Open GIS Consortium Inc.

Date: 2003-01-17

Reference number of this OpenGIS© IP initiative document: **OGC 03-026**

Supersedes document: 02-055r2

Version: 0.3

Category: OpenGIS© Interoperability Program Report

Editor: Joshua Lieberman

OWS 1.2 Service Information Model

Warning

This document is not an OGC Standard or Specification. This document presents a discussion of technology issues considered in an Interoperability Initiative of the OGC Interoperability Program. The content of this document is presented to create discussion in the geospatial information industry on this topic; the content of this document is not to be considered an adopted specification of any kind. This document does not represent the official position of the OGC nor of the OGC Technical Committee. It is subject to change without notice and may not be referred to as an OGC Standard or Specification. However, the discussions in this document could very well lead to the definition of an OGC Implementation Specification.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OpenGIS® Interoperability Program Report
Document stage: Final
Document language: English

Contents

i. Preface

This document is an Interoperability Program Report from the OGC Open Web Services 1.2 Testbed. It specifies and discusses a common information model for OGC Web Services, also known variously or in part as service capabilities or service metadata. Included here is both a presentation of the present model itself and a discussion of how present and future instances of service information can be developed from it.

ii. Submitting organizations

The following organizations submitted this document to the Open GIS Consortium Inc.

CubeWerx, Inc.
200 Rue Montcalm, Suite R-13
Hull, QC J8Y 3B5
Canada

Galdos Systems, Inc.
200-1155 West Pender Street,
Vancouver, BC V6E 2P4
Canada

Polexis Inc.
2815 Camino del Rio So.
San Diego, CA 92108
U.S.A

Syncline Inc.
373 Washington Street
Boston, MA 02108
U.S.A.

iii. Document Contributor Contact Points

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

CONTACT	COMPANY	ADDRESS	PHONE/FAX	EMAIL
Joshua Lieberman	Syncline Inc.	373 Washington Street, Boston, MA 02108	Tel: 617-603-2209 Fax: 617-986-1001	jlieberman@syncline.com
Saul Farber	Syncline Inc.	373 Washington Street, Boston, MA 02108	Tel: 617-603-2248 Fax: 617-986-1001	saul@syncline.com
Peter Vretanos	Cubewerx	200 Rue Montcalm, Suite R-13 Hull, Quebec J8Y 3B5 CANADA	+1 416 701 1985	pvretano@cubewerx.com
Richard Martell	Galdos	200-1155 West Pender Street, Vancouver, BC V6E 2P4 Canada		rmartell@galdosinc.com
Jeff Lansing	Polexis Inc.	2815 Camino del Rio So., San Diego, CA 92108	+1 619 543 7225	jeff@polexis.com
Jeff de la Beaujardiere	NASA	NASA Goddard Space Flight Center Code 933 Greenbelt MD 20771 USA	+1 301 286 1569	delabeau@iniki.gsfc.nasa.gov
Stephane Fellah	PCI Geomatics	490 St. Joseph Blvd. Hull, Quebec J8Y 3Y7	+1 (819) 770-0022 ext 223	fellah@pcigeomatics.com
Jérôme Sonnet	Ionic Software s.a.			jerome.sonnet@ionicsoft.com
John Davidson	Image Matters, LLC	105 S. King St. Leesburg, VA 20175	+1 703 669-5510	johnd@imagemattersllc.com

iv. Revision history

Date	Release	Author	Paragraph modified	Description
2002-089-14	0.1	Lieberman		First draft of this document for OWS 1.2
2003-01-18	0.2	Lieberman		Final scrub

v. Changes to the OpenGIS® Abstract Specification

The OpenGIS© Abstract Specification does not require changes at present to accommodate the technical contents of this document. Attention is drawn, however, to the possibility that future revisions of this document may require revisions to Topic 12 and other AS topics.

vi. Future Work

Improvements in this document are desirable to reflect the experience of those implementing this service information model in their own service types and instances.

.

Foreword

Attention is drawn to the possibility that some of the elements of this part of OGC 02-055r1 may be the subject of patent rights. The Open GIS Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

This OWS 1.2 edition cancels and replaces the first edition of a common capabilities model document (OGC 02-020), OGC CA Common Capabilities Model R7 Part I: Tour and Roadmap

OGC 02-055r1 consists of the following parts, under the general title OWS 1.2 *Service Information Model*:

- *Part 1: Service Model - Information viewpoint*
- *Part 2: Informative description of the model*
- *Part 3: How-to Guide*
- *Part 3: UML form of the model*
- *Part 4: Areas for Future Work*
- *Annex A: XML Schema for Basic SIM*
- *Annex B: Capabilities document examples*

Introduction

OpenGIS Consortium (OGC) web-based geospatial services are designed to provide geographic information over the Internet. As such, they share many capabilities and characteristics of more generally defined Web Services, but also have features unique to geospatial needs. In particular, OGC Web Services are by definition self-describing and support a GetCapabilities operation which details supported operations and content. The response returned from a service's GetCapabilities operation is intended to supply all the information required for a client to make use of ("consume") that service. It can be considered as a complete service information model, expressed in UML, XML Schema, or as XML sample instances.

Previous versions of the GetCapabilities information model fulfilled this basic role, but were limited in several respects:

- Monolithic (i.e. could only be requested as one complete document),
- Unique from one type of service to another, particularly in terms of service invocation description,
- Incompatible in detail and not related to registry information models on which OGC and other catalog services are based,
- Not aligned with wider service description and metadata standards, and
- Information model not defined with XML Schema or UML.
- No support for service and data types.
- Little or no support for service chains.

The latter two limitations were addressed by the R4 capabilities information model described in OGC Document 01-084. Further work in OWS1.1 on the R7 model (OGC Document 02-022) and now in OWS 1.2 on the common service information model (SIM) have refined and extended this model to address the other limitations listed above.

The present document covers SIM for OGC Web Services. It describes this model on the three necessary specification levels:

- UML conceptual object model,
- Schema specification model (expressed as XML Schema),
- XML illustrative instance documents with detailed how-to annotations.

Elements of the SIM are drawn to the extent possible from existing specifications for service and content metadata (e.g. ISO 19115, ISO 19115), as well as encodings such as GML and SLD.

This document is related to the Architecture Requirements IPR (OGC Document 02-057) and Architecture IPR (OGC Document 02-056) in that it presents the RM-ODP Information Viewpoint of Service Architecture outlined in the latter document and drawn from Topic 12 of the OGC Abstract Specification. As such, it should be considered in the context of the underlying service architecture to which it adheres, rather than as an entirely standalone document.

OWS Service Information Model

1 Scope

This OpenGIS® Interoperability Program Report (IPR) is a partial description of OGC Web Services. It is a specification and description of a common information model for service capabilities description. This IPR provides UML, XML Schema, and XML vocabulary clauses for the information model. Other clauses describe model extensions for specific service and content types.

2 Conformance

Not required for an IP IPR, DIPR, or Discussion Paper.

3 Normative references

The following normative documents contain provisions that, 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.

CGI, *The Common Gateway Interface,* National Center for Supercomputing Applications, <http://hoohoo.ncsa.uiuc.edu/cgi/>

EPSG, *European Petroleum Survey Group Geodesy Parameters*, Lott, R., Ravanas, B., Cain, J., Girbig, J.-P., and Nicolai, R., eds., <http://www.epsg.org/>

FGDC-STD-001-1988, *Content Standard for Digital Geospatial Metadata (version 2),* US Federal Geographic Data Committee, <http://www.fgdc.org/metadata/contstan.html>

IETF RFC 2045 (November 1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Freed, N. and Borenstein N., eds., <http://www.ietf.org/rfc/rfc2045.txt>

IETF RFC 2119 (March 1997), *Key words for use in RFCs to Indicate Requirement Levels*, Bradner, S., ed., <ftp://ftp.isi.edu/in-notes/rfc2119.txt>.

IETF RFC 2616 (June 1999), *Hypertext Transfer Protocol – HTTP/1.1*, Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., eds., <http://www.ietf.org/rfc/rfc2616.txt>

IETF RFC 2396 (August 1998), *Uniform Resource Identifiers (URI): Generic Syntax*, Berners-Lee, T., Fielding, N., and Masinter, L., eds., <http://www.ietf.org/rfc/rfc2396.txt>

ISO 8601:1988(E), *Data elements and interchange formats - Information interchange - Representation of dates and times*.

