoning software underneath. The requester must specify the query language of both request and response as listed under the parameters for each relevant interface (Section 8).

The CSW requirement for an XML encoding of the operations of this Application Profile (in addition to the KVP encoding) necessitates both KVP and XML encodings of both query languages for requests and responses. This section defines the schemas for all four of these possibilities using the queryable and returnable properties from Section 7.3.

8.9.1 Format of Query Requests

Queries conforming to this specification may optionally include the properties listed as queryable in Section 7.3.

8.9.1.1 Format of FILTER Query Requests

8.9.1.1.1 KVP Encoding

The KVP encoding of a query is normally a simple text string. However, the FILTER specification [OGC 04-095] only provides an XML encoding, so the KVP encoding of a query shall contain a text string that is the XML encoding of the <ogc:filter> element and all of its children.

The required result set (Brief, Summary or Full as described in Section 8.9.2) is specified using attributes of the KVP encoding for relevant interfaces (see Section 8.1 to 8.8), and is not included in the actual query in the way that it is for the XML encoding (Section 8.9.1.1.2).

An example KVP-encoded FILTER query is shown below.

EXAMPLE 2 - KVP Encoding of a FILTER Query Request:

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
<ogc:Filter>
 <ogc:And>
  <ogc:PropertyIsEqualTo>
   <ogc:PropertyName>/Record/title</ogc:PropertyName>
   <ogc:Literal>Low tide distribution of wintering waders and shelduck on the Severn
Estuary in relation to the proposed tidal barrage</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
   <ogc:PropertyName>/Record/subject</ogc:PropertyName>
   <ogc:Literal>BeachTopography</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsGreaterThanOrEqualTo>
   <ogc:PropertyName>/Record/modified</ogc:PropertyName>
   <ogc:Literal>2004-03-01</ogc:Literal>
  </ogc:PropertyIsGreaterThanOrEqualTo>
  <ogc:PropertyIsBetween>
   <owlcsw:BoundingBox crs="urn:ogc:def:crs:OGC::wxx">
    <owlcsw:LowerCorner>
     < owlcsw: x>14.05</ owlcsw: x>
     < owlcsw: y>46.46</ owlcsw: y>
    </owlcsw:LowerCorner>
    <owlcsw:UpperCorner>
     < owlcsw: x>17.24</ owlcsw: x>
     < owlcsw: y>48.42</ owlcsw: y>
    </owlcsw:UpperCorner>
   </owlcsw:BoundingBox>
  </ogc:PropertyIsBetween>
 </ogc:And>
</ogc:Filter>
```

8.9.1.1.2 XML Encoding

The XML encoding of a FILTER query shall conform to the OGC Filter Encoding Specification [OGC 04-095] as described in the CSW Specification [OGC 07-006r1] for the csw:Query element. The details of this specification are not formally described here.

Those queryable properties that map directly to a value with a simple data type (see Section 8.9.2 for a definition of the data types for all properties) can be included using a simple PropertyName and Literal. Properties that have complex data types (for example, owlcsw:BoundingBox) must be queried using the internal structure of those data types. The required result set (Brief, Summary or Full, as described in Section 8.9.2) is specified in the <csw:ElementName> or <csw:ElementSetName> element of the query.

An example XML-encoded FILTER query is shown below.

EXAMPLE 3 – XML Encoding of a FILTER Query Request:

```
<csw:Query>
 <!-- this returns the full record, otherwise use owlcsw:BriefRecord or
owlcsw:SummaryRecord -->
 <csw:ElementName>/owlcsw:Record</csw:ElementName>
 <csw:Constraint version="1.0.0">
  <ogc:Filter>
   <ogc:And>
    <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>/Record/title</ogc:PropertyName>
     <ogc:Literal>Low tide distribution of wintering waders and shelduck on the Severn
Estuary in relation to the proposed tidal barrage</ogc:Literal>
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>/Record/subject</ogc:PropertyName>
     <ogc:Literal>BeachTopography</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsGreaterThanOrEqualTo>
     <ogc:PropertyName>/Record/modified</ogc:PropertyName>
     <ogc:Literal>2004-03-01</ogc:Literal>
    </ogc:PropertyIsGreaterThanOrEqualTo>
    <ogc:PropertyIsBetween>
     <owlcsw:BoundingBox crs="urn:ogc:def:crs:OGC::wxx">
      <owlcsw:LowerCorner>
       <owlcsw:x>14.05</owlcsw:x>
       <owlcsw:y>46.46</owlcsw:y>
      </owlcsw:LowerCorner>
      <owlcsw:UpperCorner>
       <owlcsw:x>17.24</owlcsw:x>
       <owlcsw:y>48.42</owlcsw:y>
      </owlcsw:UpperCorner>
     </owlcsw:BoundingBox>
    </ogc:PropertyIsBetween>
   </ogc:And>
  </ogc:Filter>
 </csw:Constraint>
</csw:Query>
```

8.9.1.2 Format of SPARQL Query Requests

SPARQL offers four different query forms: SELECT, CONSTRUCT, ASK and DESCRIBE. As a minimum, implementations of this Application Profile that support SPARQL must support the SELECT query forms. Other query forms may be supported optionally. In either case, the supported query forms must be described by the GetCapabilties operation.

8.9.1.2.1 KVP Encoding

The KVP encoding of a query is a simple text string. This shall contain a Character String that conforms to the SPARQL specification, and is valid for that specification.

EXAMPLE 4 - KVP Encoding of a SPARQL Query Request:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX owlcsw: < http://www.opengis.net/cat/owlcsw.owl#>

SELECT ?title ?subject ?modified

WHERE
{
?x       dc:title                    ?title            .
FILTER regex(?title, "Low tide distribution of wintering waders and shelduck on the Severn
Estuary in relation to the proposed tidal barrage", "i")
?x       dc:subject                  ?subject          .
FILTER regex(?subject, "BeachTopography")
?x       dct:modified                ?modified         .
FILTER (?modified > xsd:dateTime("2004-03-01T00:00:00Z")
?x       owlcsw:boundingbox          ?owlcsw:LowerCorner         .
?x       owlcsw:boundingbox          ?owlcsw:UpperCorner         .
?y       owlcsw:LowerCorner          ?owlcsw:lx        .
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
?y       owlcsw:LowerCorner          ?owlcsw:ly         .
?z       owlcsw:UpperCorner          ?owlcsw:ux         .
?z       owlcsw:UpperCorner          ?owlcsw:uy         .
FILTER (?owlcsw:lx > 14.05 && ?owlcsw:ly > 46.46 && ?owlcsw:ux > 17.24 && ?owlcsw:uy >
48.42)
}
```

8.9.1.2.2 XML Encoding

This application profile adopts the SPARQL Protocol for RDF [W3C SPARQL WSDL], which specifies how a SPARQL query can be encoded using WSDL [W3C WSDL]. An XML encoding of SPARQL is required for submission in an XML encoded CSW web service (for example, with HTTP POST), as required by the CSW specification. The reader is referred to the specification for the SPARQL Protocol for RDF [W3C SPARQL WSDL] for full details of the syntax of a request using that format, and an example is provided below. The SPARQL Protocol for RDF specification provides examples using a SOAP envelope. This conforms to the requirements for CSW, as discussed further in Section 10.3.2 of the CSW Specification [07-006r1], but the following example shows only the portion of a query that would appear within the SOAP envelope (as in Section 8.9.2.2).

EXAMPLE 5 – XML Encoding of a SPARQL Query Request:

```
<st:query-request>
 <st:query>SELECT ?title ?subject ?modified {
  ?x http://purl.org/dc/elements/1.1/title ?title . FILTER regex(?title, "Low tide
distribution of wintering waders and shelduck on the Severn Estuary in relation to the
proposed tidal barrage", "i")
  ?x http://purl.org/dc/elements/1.1/subject ?subject . FILTER regex(?subject,
"BeachTopography")
  ?x http://purl.org/dc/terms/modified ?modified  . FILTER (?modified >
http://www.w3.org/2001/XMLSchema/dateTime("2004-03-01T00:00:00Z")
  ?x http://www.opengis.net/cat/owlcsw/boundingbox
 ?http://www.opengis.net/cat/owlcsw/LowerCorner  .
  ?x http://www.opengis.net/cat/owlcsw/boundingbox
 ?http://www.opengis.net/cat/owlcsw/UpperCorner .
  ?y http://www.opengis.net/cat/owlcsw/LowerCorner ?http://www.opengis.net/cat/owlcsw/lx.
  ?y http://www.opengis.net/cat/owlcsw/LowerCorner ?http://www.opengis.net/cat/owlcsw/ly.
  ?z http://www.opengis.net/cat/owlcsw/UpperCorner ?http://www.opengis.net/cat/owlcsw/ux.
  ?z http://www.opengis.net/cat/owlcsw/UpperCorner ?http://www.opengis.net/cat/owlcsw/uy.
FILTER (?http://www.opengis.net/cat/owlcsw/lx > 14.05 &amp;amp;&amp;amp;
?http://www.opengis.net/cat/owlcsw/ly > 46.46 &amp;amp;&amp;amp;
?http://www.opengis.net/cat/owlcsw/ux > 17.24 &amp;amp;&amp;amp;
?http://www.opengis.net/cat/owlcsw/uy > 48.42)}</st:query>
</st:query-request>
```

8.9.2 Format of Query Responses

CSW defines three subsets of properties that may be returned in response to queries. This Section describes the contents of each subset and the syntax in both the FILTER and SPARQL languages. In all of the subsets, only the title and identifier are mandatory.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Implementations of this Application Profile must return a Record element that represents the object to which the queryable properties apply. For example, if a query request specifies a title and subject for a resource, the returned Record element must represent the resource that has that title and subject. How this is done will depend on the specific implementation of this Application Profile, and specifically the structure of the ontology that is stored within the registry. If the ontology is designed such that the queryable properties are immediate children of the element to which they apply, the results can be most easily returned in the desired format (particularly in the case of SPARQL queries). However, if the ontology contains several layers of nesting of the queryable properties within the resource to which they apply, then implementations of this Application Profile must ensure that the correct level of resource is returned as the Record element. This Application Profile simply assumes that a query using the queryable properties will return a Record that represents the resource to which those properties apply.

The format used for the query request does not dictate the format of the query response. Sections 8.2 to 8.8 describe how both languages may be specified for an interface. A user may provide a query request in one language and receive a response in another language if she chooses.

8.9.2.1 Format of FILTER Response

Responses that conform to the FILTER query language return a Record element for each resource that fulfils the query request. The following subsections provide the schemas for the different versions of the Record element. The Brief and Summary Record elements are similar to those offered by the generic CSW specification [OGC 07-006r1], but the Full Record element has some variations.

8.9.2.1.1 Full Record

The Full Record that is returned by implementations of this Application Profile in response to a query is similar to the CSW Full Record element, but contains a modification in the Bounding Box and the addition of elements to allow a full OWL ontology to be returned. Note that the inclusion of a particular OWL element would automatically include all of that element's children.

The representation of OWL in the query response is intended to conform to the OWL specification and have a similar structure to that shown in that specification [W3C OWL]. If the description of the representation in this Application Profile is not clear, the OWL specification should be used as a guideline.

The specification of the Full Record appears below.