ISO 19115, *Geographic information — Metadata*

OGC AS 12 (January 2002), *The OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture (Version 4.3)*, Percival, G. (ed.), <http://www.opengis.org/techno/specs.htm>

UCUM, *Unified Code for Units of Measure,* Schadow, G. and McDonald, C. J. (eds.), <http://aurora.rg.iupui.edu/~schadow/units/UCUM/>

XML 1.0 (October 2000), *Extensible Markup Language (XML) 1.0* (*2nd edition)*, World Wide Web Consortium Recommendation, Bray, T., Paoli, J., Sperberg-McQueen, C.M., and Maler, E., eds., <http://www.w3.org/TR/2000/REC-xml>

4 Terms and definitions

operation
specification of a transformation or query that an object may be called to execute [OGC AS 12]

interface
named set of **operations** that characterize the behavior of an entity [OGC AS 12]

service
distinct part of the functionality that is provided by an entity through **interfaces** [OGC AS 12]

service instance
server
actual implementation of a **service** or conceptual role as recipient of an operation request.

client
software component that can invoke an **operation** from a **server** or conceptual role as originator of an operation request.

request
invocation of a **server operation** by a **client**

binding
specific syntax and parameter values used by a **client** to invoke a specific **server operation**

response
result of an **operation** returned from a **server** to a **client**

map
pictorial representation or portrayal of geographic data

spatial reference system (SRS or CRS)
a projected or geographic coordinate reference system

service capabilities
service-level metadata describing the **types**, **operations**, **content,** and **bindings** available at a **service instance**. Organization, classification, and presentation of those entities may also be conveyed by the capabilities information.

capabilities schema
XML schema which prescribes and constrains the syntax and vocabulary for the expression of service capabilities in XML.

capabilities XML
specific instance of service-level metadata describing a **service instance**.

5 Conventions

5.1 Normative Verbs

In the sections labeled as normative, the key words "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in [IETF RFC 2119].

5.2 Abbreviated Terms

DCP	Distributed Computing Platform
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
MIME	Multipurpose Internet Mail Extensions
OGC	Open GIS Consortium
OWS	OGC Web Service
URL	Uniform Resource Locator
WMS	Web Map Service
XML	Extensible Markup Language
API	Application Program Interface
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
DCE	Distributed Computing Environment
DCP	Distributed Computing Platform
DCOM	Distributed Component Object Model
IDL	Interface Definition Language
ISO	International Organization for Standardization
OGC	Open GIS Consortium
UML	Unified Modeling Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
WSDL	Web Services Definition Language

5.3 UML Notation

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

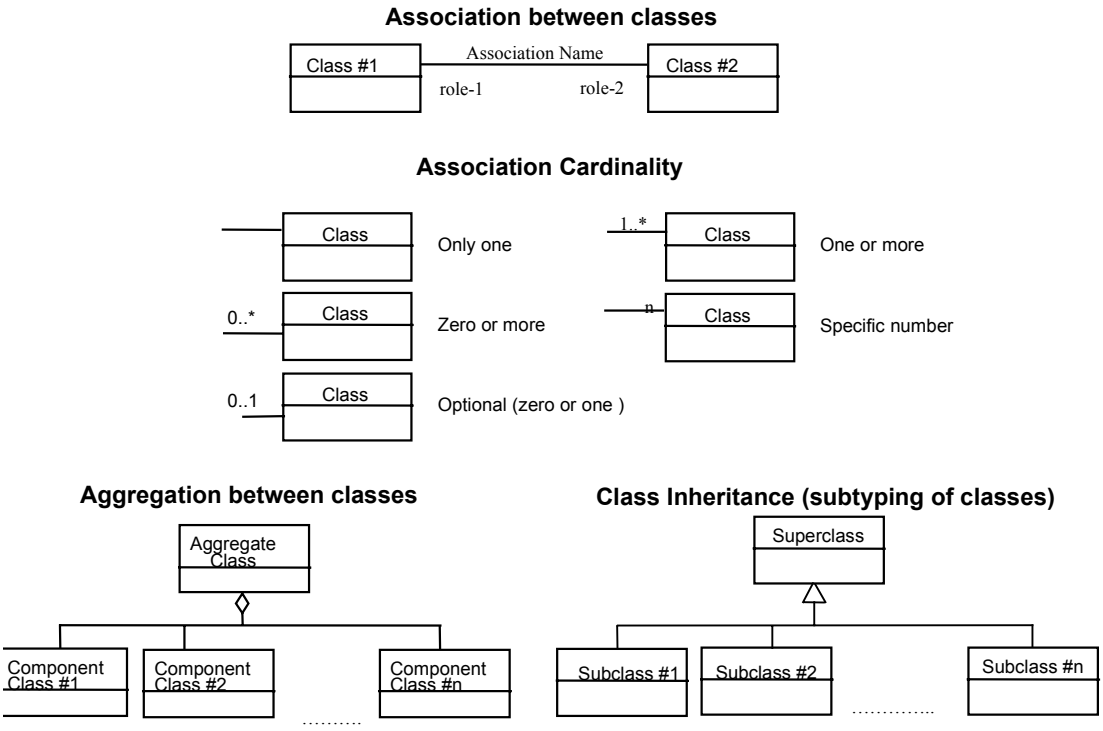

Figure 1 — UML notation

In this diagram, the following three stereotypes of UML classes are used:

a) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

b) <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.

c) <<CodeList>> is a flexible enumeration that uses string values for expressing a list of potential values.

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

a) CharacterString – A sequence of characters

b) Integer – An integer number

c) Double – A double precision floating point number

d) Float – A single precision floating point number

5.4 XML, XML Schema, and XML Namespaces

XML Schema 1.0 is used in this document to define the syntax and vocabulary of XML documents. It is not assumed that XML documents will be fully validated against their respective schemas for normal service invocation. It is assumed, however, that XML parsing and validation will follow normal XML syntax rules, for example regarding XML Namespaces and their definition within XML documents. Specific namespace URI's are suggested and their standardization is encouraged. Specific namespace prefixes area used, but their standardization is not suggested in order to maintain maximum flexibility in aggregating XML from different namespaces.

5.5 Xpath Notation

To refer to specific portions of an XML document, this specification uses a subset of the syntax defined by [W3C XPath]. In particular:

`A/B` refers to all elements <B> which are an immediate child of any element <A> in the document.

`/A` refers to the root element <A> of the document.

`C/@D` refers to the value of the attribute D of any element <C> in the document.

6 Service Model - Information Viewpoint

In the context of geospatial services, the Service Information Model (SIM) represents one of several viewpoints into a comprehensive service architectural model. The information viewpoint is used to describe the information required by an Open Distributed Processing (ODP) application through the use of information models and schemas, which describe the state and structure of application objects. The information viewpoint is concerned with the kinds of information handled by the system and the constraints on their use and interpretation. Of the five viewpoints described in [n], the information viewpoint is the second least implementation-dependent after the enterprise view, which concerns itself with the environment and problem context of an ODP application. Since the present report focuses on just one viewpoint, the reader is referred to the companion report on Service Architecture, OGC 02-056, for definition and discussion of the service components with which the SIM concerns itself.

The following sections expand on the concept of the information viewpoint in a manner inspired by but not necessarily tied to [1]. While the concepts are general in nature, the specific scope of this document is service information / metadata. Non-service-oriented metadata such as content or registry metadata are covered more thoroughly in the appropriate specific documents.

6.1 Viewpoint presentation

The essential payload of an ODP application is information, much as the essential purpose of an irrigation system is to move water. From this viewpoint, the nature of the

information pathways and controls is irrelevant as long as the right information is discovered, transformed, presented, and delivered to the right destination. There may be, however, any number of levels of meta-information which describe what that information is and could be, as well as the operations to be performed upon it.

In the context of an ODP application, information falls into a number of categories or types, for example:

- Payload information
- Processing information
- Transport information
- Application state information
- Meta-information for any of the above.

Since this document concerns itself with information architecture, it mainly concerns itself with the last category of information, reserving the first four mainly for purposes of illustration.

6.2 Role of Information Models

The idealized ODP application would perform any operation on any information. In reality, applications need to specialize on particular classes of information in order to function effectively. The role of the information model is to define the general characteristics of information which can be processed by a given type of ODP application. This is termed an information model. There is no correct nor general model per se, but a good information model will strike an optimal balance between the processing efficiency of fixed structure and the flexibility to accommodate a wide spectrum of information characteristics, by identifying the information structures (e.g. objects – properties - types) which remain static across a particular information domain.