Code 10 – Definition of a Full Record using a Filter Response:

```
<xsd:element name="Record" type = "owlcsw:RecordType"="csw:AbstractRecord"/>
<xsd:complexType name="RecordType">
 <xsd:complexContent>
  <xsd:extension base="csw:DCMIRecordType">
   <xsd:sequence>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
     <xsd:element name="AnyText" type="csw:EmptyType" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owlcsw:BoundingBox" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="rdfs:Resource" type="rdfs:Resource" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="rdfs:Class" type="rdfs:Class" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="rdfs:Literal" type="rdfs:Literal" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="rdf:XMLLiteral" type="rdf:XMLLiteral" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="rdf:Property" type="rdf:Property" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="rdfs:Datatype" type="rdf:Datatype" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="rdf:List" type="rdf:List" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="rdf:Container" type="rdf:Container" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="rdf:Alt" type="rdf:Alt" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="rdf:Bag" type="rdf:Bag" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="rdf:Seq" type="rdf:Seq" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="rdfs:ContainerMembershipProperty"
type="rdf:ContainerMembershipProperty" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="owl:Ontology" type="owl:Ontology" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:Class" type="owl:Class" minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="owl:AllDifferent" type="owl:AllDifferent" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:DataRange" type="owl:DataRange" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:DeprecatedClass" type="owl:DeprecatedClass" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:Restriction" type="owl:Restriction" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:DeprecatedProperty" type="owl:DeprecatedProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:FunctionalProperty" type="owl:FunctionalProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:OntologyProperty" type="owl:OntologyProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:AnnotationProperty" type="owl:AnnotationProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:DatatypeProperty" type="owl:DatatypeProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:ObjectProperty" type="owl:ObjectProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:SymmetricProperty" type="owl:SymmetricProperty" minOccurs="0"
maxOccurs="unbounded"/>
     <xsd:element ref="owl:InverseFunctionalProperty" type="owl:InverseFunctionalProperty"
minOccurs="0" maxOccurs="unbounded"/>
     <xsd:element ref="owl:TransitiveProperty" type="owl:TransitiveProperty" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
   </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

8.9.2.1.1 Summary Record

The Summary Record is designed to allow an important subject of information about a resource in the registry to be returned without the full ontology. The user (human or machine) may then decide whether to retrieve the full ontology information for the resource.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

The following XML identifies the format for a Summary Record under this Application Profile.

Code 11 – Definition of a Summary Record using a Filter Response:

```
<xsd:element name="SummaryRecord"
 = "owlcsw:SummaryRecordType"
 ="csw:AbstractRecord"/>
<xsd:complexType name="SummaryRecordType">
 <xsd:complexContent>
  <xsd:extension base="csw:AbstractRecordType">
   <xsd:sequence>
    <xsd:element="dc:identifier" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element="dc:title" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element="dc:type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element="dc:subject" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element="dc:format" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element="dc:relation" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element="dct:modified" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element="dct:abstract" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="owlcsw:BoundingBox" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

8.9.2.1.1 Brief Record

The Brief Record contains a more limited set of information about a resource. The following XML identifies the format for a Brief Record under this Application Profile.

Code 11 – Definition of a Brief Record using a Filter Response:

```
<xsd:element name="BriefRecord"
 = "owlcsw: BriefRecordType"
 ="csw:AbstractRecord"/>
<xsd:complexType name="BriefRecordType">
 <xsd:complexContent>
  <xsd:extension base="csw:AbstractRecordType">
   <xsd:sequence>
    <xsd:element="dc:identifier" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element="dc:title" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element="dc:type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="owlcsw:BoundingBox" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

8.9.2.2 Format of SPARQL Response

The contents of the SPARQL responses are identical to the subsets of properties contained in the Full, Summary and Brief records for a FILTER query response.

The format of the SPARQL responses must conform to the SPARQL Query Results XML Format [W3C SPARQL XML], which is the same as that contained in the

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

SPARQL Protocol for RDF that uses WSDL [W3C SPARQL WSDL]. The latter combines the former with a SOAP envelope.

In the SPARQL response, the Result element is equivalent to the Record, SummaryRecord and BriefRecord elements in the FILTER response. Each Result element contains a description for a resource, with all relevant children and attributes.

Below is an example of a SPARQL Response to a request for a Full Record. It shows a set of specifically identified attributes, together with the OWL Class for the Resource that fulfils the query from an example OWL ontology (with the "is" namespace). In some cases the queryable and returnable properties of the resource are not immediate children of the resource, so additional levels of nesting must be returned. This Application Profile is designed to be used with any OWL ontology, and this example does not prescribe any particular structure, it merely illustrates how an ontology might appear in a SPARQL query response.

EXAMPLE 6 – SPARQL Query Response for Full Result Set:

```
<st:query-result>
 <sr:sparql>
  <sr:head>
   <sr:variable name="identifier"/>
   <sr:variable name="title"/>
   <sr:variable name="subject"/>
   <sr:variable name="type"/>
   <sr:variable name="modified"/>
   <sr:variable name="abstract"/>
   <sr:variable name="format"/>
   <sr:variable name="creator"/>
   <sr:variable name="language"/>
   <sr:variable name="owlcsw:BoundingBox"/>
   <sr:variable name="owl:Class"/>
  </sr:head>
  <sr:results distinct="false" ordered="false">
   <sr:result>
    <sr:binding name="identifier">
 <sr:literal>http://compass.edina.ac.uk/ontologies/Clark94WadersPub23.owl#Low_tide_distri
bution_of_wintering_waders_and_shelduck_on_the_Severn_Estuary_in_relation_to_the_proposed_
tidal barrage-JournalArticle</sr:literal>
    </sr:binding>
    <sr:binding name="title">
     <sr:literal>Low tide distribution of wintering waders and shelduck on the Severn
Estuary in relation to the proposed tidal barrage</sr:literal>
    </sr:binding>
    <sr:binding name="subject">
     <sr:literal>BeachTopography</sr:literal>
    </sr:binding>
    <sr:binding name="type">
     <sr:literal>publication</sr:literal>
    </sr:binding>
    <sr:binding name="modified">
     <sr:literal>2006-08-23</sr:literal>
    </sr:binding>
    <sr:binding name="abstract">
     <sr:literal>Waders and shelduck were counted at low tide on 162 sectors comprising
85% of the intertidal area (21 467 ha) of the Severn Estuary on 12 occasions during winter
1987/88. On average, 50% of birds present at low tide utilized just 13 sectors (12% of the
area); 90% of birds occurred on only 56 sectors, leaving large expanses of intertidal sand
virtually devoid of birdlife. Dunlin, the numerically dominant species, occurred widely on
the middle and outer estuary, whereas shelduck predominantly occurred on the outer estuary
and redshank around many tributary river mouths. Curlew, the most ubiquitous species, was
the only one concentrated on the inner estuary. Severe gales in both late December and
```


Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
mid-January concentrated all main species within fewer sectors, probably by the short-term
removal of surface sediment from substantial areas. It is estimated that the proposed
tidal barrage would eliminate intertidal areas accounting for between c. 40% (for shelduck
and curlew) and 80% (for redshank) of current total low tide usage by the internationally
important populations present.</sr:literal>
     </sr:binding>
     <sr:binding name="format">
      <sr:literal>pdf</sr:literal>
     </sr:binding>
     <sr:binding name="creator">
      <is:Author rdf:ID="Clark N.A. 1"
xmlns:is="http://compass.edina.ac.uk/ontologies/informationsource0-4.owl#">
       <is:has-secondname
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A</is:has-secondname>
       <is:has-firstname
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">N</is:has-firstname>
       <is:has-lastname
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Clark</is:has-lastname>
      </is:Author>
      <is:Author rdf:ID="Prys-Jones P.R."
xmlns:is="http://compass.edina.ac.uk/ontologies/informationsource0-4.owl#">
       <is:has-lastname rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Prys-
Jones</is:has-lastname>
       <is:has-firstname
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P</is:has-firstname>
       <is:has-secondname
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">R</is:has-secondname>
      </is:Author>
     </sr:binding>
     <sr:binding name="language">
      <sr:literal>en</sr:literal>
     </sr:binding>
     <sr:binding name="owlcsw:BoundingBox">
      <owlcsw:BoundingBox crs="urn:ogc:def:crs:OGC::wxx">
       <owlcsw:LowerCorner>
        <owlcsw:x>15.25</owlcsw:x>
        <owlcsw:y>30.46</owlcsw:y>
       </owlcsw:LowerCorner>
       <owlcsw:UpperCorner>
        <owlcsw:x>16.44</owlcsw:x>
        <owlcsw:y>40.42</owlcsw:y>
       </owlcsw:UpperCorner>
      </owlcsw:BoundingBox>
     </sr:binding>
     <sr:binding name="owl:Class">
      <is:JournalArticle
rdf:ID="Low_tide_distribution_of_wintering_waders_and_shelduck_on_the_Severn_Estuary_in_re
lation_to_the_proposed_tidal_barrage-JournalArticle"
xmlns:is="http://compass.edina.ac.uk/ontologies/informationsource0-4.owl#">
       <is:has-title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Low tide
distribution of wintering waders and shelduck on the Severn Estuary in relation to the
proposed tidal barrage</is:has-title>
       <is:has-publisher>
        <is:Publisher rdf:ID="Wiley-Blackwell"></is:Publisher>
        </is:has-publisher>
       <is:has-date-of-publication>
        <is:DateOfPublication rdf:ID="DateOfPublication">
         <is:date-has-value rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1994-01-
01</is:date-has-value>
        </is:DateOfPublication>
       </is:has-date-of-publication>
       <is:has-digital-realization>
        <is:File rdf:ID="pdf">
         <is:has-address>
          <is:URL rdf:ID="URL">
           <is:has-value
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">http://www.blackwell-
synergy.com/doi/abs/10.1111/j.1095-8312.1994.tb00954.x</is:has-value>
           <is:has-value
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">DOI:10.1111/j.1095-
8312.1994.tb00954.x</is:has-value>
```

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

```
          </is:URL>
         </is:has-address>
        </is:File>
       </is:has-digital-realization>
       <is:has-lastpage rdf:datatype="http://www.w3.org/2001/XMLSchema#int">217</is:has-
lastpage>
       <is:has-firstpage rdf:datatype="http://www.w3.org/2001/XMLSchema#int">199</is:has-
firstpage>
      </is:JournalArticle>
     </sr:binding>
    </sr:result>
   </sr:results>
  </sr:sparql>
</st:query-result>
```

8.10 Implementation Guidance

Implementers are advised that a range of useful software is available to assist with the manipulation of ontologies and reasoning over them.

Ontologies are dynamic and variable and applications that use this Profile may allow for ontology editing. Discovery operations like GetCapabilities and GetDomain may be used to discover the ontology in preference to assuming a stable structure.

8.11 Security Considerations

Security is not addressed in this Application Profile. Security functions are not provided by the Profile.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex A
(normative)

Abstract Test Suite

A.1 Test Module for General Capabilties

A.1.1 Test case for GetCapabilities

a) Test purpose(s): The GetCapabilities document must accurately describe the services that the catalogue offers.

b) Test method: For all capabilities described by GetCapabilities, validate the reported capability against that acturally provided. Pass if validation succeeds. Fail otherwise.

Test type: Basic

A.1.2 Test case for OWL Files

c) Test purpose(s): The interfaces must access valid OWL files and be able to interpret them.

d) Test method: For any valid OWL file, validate the ability of the catalogue to access, read and return something from the OWL file. Pass if validation succeeds. Fail otherwise.

Test type: Basic

A.2 Test module for Discovery operations

A.2.1 Test case for valid operation request

e) Test purpose(s): The body of a request must conform to the operation specification and the information model specified in this document.

f) Test method: For all operation request messages, validate the request against its schema definition and against the information model. Pass if validation succeeds. Fail otherwise.

Test type: Basic

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

A.2.2 Test case for valid query response message

g) Test purpose(s): The body of a query response message must conform to the information model, the query response format specified in this document and the OWL specification.

h) Test method: For all query response messages, validate the response against its schema definition and against the OWL specification. Pass if validation succeeds. Fail otherwise.

Test type: Basic

A.2.3 Test case for query language

i) Test purpose(s): The catalogue must support the query languages as advertised.

j) Test method: For both query languages, validate both request and response to ensure that the query language is understood and supported. Pass if validation succeeds. Fail otherwise.