There are a wide variety of ways in which to define and document information models within the scope of OWS 1.2, which fall onto at least five different levels, in roughly descending order of generalization:

- Object Model (e.g. UML)
- Encoding Syntax (e.g. XML Schema)
- Encoding Style (e.g. parameterBindings, Schematron, tutorials)
- Data Dictionary (e.g. domain, metadata, etc)
- Sample Instance (e.g. XML)

Each of these levels of documentation serves an important purpose in providing a complete definition of an information model. There are two reasons for this. First, the mapping between one level and another is non-unique (in general), requiring a step-by-step connection from object model generality to message or "wire" format compatibility. Second, it is often the case that a given person is most comfortable approaching information models on one or two specific levels. Documention of a model on multiple levels will tend to make it accessible to the largest audience of users.

6.3 Definition and Role of Metadata

There is an essential but subtle distinction between information to be processed (or processing instructions) and meta-information which communicates the structure, syntax, and semantic content of that information. This is also termed metadata (sensu latu). There are no hard and fast rules for making the distinction between data and "data about data", although metadata tends to be useful on a more generalized level in the sense of

- cardinality (e.g. feature collections rather than features),
- typing (e.g. feature type definitions rather than feature instances, operation request schemas rather than requests themselves, service types rather than service instances), and/or
- temporality (e.g. a sequence of observations rather than specific point-in-time observations).

At some point a judgement is made that one piece of information is essential while another associated piece of meta-information or description serves mainly to make the first piece more useful or more meaningful. Service information consists of those latter pieces of information which make service interactions more useful or meaningful. The scope of this report is a model for meta-information of this type.

Just as there is no a priori distinction between data and metadata, there is also no a priori scope for metadata, particularly in regards to service offers of specific content. For example, there may be useful descriptive information for feature collections, for individual features, for individual nodes within features, as well as for higher-level entities such as "runways" which may be composed of two or more associated / linked features. This can result in specific defined relationships between content objects, between content and meta-content, and also between meta-content objects. Each service type or service instance may in practice have associated with it a unique model of content and content description, which might be described more usefully as a "content map".

The emphasis of this report is on those elements of service description which appear to be common to all geospatial services. The service information model must of necessity also be extensible to describe metadata which is characteristic of a specific service type or instance. While the content maps of those OGC services which have been defined and implemented to date have been of limited complexity and variability (e.g. WMS -> layers, WFS -> feature collections), OWS 1.2 activities in defining new service types has

made clear that explicit content maps are becoming a needed additional component of the SIM. Specific proposals for extending the SIM in this direction are developed in Clause 10 of this report.

7 Service Model – Other Viewpoints

While this report deals mainly with the information viewpoint of a service architecture, there are a few aspects of service information which belong here but are best expressed through other viewpoints such as technology or computation.

7.1 Role of Encoding Technology

Information models become accessible through service interfaces by means of data encodings. Encodings naturally begin to involve specific communications technologies, so there are usually a number of encoding alternatives for any conceptual information model (which may be expressed, for example, in UML). For the purposes of this document, the preferred encoding technologies are XML and XML Schema. This should not, however, exclude the possibility of alternative encodings where such encodings can also support interoperability of service interfaces. As service information can be considered metadata, the appropriate class of service interface for interacting with it is variously termed “registry” or “catalog”. These terms are freighted with particular meaning for particular audiences and information communities.

7.1 Computational Role of Registries

Just as a discussion of a service information model is incomplete without consideration of how it might be encoded, so also specific service interfaces for interacting with service information are an appropriate topic of consideration.

7.2 BasicRegistry Interface: GetCapabilities operation

7.3 GetCapabilities syntax

8 Service Information Model (Informative)

This clause provides an informative discussion of the SIM design criteria and resulting components.

8.1 Key SIM Design Concepts

This clause introduces and discusses several of the key design criteria which went into formulating the present SIM.

8.1.1 Object model

It is important both for useful definition of service information and for adherence to OGC policy, that the SIM be expressible as an object model in UML. This is equivalent to stating that there must be a platform-independent, common information model for service information.

Another aspect of model development is the reuse of standard model elements. In particular, reuse and extension of registry models such as ogcRIM as well as metadata models such as ISO19115, ISO19119, and encoding models such as GML.

8.1.2 Capabilities as a Container

With more voluminous service content and more complex service configurations, it is desirable to shift the focus of service capabilities description from that of an all-encompassing metadata document to a modular container for the transport of specific metadata objects. While there is some loss of human readability with this approach in terms of briefer service capabilities, the improvements in machine readability and utility are felt to be worth the tradeoff. In particular, if one takes the view that exchange of capabilities is a basic registry function, then a generalized modular mechanism for service information exchange is required.

8.1.3 Strong typing

One proven technique for making information more useful, meaningful, and discoverable, is a typing framework. In previous service capabilities, service and content types have been informally communicated, but they should be distinct objects with distinct relationships to other capabilities objects. In the case of OGC services, a distinction is drawn between service types (descriptions of typical interface combinations) and content types (descriptions of content organization). The separation of content metadata such as ISO 19115 into type and instance descriptions is an interesting corollary of this design requirement which is beyond the scope of the present document.

8.1.4 Separation of content and presentation

One or more services may offer the same content (e.g. feature dataset) in more than one format or style. It is important in these cases to distinguish between description of the essential content itself, and description of the manner in which it is being presented in a particular service invocation. Separation of content and presentation is an important general principle, but particularly required for a properly normalized object model.

8.1.5 Combination of Prescription and Validation

An important limitation of syntax encodings such as XML Schema is that they are typically better for determining whether particular information (e.g. a service request) is correctly constructed (validation), than for guiding its construction in the first place (prescription). In other words, XML Schema can tell you where you went wrong, but not how to get it right in the first place. The SIM seeks (principally by way of the

parameterbinding) to provide metadata prescription as well as validation, but further "recipe" improvements are expected in the future.

8.1.6 Use of WSDL

As Web Services become important implementations of ODP, many useful standards for service information are being developed and refined. Web Services Definition Language has been identified as one of those useful standards, in spite of some number of present limitations with regard to OGC service descriptions.

8.1.7 Links to content metadata

As service information becomes more formally defined in SIM, it becomes more important to develop useful 2-way connections between information which focuses on content (e.g ISO 19115, FGDC) and service information which involves that content. OnlineURL and MetadataURL elements are a start in this direction, but further developments should be made to enable seamless transitions between content discovery and service discovery.

8.1.8 Extensibility

As discussed above, optimal SIM design will provide a core information model common to all OGC services, while also providing points at which the model can be extended for service or content-specific metadata.

8.2 SIM Components

The following clauses set forth the basic elements of the OWS 1.2 Service Information Model. Comprehensive normative and informative descriptions of these elements follow in later clauses and annexes.

8.2.1 The OGC_Capabilities element

This element is the root element of the SIM schema model. It is a container which holds any number of different types of RegistryEntries. It is useful for transporting RegistryEntries from one place to another (e.g. the GetCapabilities operation), for submitting/updating groups of RegistryEntries (e.g. the RegisterService operation) or for generally perusing groups of logically related RegistryEntries. It currently separates the (often large) lists of ContentInstances and ContentTypes into sub-lists (ContentInstanceList and ContentTypeList respectively), but this is purely for aesthetic value, adding no functional purpose or difference from the treatment of PresentationSchemes, ServiceTypes and ServiceOffers

8.2.2 Service Type

Service type definitions provide the semantic hierarchy by which individual service instances are interpretable. Even though services are defined by their operational interfaces, this doesn't tell the user exactly what response will be received for a given specific request. Implementation may be hidden, but the user must still be able to infer the behavior of a service. For OWS1.2, a WSDL interface document forms the bulk of service type definitions, although generous documentation nodes in the form of RDF or even text may be helpful.

There is not a complete correspondence between a service type in WSDL and a service specification. A service specification not only defines the interface syntax, but usually proposes required minimum enumerations of request parameters (e.g. output formats) to promote interoperability between service implementations. It is not really feasible to express a “required-but-extensible” parameter domain in either XML Schema or in WSDL. It therefore makes sense (with a nod to Rob Atkinson) to include optional parameterBinding elements in the ServiceType object as well to prescribe efficiently such interoperable parameter profiles. The service type profiles are then over-ridden by the parameterBindings within the ServiceOffer or ContentInstances.

A service may provide its own service type definition, but is most likely to provide no more than a reference to some general service type definition, most likely managed by a service type registry.

8.2.3 Service Instance

Service instance definitions are the essential instructions for exploiting a given service instance, and the essential entry object for a service instance registry. A WSDL implementation document is generally useful but insufficient for this purpose. WSDL specifications refer to bindings and service endpoints as defining a service instance, presuming that a service is really an exact implementation of some service type. In the OGC world, at least, this is complicated both by the restrictions / extensions imposed by a service instance on a service type (e.g. available coordinate system transformations), and by the parameters which must be specified to access specific service content (e.g. layer, style, feature type, BBOX).