Test type: Basic

A.2.4 Test case for queryable and returnable properties

k) Test purpose(s): The queryable properties must be understood as reported and understandable and applicable parent objects must be returned.

l) Test method: For all operations, validate the use of queryable and returnable properties to ensure that valid responses are returned. Pass if validation succeeds. Fail otherwise.

Test type: Basic

A.2.5 Test case for retrieval of an OWL file

m) Test purpose(s): The catalogue must be able to retrieve an OWL file in its entirety using GetRecords.

n) Test method: For any valid OWL file, validate that the catalogue can access and return the OWL file in a valid format.

Test type: Basic

A.2.6 Test case for reasoning

o) Test purpose(s): The reasoning operations must return correct and valid responses.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

p) Test method: For any valid OWL file, validate that the reasoning operations work correctly.

Test type: Basic

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex B
(informative)

Example Use Cases

B.1 Scenario

B.1.1 Knowledge Discovery Scenario

Jane is a marine scientist and wants to buy equipment to study a particular phenomenon, about which she has little knowledge. She wants to find ALL the instruments that are capable of studying that phenomenon and knowledge about how to use those instruments and of how that phenomenon has been studied in the past. She is also interested in gathering journal articles and spatial datasets of past research on this phenomenon within her specific study area.

B.1.2 Knowledge Use Scenario

Jane has discovered that there has been a lot of research on this phenomenon in the past, but that a new measurement instrument has been developed that measures this phenomenon to a much greater accuracy and which has not been used in her study area before. She wants to compare research she will undertake with this new instrument to all the earlier research undertaken by various research groups. She needs to collect the data of this past research, and in some cases reformat the data, and do some statistical analysis that compares the past results with her data.

B.2 Abstract Use Cases

The following abstract use cases define a generic use case that any measurement instrument or scientific resource can be plugged into.

B.2.1 Knowledge Discovery A

Find measurement instruments that measure specific qualities of a phenomenon (e.g. find instruments that measure sediment concentration of the water column, then identify what other qualities these instruments measure).

The user wants to know what instruments can measure a given environmental property or quality, for example salinity, which can be calculated from instruments measuring conductivity (e.g. to obtain an instrument to perform the measurement, or to analyse those instruments for some other purpose). Associated information about how to use the instrument and past research regarding the instrument itself or the use of the instrument will also be needed.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Use Case Description	
Name	Knowledge Discovery A
Priority	High
Description	User searches for knowledge in a scientific knowledge infrastructure
Pre-condition	The knowledge infrastructure is available with scientific resources. Client is available to user.
Flow of Events – Basic Path	
Step 1.	The user is presented with an ontology of concepts in the domain and selects the concepts of interest.
Step 2.	The application formulates a request based on this Application Profile using a GetRecords request with the select ontology concepts.
Step 3.	The application receives the response from the OWL-CSW interface and displays it to the user in the form of a list of resources.
Step 4.	The user browses resources. Resources available include journal articles, spatial datasets associated with journal articles, measurement devices or the real-world properties measured, web services that serve data or make related functionality available, web pages of the producers of the instrument etc. The user may also select items that are associated by an ontology concept.
Post-condition	The user identifies a set of resources that directly answers the query

B.2.2 Knowledge Discovery B

For region Y, find datasets that represent a certain quality, and find associated instruments (e.g. find sediment concentration datasets, then find instrument that have produced these datasets so that they can be compared against other instruments that might measure differently).

Spatial datasets typically do not include a reference to the type of instrument used to measure it, and yet this knowledge is valuable as it may be used to better evaluate or process the data. For example, the user may want to know the accuracy and precision of the instrument's measurements, or the appropriateness of the calibration status that has been provided.

The link between the spatial dataset and the instrument is provided in the ontology that specifies the parameters that the instrument produces.

Use Case Description	
Name	Knowledge Discovery B
Priority	High
Description	The user searches for knowledge in a scientific knowledge infrastructure

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Use Case Description	
Pre-condition	The knowledge infrastructure is available with scientific resources, including spatially defined resources. Client is available to user.
Flow of Events – Basic Path	
Step 1.	The user selects a quality from the ontology and a location of interest.
Step 2.	The application formulates a request based on this Application Profile using a GetRecords request with the select ontology concepts. The user may also choose to search for semantically related terms (for example, by subsumption).
Step 3.	The application receives the response from the OWL-CSW interface and displays it to the user in the form of a list of resources.
Step 4.	The user browses resources. Resources available include journal articles, spatial datasets associated with journal articles, measurement devices or the real-world properties measured, web services that serve data or make related functionality available, web pages of the producers of the instrument etc. The user may also select items that are associated by an ontology concept.
Post-condition	The user identifies a set of resources that directly answers the query

B.2.3 Knowledge use

For location Y, find data measuring phenomenon X in past research so that it can be compared to the data collected with Instrument A (e.g. in the Firth of Forth, find data measuring salinity and compare to data collected with a new instrument, using two data sets that are in a different format and require a data re-formatting service).