All of this "recipe" information taken together is referred to as a ParameterBinding. The concept of a complete recipe is an important one, particularly for machine clients to be able to bind dynamically to diverse services. There are several possible paths to achieving it, however, most representing some redundancy or linkage of information between different capabilities components.

A service metadata component is generally represented as a subset of the information defined in ISO 19119, plus a key linking it to the service instance components with which it belongs. A requirement has been expressed for servicetype-specific metadata, so there is an additional section in the serviceInstance for this as well.

8.2.4 Content Type

The OGC Service Model appears to be rather unique among Web Services initiatives in emphasizing the importance of service content and content type distinct from and on par with services and operations themselves. This may be somewhat alarming for those initiatives, but OWS can nonetheless carry on and support interoperability between service offerings by means of strongly typing service content. The means of achieving this typing is through a hierarchy of content types, expressed with encoding schemas / profiles related to GML, SensorML, SLD, and so on. The most specialized content type definitions will qualify as data product descriptions which can support interchangeability or at least comparability of content between services (e.g. trading one service's USGS DRG 7.5' topo quad offering for another's).

8.2.5 Content Instance

This registry entry / capabilities object represents the core of the content section of previous service capabilities documents. It consists of the essential information to define each unique content offering, including links to the relevant service instance and content type, as well as a subset of ISO 19115 metadata. For many services, this is the only content component which would need to be provided. It might also serve as the basic registry entry object for a specialized data or sensor instance registry. In any case, it serves as the connection between the metadata for content and the services which offer that content.

There is a certain inevitable hierarchical quality to children of content instance objects, such as WMS layers that also have enumerated styles and time domains. The intent here, however, was to flatten the list of content offerings as much as possible to a list of uniquely identified content records for better manageability as registry entries. This approach grew out of the observation that most actual usage of the WMS layer hierarchy served the purpose of content "presentation" for such things as content grouping and client menus, where a particular layer might occur more than once, rather than being used for content definition. This presentation information has therefore been extracted to a separate Content Organization object.

8.2.6 PresentationScheme

A PresentationScheme is a registry entry which groups a set of hierarchically arranged classification nodes, each of which groups (classifies) one or more other registry entries. For the purposes of content presentation, a specialized node, the PresentationNode provides an optional way of organizing a fixed set of content instances (as Presentations) into an unlimited number of hierarchies. The specific feature of the Presentation and PresentationNode objects is that they include optional parameterBindings. The parameterBinding in the Presentation element links a specific invocation of a contentInstance into that PresentationNode (e.g. a specific style). The parameterBinding in the PresentationNode itself allows the content aggregated into that node to be invoked as a unit.

For a WMS Portrayal Service, PresentationNodes might enable map composition groupings, or categorization by organization. For a Gazetteer service, they might define alternate vocabularies of placenames based on particular themes or extents. For a WFS, a Presentation might represent a particular subset or view of a feature collection.

8.2.7 ContentMap

A PresentationScheme is a registry entry which groups a set of hierarchically arranged classification nodes, each of which groups (classifies) one or more other registry entries. For the purposes of content presentation, a specialized node, the PresentationNode provides an optional way of organizing a fixed set of content instances (as Presentations) into an unlimited number of hierarchies. The specific feature of the Presentation and PresentationNode objects is that they include optional parameterBindings. The parameterBinding in the Presentation element links a specific invocation of a contentInstance into that PresentationNode (e.g. a specific style). The parameterBinding in the PresentationNode itself allows the content aggregated into that node to be invoked as a unit.

For a WMS Portrayal Service, PresentationNodes might enable map composition groupings, or categorization by organization. For a Gazetteer service, they might define alternate vocabularies of placenames based on particular themes or extents. For a WFS, a Presentation might represent a particular subset or view of a feature collection.

8.3 Relationship between SIM and ogcRIM

There are at least three different ways in which the relationship between the OGC Registry Information Model (ogcRIM) and the OGC Service Information Model can be viewed.

8.3.1 Abstraction - Implementation

In this view, the RIM provides the base abstract objects which are extended to form specialized information objects which make up the SIM.

8.3.2 Metamodel - Model

In this view, the RIM provides the abstract building blocks which are combined, rather than extended to provide an alternative expression of the SIM in which, for example, additional object properties are expressed as slots and object relationships are expressed as associations, and so on.

8.3.3 Container - Content

In this view, the RIM is implemented in a general-purpose registry which ingests SIM-based capabilities documents and performs some general or specific mapping from SIM objects onto RIM objects. This mapping could range from treating the entire capabilities

document as arbitrary XML to mapping SIM objects properties individually onto those of RIM objects following the metamodel expression of SIM in 7.3.2.

8.4 Key RIM Components

EbRIM introduces two key concepts upon which the OWS-1.2 capabilities document is built, and one optional concept which can help organize content in an OWS-1.2 service capabilities document.

a) RegistryObject (key concept 1)

   RegistryObject is the UML class which represents any object which can be stored in a registry. Basically, if you want to store something (a piece of metadata, a bit of content, an association between the two) in a registry, the XML that you store must be derived from RegistryObject.

b) RegistryEntry (key concept 2)

   RegistryEntry is the UML class which represents any “first-class” metadata object which can be managed by a registry. Things like service-descriptions, content-descriptions and the like are derived from the RegistryEntry type.

c) ClassificationScheme (optional concept)

   The ClassificationScheme is a means of creating arbitrary hierarchical structure out of otherwise unassociated RegistryObjects. A ClassificationScheme (generally called a "classification") is a nested tree of "ClassifiedObjects," or links to RegistryObjects. This allows one to, for example, create a tree which represents the relationship of the hierarchy of types of "GML roads."

8.5 Management of SIM content in a registry

TBD

9 How-to Guide to Service Information and Capabilities

9.1 Section Overview

The OWS-1.2 SIM defines five sections which are specializations of the RegistryEntry class. The section names are the following:

a) ServiceType

b) ServiceOffer

c) ContentType

d) ContentInstance

e) PresentationScheme

These sections each extend the RegistryEntry base class in order to add functionality for storing and describing particular types of information. Together, these 5 types strive to provide a space for all information necessary to fully describe the capabilities of a distributed geospatial web service. The following is an examination in greater depth of each type of RegistryEntry. All together, an OWS-1.2 capabilities document is made up of some or all of these sections, grouped under the OGCCapabilities container element. A minimal capabilities document is completely empty (a null document, I guess), while a capabilities document representing a cascading or multi-service platform might link many ServiceTypes and ServiceInstances together into a finely woven ClassificationScheme.

9.1.1 ServiceType

The ServiceType RegistryEntry represents a place for metadata about a particular *type* of service. Some information that could appear in a ServiceType declaration might be:

- Location of the authoritative WMS specification
- WSDL which defines portions of the WMS interface which are common to *every* WMS.
- ISO-19119 metadata which is common to *every* WMS (i.e. common to the WMS specification)
- ParameterBinding(s) which define normative parameter profiles (e.g. every WFS must support GML as an output type).

It is worth noting that most registries won't need to store or extrude any ServiceType declarations, as most "official" ServiceType definitions will be centrally hosted and only used by reference to their URLs or URIs. So while developing a ServiceType for your service is necessary, it will probably not be that interesting to registry implementors, as there will be relatively few total ServiceType declarations.

9.1.2 ServiceOffer

The ServiceOffer RegistryEntry represents the specific information needed to link an abstract ServiceType declaration to a concrete service implementation. The ServiceOffer RegistryEntry might contain the following sections:

- WSDL describing the endpoints of the WMS interface, linked to the WSDL (in the servicetype section) which defines the common WMS portions.
- ISO 19115 and 19119 metadata about this particular service.

- ParameterBindings which declare "global" parameter defaults or values for this service.

9.1.3 ContentType

The ContentType RegistryEntry declares any information which is relevant to all instances of a particular type of data. This generally consists of a schema of some sort (XMLSchema and GML Application Schemas are supported) which describes the layout and structure of the data in question. The ContentType might also store metadata about the type, such as the authoritative contact for its definition, or its origin.

9.1.4 ContentInstance

The ContentInstance RegistryEntry declares a specific instance of a piece of content. This has different meanings in different contexts, so here are a few examples:

- A layer in a WMS
- A FeatureCollection in a WFS
- A coveragelayer in a WCS

Some metadata is stored for each piece of content, including some non-spatial metadata (authority, contact info, etc) and (optionally) some spatial metadata (SRS, bboxes, etc). However, each service needs to store different metadata about its content. A WMS, for example, must expose "Style" information for each layer, and a WCS must expose "Format" information for each coverage. This is accommodated by a "ServiceSpecificMetadata" section which can be extended by each service to fit the necessary metadata.