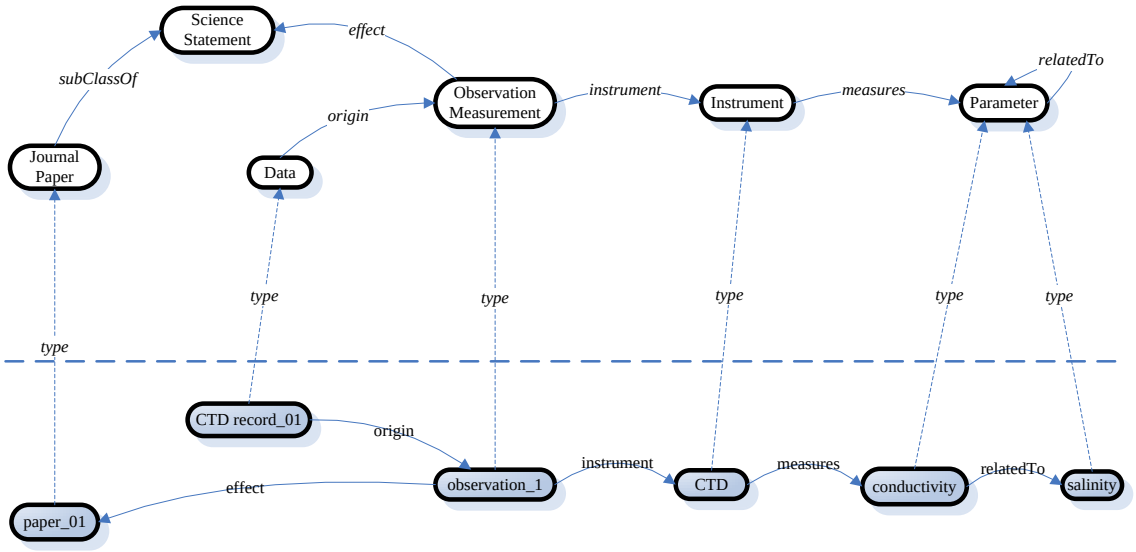
Use Case Description	
Name	Knowledge Use
Priority	Medium
Description	The agent searches for data in a knowledge infrastructure, recognises the disparity between theories used to measure the data sets and represents these theories
Pre-condition	The knowledge infrastructure is available with scientific resources, including spatially defined resources. Process of analysis is expressed in an ontology. Client to interpret and implement collection and analysis of datasets.
Flow of Events – Basic Path	

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Use Case Description	
Step 1.	The user selects phenomena X from the ontology and a location of interest.
Step 2.	The application formulates a request based on this Application Profile using a GetRecords request with the select ontology concepts. The user may also choose to search for semantically related terms (for example, by subsumption).
Step 3.	The application receives the response from the OWL-CSW interface and displays it to the user in the form of a list of resources.
Step 4.	Relevant datasets are shown to user, including knowledge of the instruments that measured them and the theories used. The user may select the theories ot see other resources that are related by theory.
Step 5.	Output data is provided to the user
Post-condition	The user has a reference to a dataset of interest.

B.3 Use Case Example

According to the description of the abstract use case, knowledge discovery involves information about instruments, environmental properties, and scientific resources such as journal articles and data sets. This section provides an example of an infrastructure that organises these resources for knowledge discovery and use. The resources involved in the use case and their relations are represented in Figure B.1.



Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Figure B.1. The Relationships between the Resources involved in the Use Case

The upper part describes the classes of the resources and the relationships among them. Instruments are used in observations to measure parameters, and the records from the measurements are stored in data sets. Scientific discoveries from observations are presented in journal articles, which can be expressed as a kind of science. The bottom part gives an example of instances.

B.3.1 Knowledge Discovery A

The knowledge discovery step applies a query that identifies instruments that are related to a given environmental property, as well as resources of research that are associated with the selected instrument. According to the description given in Figure B.1, an example ontology is created to organise these resources, as shown in Figure B.2.

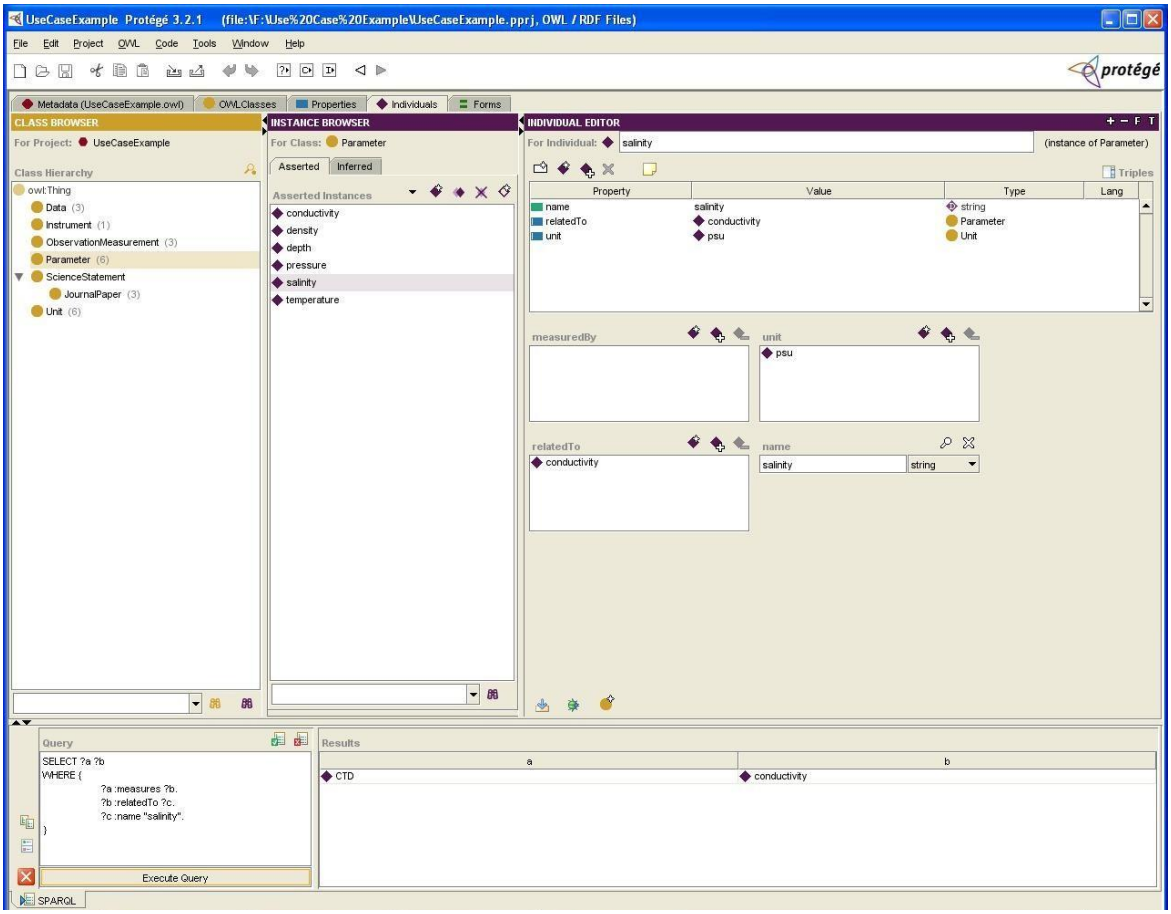


Figure B.2. A Query of Instruments for Measuring a given Parameter

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

A SPARQL query can be used with the OWL-CSW interface to perform the retrieval of instruments for measurement of the given parameter (Figure B.2). All the resources that are associated directly or indirectly with the selected instrument can be identified according to their relationships as described in Figure B.1 (Figure B.3).