9.1.5 PresentationScheme (ClassificationScheme)

A PresentationScheme is a specific implementation of the ClassificationScheme model. It allows one to "group" together any bundle of objects into a "tree" or taxonomy. This is useful for expressing hierarchical relationships across sets of content or service**s.**

9.2 Step By Step Instructions for Creating an OWS1.2 Capabilities Document

This section details the step by step instructions for creating a new OWS-1.2 capabilities document based on the "skeleton" capabilities document provided in Appendix B.

9.2.1 Prerequisite Decisions

Before creating an OWS-1.2 capabilities document, you must decide on some identifiers, or identifier patterns.

9.2.1.1 ServiceTypeId

This is a unique identifier for your ServiceType. If there already exists a ServiceType definition for your service, *use the id of the already existing ServiceType definition.* This definition will at some point be globally hosted.
e.g. http://www.opengis.net/servicetypes/WMS/1.1.13

9.2.1.2 ServiceInstanceId

This is the unique identifier for your ServiceInstance. This should be globally unique, so try to incorporate a guaranteed unique identifier, such as your fully qualified domain name or company/personal identifier.
e.g. http://syncline.com/WMS/1.1.13/instance-name

9.2.1.3 ContentTypeId

If you are declaring *new* ContentTypes, you must decide on a unique identifier for each one. Rather than define new ContentTypes for each service's content, try to re-use already existing ContentTypes for content that you serve. I.e. there should be only one ContentTypeId for *all* WMS-Layers, as they are the same type of content.
e.g. http://www.opengis.net/contenttypes/wms-layer

9.2.1.4 ContentInstanceId

You must declare a unique id for each ContentInstance that you declare. The id need not be globally unique, only unique within your service.
e.g. layer-id-1

9.2.2 ServiceType

9.2.2.1 Fill in Attributes

- For the serviceTypeId attribute, fill in your chosen serviceTypeId.
- For the title attribute, fill in a suitable title for this ServiceType.

9.2.2.2 Insert Type-WSDL

If WSDL exists for your service (it should eventually, but might not yet), place the <types>, <message>, and <port> sections into the <typeWSDL> section.

9.2.3 ServiceOffer

9.2.3.1 Fill in Attributes

- For the title attribute, fill in a suitable title for this ServiceType.
- For the serviceInstanceId, fill in the chosen serviceInstanceId value.
- For the serviceTypeId, fill in the serviceTypeId for this service.

9.2.3.2 Insert Instance-WSDL

If WSDL exists for your service (it should eventually, but might not yet), place the <binding> and <service> sections into the <typeWSDL> section.

9.2.3.3 Fill in ISO-19119 metadata

Fill in the relevant ISO-19119 metadata for your service.

9.2.3.4 Fill in relevant ParameterBindings

>>>needs more work<<<

9.2.4 ContentType

9.2.4.1 Fill in Attributes

- For the title attribute, fill in a suitable title for this ContentType.
- For the contentTypeId, fill in the chosen value.

9.2.4.2 Fill in the Definition Element

A ContentType generally holds schema for a particular kind of content. Place that schema (in some sort of XML form) as a child of the "definition" element, and fill in the "markupLanguage" attribute appropriately. markupLanguage must be one of:

- "http://www.opengis.net/gml/1.0"
- "http://www.opengis.net/gml/2.0"
- "http://www.opengis.net/gml/3.0"
- "http://www.opengis.net/sld/0.7x"
- "http://www.opengis.net/sensorml"

- "##binary"
- "##other"

9.2.5 ContentInstance

9.2.5.1 Fill in Attributes

- For the title attribute, fill in a suitable title for this ContentInstance.
- For the contentTypeId, fill in the id value of the ContentType of which this ContentInstance is a member.
- For the serviceTypeId, fill in the id value of the ServiceType from which this ContentInstance is served.
- For the contentInstanceId, fill in the chosen value for this ContentInstance.

9.2.5.2 Fill in VendorSpecific Info

Fill in any Vendor Specific information in the "VendorSpecific" tag. This can optionally be validated against XMLSchema using the xsi:schemaLocation tag.

9.2.5.3 Fill in ContentInstanceMetadata Info

Fill in values for the ContentInstanceMetadata sections:

** enumerate sections here **

9.2.5.4 Fill in ContentInstanceGeospatialMetadata Info

Fill in values for the ContentInstanceGeospatialMetadata sections.

These sections include declaring a (single) nativeSRS of the content, and any number of relevant "extent" elements in particular coordinate systems.

9.2.5.5 Fill in ServiceSpecificMetadata Section

This section allows you to put in your own "service-specific" block of XML, which declares any extra service specific metadata that content for your service must reveal. This includes things like "Style" tags for a WMS, or "ReturnFormat" tags for a WFS (I made that one up).

You must design this section yourself! What data does your service need to reveal about each piece of content? Style? Specific location? Format? This information goes in this section.

9.2.5.6 Fill in relevant ParameterBindings

>>>needs more work<<<

8.2.5 PresentationScheme

This section is a classification scheme with specialized children which allows one to aggregate and classify content. It also allows one to assign a parameter binding which invokes a particular content aggregation (e.g. map composition) directly.

9.2.5.7 Create one or more PresentationScheme elements

Each taxonomy could represent organization of content along a particular set of principles, or for particular information communities.

** discuss attributes **

9.2.5.8 Add PresentationNode elements to each PresentationScheme

Note that the parent attribute allows a hierarchical organization of these nodes. Also fill in the parameterBinding if there is a binding to invoke this aggregation directly.

** discuss attributes **

9.2.5.9 Add Presentations

Create a presentation within each presentationnode for each contentinstance which is being linked to that node. Add a parameterbinding if a particular binding (e.g. Style or FeatureType or filter expression) is being referred to in the presentation. Fill in the order attribute to define the printing order of the contents in that node.

** discuss attributes **

10 UML for the Service Information Model

The following object diagrams present a visualization of the Service Information Model from both a conceptual and a physical or implementation viewpoint.

10.1 Conceptual Object Diagram

Figure 2 diagrams the inheritance and aggregation relationships between information objects in the SIM. The top five objects are all drawn from the Registry Information Model (ebRIM/ogcRIM), although only one (*ClassificationScheme*) occurs as-is in the SIM. The other SIM objects (S*erviceType, ServiceOffer, ContentType, ContentInstance*,

Presentation, PresentationNode) are specializations of RIM objects. While they are expressed in classical inheritance fashion, their relationship to RIM can also be expressed in metamodel fashion, where the properties added to the base RIM objects are accommodated as slots. For example, in the UML, *PresentationNode* is a subclass of *ClassificationNode* which adds a *parameterBinding*. In the metamodel view (SIM Schema in Appendix A), the *parameterbinding* would be accommodated in a slot associated to a ClassificationNode, rather than being expressed as a new object.

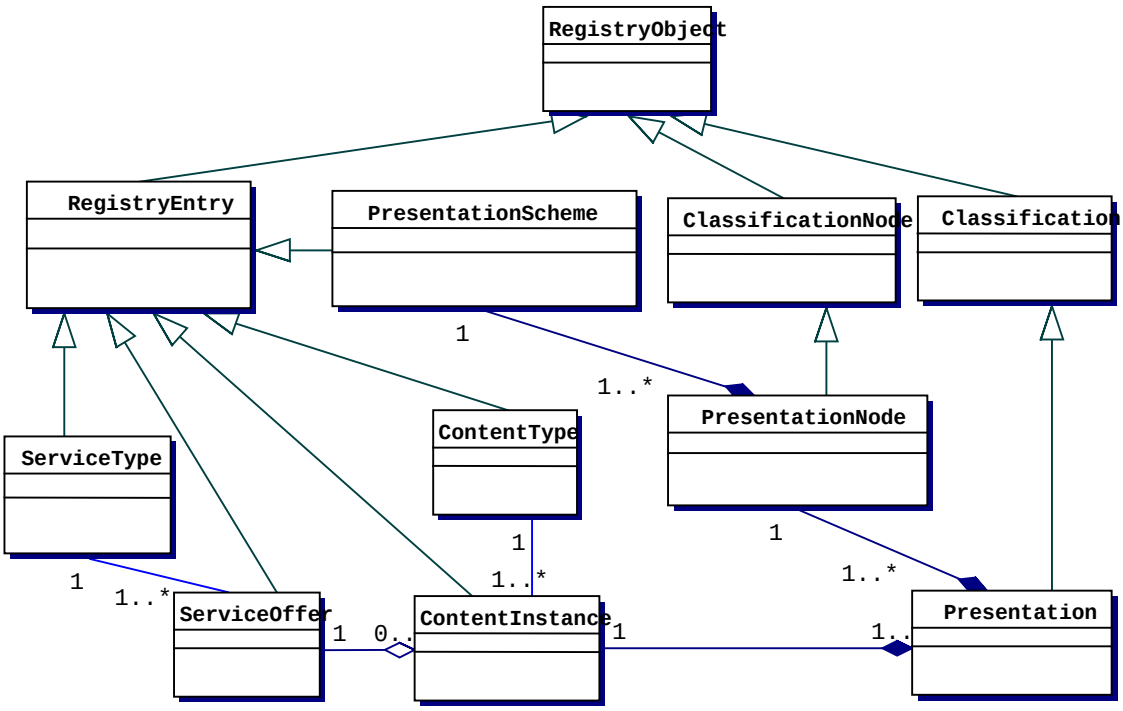

Figure 2. Service Information Model

The association relationships are also shown in this diagram, e.g. one or more *ServiceOffers* are instances of a *ServiceType*, one or more *ContentInstances* are instances of *ContentTypes*, while a *ServiceOffer* aggregates zero or more *ContentInstances*., etc.