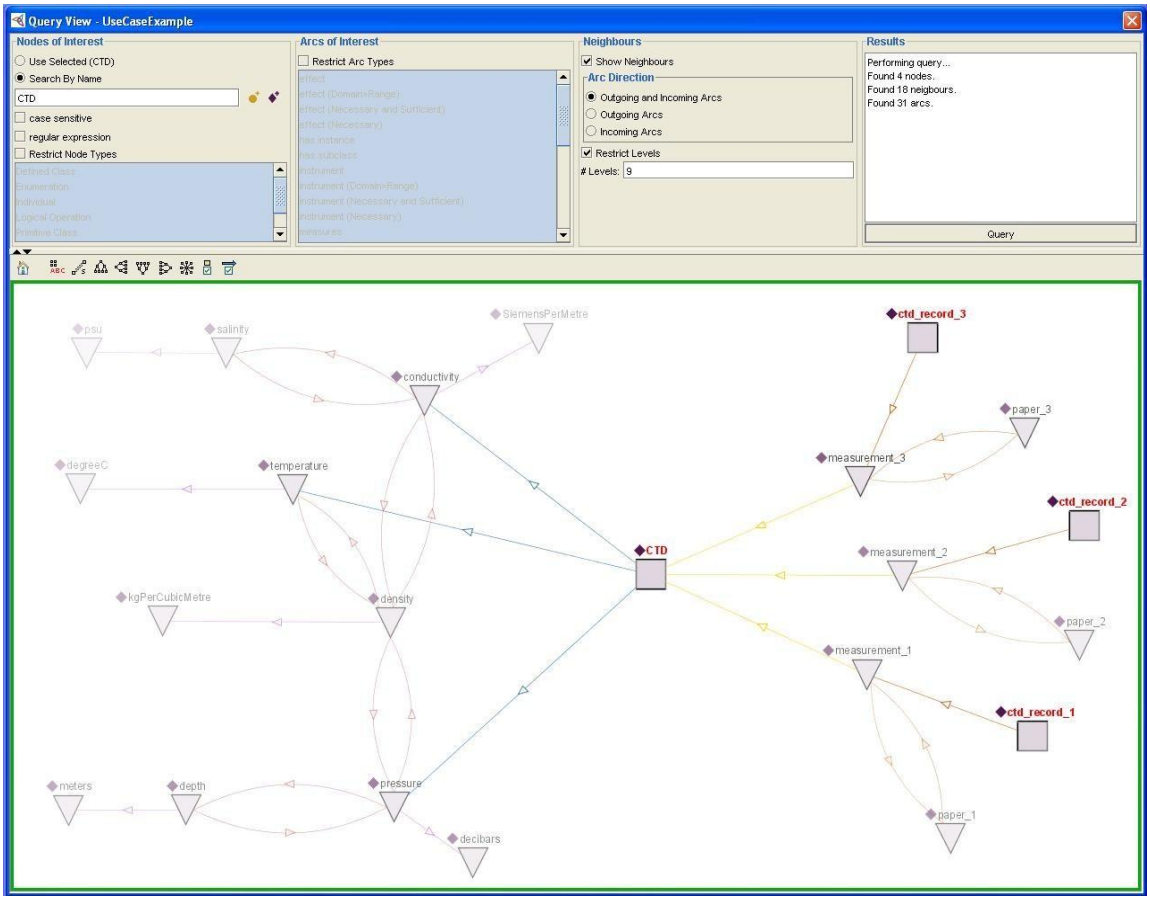


Figure B.3. A Query of Resources that are Related to the Selected Instrument

B.3.2 Knowledge Discovery B

The results of measurements are stored in data sets, which do not include a reference to the type of instrument that produces the records. This information can be stored in ontologies to support the information retrieval according to the reference between instruments and data. Since instruments are indirectly related to the data they produce through measurements, the data sets can be found in the resources related to the instrument (Figure B.3). A more accurate query can be performed to retrieve the specific type of resource that associates with the instrument through a SPARQL query (Figure B.4).