10.2 Physical Object Diagram

TBD

11 Areas for Future Work

11.1 Issues

11.1.1 "Meta-schema" for service content

11.1.2 Content associations

11.1.3 Chain capabilities and invocation

11.1.4 Chain capabilities and data flow re-direction

11.1.5 Transaction capabilities and invocation

11.2 Future Directions

11.2.1 Alternate expression of parameterbinding (Schematron?)

11.2.2 Alternate expression of metadata (RDF?)

Annex A
(normative)

XML Schema for OWS Service Information Model

A XML Schema for OWS Service Information Model

A.1 Introduction

The following examples of XML Schema represent each of the SIM objects discussed earlier. Since the authoritative schema is still being refined, the reader is referred to the OWS1.2 schema repository for the latest versions of each of the following types.

Note also that the schema for each information object imports or includes other schema documents located at specific relative locations in the repository, so that one of these fragments in isolation will not validate (with any working validator…)

A.2 OGCCapabilities

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:iso="http://www.opengis.net/iso19119"
xmlns:ows="http://www.opengis.net/ows"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.0"
xmlns:reg="http://www.opengis.net/registry"
targetNamespace="http://www.opengis.net/ows"
elementFormDefault="qualified">
   <include schemaLocation="ServiceType.xsd" />
   <include schemaLocation="ServiceOffer.xsd" />
   <include schemaLocation="ContentType.xsd" />
   <include schemaLocation="ContentInstance.xsd" />
   <include schemaLocation="../../ogcrim/0.1.20/parameterBinding.xsd"
/>
   <import namespace="urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.0"
schemaLocation="../../ogcrim/0.1.20/ebRIM.xsd" />
   <import namespace="http://www.opengis.net/gml"
schemaLocation="../../gml/2.1.1/geometry.xsd" />
   <import namespace="http://www.opengis.net/registry"
schemaLocation="../../ogcrim/0.1.20/OGCRIM.xsd" />
   <import namespace="http://www.opengis.net/iso19119"
schemaLocation="iso19119_brief.xsd" />
   <element name="OGC_Capabilities">
      <complexType>
         <sequence>
            <element name="ServiceType" type="ows:serviceTypeType"
minOccurs="0" maxOccurs="unbounded" />
            <element name="ServiceOffer" type="ows:serviceOfferType"
minOccurs="0" maxOccurs="unbounded" />
            <element name="ContentTypeList" minOccurs="0">
```

```
                <complexType>
                    <sequence minOccurs="0" maxOccurs="unbounded">
                        <element name="ContentType"
type="ows:contentTypeType" />
                    </sequence>
                    <!-- attributes here -->
                </complexType>
            </element>
            <element name="ContentInstanceList" minOccurs="0">
                <complexType>
                    <sequence minOccurs="0" maxOccurs="unbounded">
                        <element name="ContentInstance"
type="ows:contentInstanceType" minOccurs="0" maxOccurs="unbounded" />
                    </sequence>
                    <!-- attributes here -->
                </complexType>
            </element>
            <element name="PresentationScheme"
type="rim:ClassificationSchemeType" minOccurs="0" maxOccurs="unbounded"
/>
        </sequence>
        <attribute name="version" type="string" />
        <attribute name="service" type="string" />
    </complexType>
</element>
</schema>
```

A.3 ServiceType

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows"
xmlns:reg="http://www.opengis.net/registry"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xlink="http://www.w3.org/1999/xlink"
targetNamespace="http://www.opengis.net/ows"
elementFormDefault="qualified">
    <import namespace="http://schemas.xmlsoap.org/wsdl/"
schemaLocation="../../wsdl/wsdl-1.11.xsd" />
    <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="../../gml/2.1.1/xlinks.xsd" />
    <import namespace="http://www.opengis.net/registry"
schemaLocation="../../ogcrim/0.1.20/OGCRIM.xsd" />
    <include schemaLocation="../../ogcrim/0.1.20/bindingTemplate.xsd" />
    <complexType name="serviceTypeType">
        <complexContent>
            <extension base="reg:OGCRegistryEntryType">
                <sequence>
                    <element name="typeWSDL" minOccurs="0">
                        <complexType>
                            <sequence>
                                <element ref="wsdl:types" minOccurs="0" />
                                <element ref="wsdl:message" minOccurs="0"
maxOccurs="unbounded" />
```

```
                        <element ref="wsdl:portType" minOccurs="0"
maxOccurs="unbounded" />
                    </sequence>
                    <attribute name="targetNamespace" type="anyURI"
use="optional" />
                </complexType>
            </element>
            <element name="parameterBinding"
type="ows:parameterBindingType" minOccurs="0" maxOccurs="unbounded" />
            <element ref="iso:ISO19119" minOccurs="0" />
        </sequence>
        <attribute name="serviceTypeId" type="anyURI" />
        <attributeGroup ref="xlink:simpleLink" />
    </extension>
  </complexContent>
</complexType>
</schema>
```

A.4 ServiceOffer

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0.1 U (http://www.xmlspy.com) by Joshua
Lieberman (Joshua Lieberman) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows"
xmlns:reg="http://www.opengis.net/registry"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:iso="http://www.opengis.net/iso19119"
xmlns:xlink="http://www.w3.org/1999/xlink"
targetNamespace="http://www.opengis.net/ows"
elementFormDefault="qualified">
   <import namespace="http://schemas.xmlsoap.org/wsdl/"
schemaLocation="../../wsdl/wsdl-1.11.xsd" />
   <import namespace="http://www.opengis.net/iso19119"
schemaLocation="iso19119_brief.xsd" />
   <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="../../gml/2.1.1/xlinks.xsd" />
   <import namespace="http://www.opengis.net/registry"
schemaLocation="../../ogcrim/0.1.20/OGCRIM.xsd" />
   <include schemaLocation="../../ogcrim/0.1.20/parameterBinding.xsd"
/>
   <complexType name="serviceOfferType">
      <complexContent>
         <extension base="reg:OGCRegistryEntryType">
            <sequence>
               <element name="instanceWSDL" minOccurs="0">
                  <complexType>
                     <sequence>
                        <element ref="wsdl:binding" minOccurs="0"
maxOccurs="unbounded" />
```

```
                        <element ref="wsdl:service" minOccurs="0"
maxOccurs="unbounded" />
                    </sequence>
                </complexType>
            </element>
            <element ref="iso:ISO19119" minOccurs="0" />
            <element ref="ows:ServiceSpecificMetadata"
minOccurs="0" />
            <element name="parameterBinding"
type="ows:parameterBindingType" minOccurs="0" maxOccurs="unbounded" />
        </sequence>
        <attribute name="serviceInstanceId" type="anyURI" />
        <attribute name="serviceTypeId" type="anyURI" />
        <attributeGroup ref="xlink:simpleLink" />
      </extension>
    </complexContent>
  </complexType>
</schema>
```

A.5 ContentType

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows"
xmlns:reg="http://www.opengis.net/registry"
xmlns:iso="http://www.opengis.net/iso19119"
xmlns:xlink="http://www.w3.org/1999/xlink"
targetNamespace="http://www.opengis.net/ows"
elementFormDefault="qualified">
  <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="../../gml/2.1.1/xlinks.xsd" />
  <import namespace="http://www.opengis.net/iso19119"
schemaLocation="iso19119_brief.xsd" />
  <import namespace="http://www.opengis.net/registry"
schemaLocation="../../ogcrim/0.1.20/OGCRIM.xsd" />
  <include schemaLocation="CommonMetadataTypes.xsd" />
  <complexType name="contentTypeType">
    <complexContent>
      <extension base="reg:OGCRegistryEntryType">
        <sequence>
          <!-- ommitted while details of Stephane Fellah's
proposal emerge -->
          <!--<element ref="ows:keywords" minOccurs="0" /> -->
          <element name="definition">
            <complexType>
              <sequence minOccurs="0">
                <any namespace="##any" processContents="lax"
/>
              </sequence>
              <attribute name="markupLanguage"
type="ows:mlListType" />
              <attributeGroup ref="xlink:simpleLink" />
            </complexType>
          </element>
        </sequence>
```

```
            <attribute name="contentTypeId" type="token"
use="required" />
            <!--    the definition does not actually have to be
present...it can be referenced by
                        url.  However, the document referenced might not
be exactly the definition, so we
                        need to point to the correct location with an
xpath string
            -->
            <attributeGroup ref="xlink:simpleLink" />
         </extension>
      </complexContent>
   </complexType>
   <simpleType name="mlListType">
      <restriction base="token">
         <enumeration value="http://www.opengis.net/gml/1.0" />
         <enumeration value="http://www.opengis.net/gml/2.0" />
         <enumeration value="http://www.opengis.net/gml/3.0" />
         <enumeration value="http://www.opengis.net/sld/0.7x" />
         <enumeration value="http://www.opengis.net/sensorml" />
         <enumeration value="##binary" />
         <enumeration value="##other" />
      </restriction>
   </simpleType>
</schema>
```

A.6 ContentInstance

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0.1 U (http://www.xmlspy.com) by Joshua
Lieberman (Joshua Lieberman) -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows"
xmlns:gml="http://www.opengis.net/gml"
xmlns:reg="http://www.opengis.net/registry"
xmlns:iso19119="http://www.opengis.net/iso19119"
xmlns:xlink="http://www.w3.org/1999/xlink"
targetNamespace="http://www.opengis.net/ows"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <import namespace="http://www.opengis.net/gml"
schemaLocation="../../gml/2.1.1/geometry.xsd" />
   <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="../../gml/2.1.1/xlinks.xsd" />
   <import namespace="http://www.opengis.net/iso19119"
schemaLocation="iso19119_brief.xsd" />
   <import namespace="http://www.opengis.net/registry"
schemaLocation="../../ogcrim/0.1.20/OGCRIM.xsd" />
   <include schemaLocation="../../ogcrim/0.1.20/parameterBinding.xsd"
/>
   <include schemaLocation="CommonMetadataTypes.xsd" />
   <complexType name="contentInstanceType">
      <complexContent>
         <extension base="reg:OGCRegistryEntryType">
            <sequence>
               <element ref="ows:ContentInstanceMetadata" />
```

```
                <element ref="ows:ContentInstanceGeospatialMetadata"
minOccurs="0" />
                <element ref="ows:ServiceSpecificMetadata"
minOccurs="0" />
                <element name="parameterBinding"
type="ows:parameterBindingType" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
            <attribute name="contentInstanceId" type="string"
use="required" />
            <attribute name="contentTypeId" type="string"
use="required" />
            <attribute name="serviceInstanceId" type="string"
use="required" />
            <attributeGroup ref="xlink:simpleLink" />
         </extension>
      </complexContent>
   </complexType>
</schema>
```

A.7 Presentation

See ogcRIM schema. TBD

A.8 PresentationNode

See ogcRIM schema. TBD

A.9 ClassificationScheme

See ogcRIM schema.

A.10 ParameterBinding

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:ows="http://www.opengis.net/ows"
   targetNamespace="http://www.opengis.net/ows"
   elementFormDefault="qualified">
   <element name="parameterBinding" type="ows:parameterBindingType" />
   <complexType name="parameterBindingType">
      <sequence>
         <element name="operation" type="string" minOccurs="0"
maxOccurs="unbounded" />
         <element name="parameter" maxOccurs="unbounded">
            <complexType>
               <choice>
                  <element ref="ows:_domainType" />
               </choice>
               <attribute name="type" type="QName" use="optional" />
               <attribute name="name" type="token" use="optional" />
               <attribute name="use" type="token" use="optional"
default="required"/>
```

```
                <attribute name="default" type="token" use="optional"
/>
            </complexType>
         </element>
      </sequence>
   </complexType>
   <element name="_domainType" abstract="true" type="anyType" />
   <element name="range" substitutionGroup="ows:_domainType">
      <complexType>
         <sequence>
            <element name="min" type="string" />
            <element name="max" type="string" />
         </sequence>
         <attribute name="units" type="string" use="optional" />
      </complexType>
   </element>
   <element name="enumeration" substitutionGroup="ows:_domainType">
      <complexType>
         <sequence>
            <element name="value" type="string" maxOccurs="unbounded"
/>
         </sequence>
      </complexType>
   </element>
   <element name="schema" substitutionGroup="ows:_domainType">
      <complexType>
         <sequence>
            <xs:any processContents="lax" />
         </sequence>
      </complexType>
   </element>
</schema>
```

A.11 Common Metadata

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ows="http://www.opengis.net/ows"
xmlns:iso19119="http://www.opengis.net/iso19119"
xmlns:gml="http://www.opengis.net/gml"
targetNamespace="http://www.opengis.net/ows"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <annotation>
      <documentation xml:lang="en">This file contains three main
element content-models.
         1)  ContentInstanceMetadata - this is a container for generic
metadata about content.  Things like Authority, Keywords, Contact Info,
etc. go here.
         2)  ContentInstanceGeospatialMetadata - this is a container
for generic geospatial metadata about content.  Things like SRS and
extents/bboxes go here.
         3)  ServiceSpecificMetadata - this is an "extension" container
for any specialized metadata that the ContentInstance for a particular
service would need.
```

```
                    It augments the basic model in an "application-
extension" manner.
       </documentation>
   </annotation>
   <import namespace="http://www.opengis.net/gml"
schemaLocation="../../gml/2.1.1/geometry.xsd" />
   <import namespace="http://www.opengis.net/iso19119"
schemaLocation="iso19119_brief.xsd" />

   <!-- three container elements -->
   <element name="ContentInstanceMetadata">
      <complexType>
         <sequence>
            <element ref="ows:abstract" minOccurs="0" />
            <element ref="ows:alternateIdentifier" minOccurs="0" />
            <element ref="ows:authority" minOccurs="0"
maxOccurs="unbounded" />
            <element ref="ows:contact" minOccurs="0" />
            <element ref="ows:metadata" minOccurs="0" />
         </sequence>
      </complexType>
   </element>
   <element name="ContentInstanceGeospatialMetadata">
      <complexType>
         <sequence>
            <element ref="ows:nativeSRS" minOccurs="0"/>
            <element ref="ows:extent" minOccurs="0"
maxOccurs="unbounded" />
         </sequence>
      </complexType>
   </element>
   <element name="ServiceSpecificMetadata">
      <complexType>
         <sequence minOccurs="0" maxOccurs="unbounded">
            <!-- this should be set to processContents="strict", but
for now we'll leave it lax...so people can adopt without schema -->
            <any processContents="lax" namespace="##other" />
         </sequence>
      </complexType>
   </element>
   <!-- supporting elements -->
   <element name="abstract" type="string" />
   <element name="alternateIdentifier">
      <complexType>
         <simpleContent>
            <extension base="string">
               <attribute name="authority" type="string"/>
            </extension>
         </simpleContent>
      </complexType>
   </element>
   <element name="authority">
      <complexType>
         <sequence>
            <any namespace="##other" processContents="lax"
maxOccurs="unbounded" minOccurs="0" />
```