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

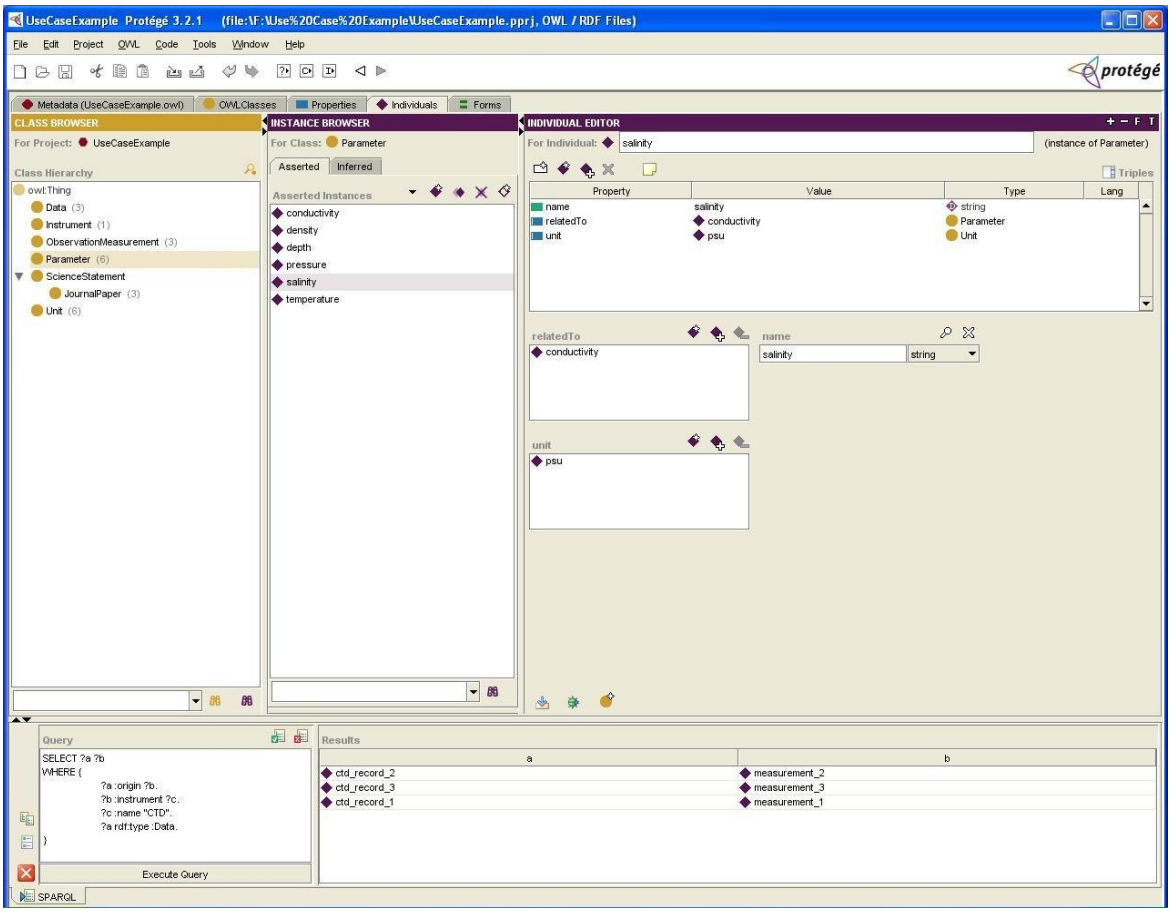


Figure B.4. Retrieval of Data Sets that are Produced from CTD Devices

B.3.3 Knowledge Use

In using knowledge expressed in an ontology, the user analyses a phenomenon by comparing the data collected by an instrument and the data from past research. Assuming that that Web services are available for searching for data by parameter and location, and for comparing two sets of data, the analysis process can be modelled in an ontology as presented in Figure B.5.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

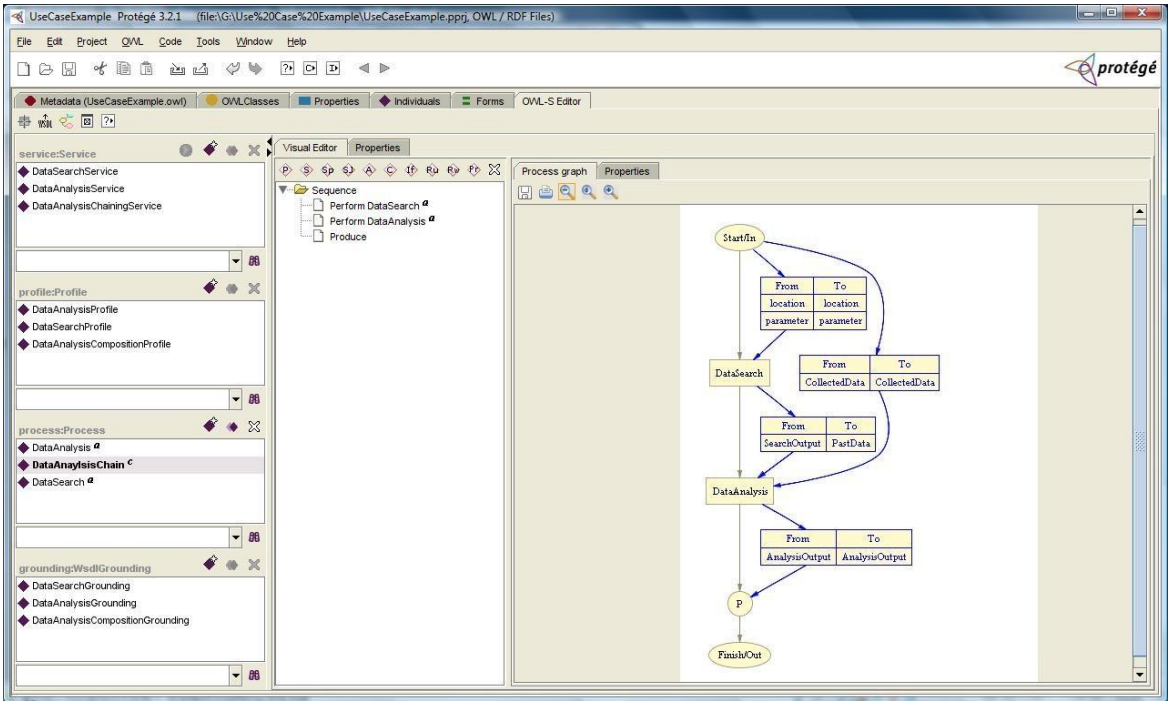


Figure B.5. Service Composition for Knowledge Use

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.

Bibliography

[1] IBM, OBJECT MANAGEMENT GROUP and SANDPIPER SOFTWARE (2008) *Ontology Definition Metamodel* . OMG Adopted Specification Document Number ptc/2008/09/07. http://www.omg.org/docs/ptc/08-09-07.pdf

[2] BAADER, F., CALVANESE, D., MCGUINNESS, D., NARDI, D. and PATEL-SCHNEIDER, P. (2003). *The Description Logic Handbook*. Cambridge University Press: Cambridge.

[3] DOGAC, A., KABAK, Y., LALECI, G.B., MATTOCKS, C., NAJMI, F. and POLLOCK, J., 2005, Enhancing ebXML Registries to make them OWL Aware. *Distributed and Parallel Databases,* **18**, 9-36.

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.