```
            </sequence>
            <attribute name="name" type="string" />
            <attribute name="url" type="anyURI" />
        </complexType>
    </element>
    <element name="contact" type="iso19119:pointOfContactType" />
    <element name="extent" type="gml:BoxType" />
    <element name="metadata">
        <complexType>
            <attribute name="url" type="anyURI" />
            <attribute name="type" type="string" />
            <attribute name="format" type="token" />
        </complexType>
    </element>
    <element name="nativeSRS" type="string" />
    <element name="serviceSpecific">
        <complexType>
            <choice>
                <any namespace="##other" processContents="lax" />
            </choice>
        </complexType>
    </element>
    <!-- ommitted while details of Stephane Fellah's proposal emerge -->
    <!--
    <element name="keywords" type="iso19119:keywordsType" />
    -->
</schema>
```

Annex B
(informative)

Capabilities Document Examples

B Capabilities Document Examples

B.1 Introduction

In this annex are presented informative examples of Service capabilities documents which illustrate some but not all of the ways in which the SIM can be implemented and extended for this purpose.

B.2 Web Map Service

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This 2002-08-07 revision follows discussions between Saul Farber
and Jeff DLB. -->
<OGC_Capabilities
   xmlns="http://www.opengis.net/ows"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ows="http://www.opengis.net/ows"
   xsi:schemaLocation="http://www.opengis.net/ows

   /home/saul/ows1.2/schemas/ows1.2/capabilities/0.1.20/OGC_Capabilitie
s.xsd">
   <ServiceOffer serviceTypeId="http://www.opengis.net/ows/wms/1.1.13"
title="MetroGIS WMS"
serviceInstanceId="org:datafinder:wms:1.1.13:dataviewer">
      <instanceWSDL
xmlns:tns="http://www.opengis.net/namespace/for/authoritative/GetCapabil
ities/R7/WSDL">
         <binding xmlns="http://schemas.xmlsoap.org/wsdl/"
name="GetCapabilitiesPOSTBinding" type="tns:GetCapabilitiesPortType">
            <http:binding
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" verb="POST" />
            <operation name="GetCapabilities">
               <http:operation
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" location="main.jsp"
/>
               <input>
                  <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" type="text/xml" />
                  <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.kvp" />
               </input>
               <output>
                  <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.wms_xml" />
```

```
                </output>
                <fault name="exception">
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.se_xml" />
                </fault>
            </operation>
        </binding>
        <binding xmlns="http://schemas.xmlsoap.org/wsdl/"
name="GetCapabilitiesGETBinding" type="tns:GetCapabilitiesPortType">
            <http:binding
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" verb="GET" />
            <operation name="GetCapabilities">
                <http:operation
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" location="main.jsp"
/>
                <input>
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.kvp" />
                </input>
                <output>
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.wms_xml" />
                </output>
                <fault name="exception">
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.se_xml" />
                </fault>
            </operation>
        </binding>
        <binding xmlns="http://schemas.xmlsoap.org/wsdl/"
name="GetMapPOSTBinding" type="tns:GetMapPortType">
            <http:binding
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" verb="POST" />
            <operation name="GetMap">
                <http:operation
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" location="main.jsp"
/>
                <input>
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.kvp" />
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" type="text/xml" />
                </input>
                <output>
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" type="image/jpeg" />
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" type="image/png" />
                </output>
                <fault name="exception">
```

```
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.se_xml" />
                </fault>
            </operation>
        </binding>
        <binding xmlns="http://schemas.xmlsoap.org/wsdl/"
name="GetMapGETBinding" type="tns:GetMapPortType">
            <http:binding
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" verb="GET" />
            <operation name="GetMap">
                <http:operation
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" location="main.jsp"
/>
                <input>
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.kvp" />
                </input>
                <output>
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" type="image/jpeg" />
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" type="image/png" />
                </output>
                <fault name="exception">
                    <mime:content
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
type="application/vnd.ogc.se_xml" />
                </fault>
            </operation>
        </binding>
        <service xmlns="http://schemas.xmlsoap.org/wsdl/" name="WMS">
            <port name="GetCapabilitiesPOSTPort"
binding="tns:GetCapabilitiesPOSTBinding">
                <http:address
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
location="http://localhost:85/mapaccess/" />
            </port>
            <port name="GetCapabilitiesGETPort"
binding="tns:GetCapabilitiesGETBinding">
                <http:address
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
location="http://localhost:85/mapaccess/" />
            </port>
            <port name="GetMapPOSTPort"
binding="tns:GetMapPOSTBinding">
                <http:address
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
location="http://localhost:85/mapaccess/" />
            </port>
            <port name="GetMapGETPort" binding="tns:GetMapGETBinding">
                <http:address
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
location="http://localhost:85/mapaccess/" />
            </port>
```

```
            </service>
        </instanceWSDL>
        <ISO19119 xmlns="http://www.opengis.net/iso19119">
            <serviceType>
                <nameValue>WMS</nameValue>
                <nameNameSpace>OGC</nameNameSpace>
            </serviceType>
            <serviceTypeVersion>1.1.20</serviceTypeVersion>
            <citation>
                <title>MetroGIS WMS</title>
            </citation>
            <abstract>MetroGIS WMS Running MapAccess OGC
Services</abstract>
        </ISO19119>
        <parameterBinding>
            <operation>GetCapabilities</operation>
            <parameter type="ServiceParameterType">
                <enumeration>
                    <value>WMS</value>
                </enumeration>
            </parameter>
            <parameter type="VersionParameterType" use="optional">
                <enumeration>
                    <value>1.1.0</value>
                    <value>1.1.1</value>
                    <value>1.1.20</value>
                </enumeration>
            </parameter>
            <parameter type="ViewParameterType" use="optional">
                <enumeration>
                    <!-- These view names are to be defined in Service
Information Model.
                         Each service type must define a schema for each
relevant view.
                         There may be other view names for other services.
                         In the WMS case, the information returned in each
view would be
                         approximately the following (schemas t.b.d.):
                         - Brief: service-level metadata + list of layer
names & titles
                         - Full: the whole enchilada
                         - ServiceOffer: just the info that in this version
of the XML is
                         included in the ServiceOffer element.
                         - ISO19119: just the info that in this version of
the XML is
                         included in the ISO19119 element.
                         - ContentInstance: ServiceOffer, ISO19119 and
complete content
                         instance metadata for a single content object (i.e.,
the
                         same as a Full response from a server that only
contains
                         that one single content object).
                    -->
                    <value>Brief</value>
```

```
                <value>Full</value>
                <value>ServiceOffer</value>
                <value>ISO19119</value>
                <value>ContentInstance</value>
            </enumeration>
        </parameter>
    </parameterBinding>
    <parameterBinding>
        <operation>GetMap</operation>
        <parameter type="WidthType">
            <range>
                <min>2</min>
                <max>8192</max>
            </range>
        </parameter>
        <parameter type="HeightType">
            <range>
                <min>2</min>
                <max>8192</max>
            </range>
        </parameter>
        <parameter type="ServiceParameterType">
            <enumeration>
                <value>WMS</value>
            </enumeration>
        </parameter>
        <parameter type="VersionParameterType">
            <enumeration>
                <value>1.1.13</value>
            </enumeration>
        </parameter>
        <parameter type="SrsParameterType">
            <enumeration>
                <value>EPSG:4326</value>
                <value>EPSG:26915</value>
            </enumeration>
        </parameter>
        <!--<parameter type="BboxParameterType">-->
            <!-- no enumeration or range, because allowed values are
SRS-specific.
                BboxType defines (for HTTP GET) sequence
"xmin,ymin,xmax,ymax".
                Client must inspect <extent> elements elsewhere in this
document.

                -extra comment by saul 8/8/02-
                We just can't figure out a good way to do this with the
current
                constraints of XMLSchema + WSDL.  Perhaps future
consideration
                of other technologies like XQuery or Schematron will
shed light on
                how to do dynamic parameter binding of extents/bboxes.
            -->
        <!--</parameter>-->
        <parameter type="FormatParameterType">
```

```
            <enumeration>
               <value>image/gif</value>
               <value>image/png</value>
               <!-- note the following legal type per MIME spec -->
               <value>image/png; PhotometricInterpretation=RGB</value>
            </enumeration>
         </parameter>
         <parameter type="TransparentParameterType" use="optional"
default="FALSE">
            <enumeration>
               <value>TRUE</value>
               <value>FALSE</value>
            </enumeration>
         </parameter>
         <parameter type="BgcolorParameterType" use="optional"
default="0xFFFFFF">
            <range>
               <min>0x000000</min>
               <max>0xFFFFFF</max>
            </range>
         </parameter>
         <parameter type="ExceptionsParameterType" use="optional">
            <enumeration>
               <value>application/vnd.ogc.se+xml</value>
               <value>application/vnd.ogc.inimage</value>
            </enumeration>
         </parameter>
         <!-- A Feature or Coverage Portrayal Service would need
            parameter elements for SLDParameterType and
            RemoteOWSParameterType
         -->
      </parameterBinding>
      <!-- Some servers will need a parameterBinding template for
         GetFeatureInfo
      -->
      <!-- Feature/Coverage Portrayal Services will need
         parameterBinding sections for GetLegendGraphic, GetStyles,
         PutStyles operations.
      -->
   </ServiceOffer>
   <ContentInstanceList>
      <ContentInstance title="2000 Census Water Features (TIGER)"
contentTypeId="http://www.opengis.org/wmslayer"
serviceInstanceId="org:datafinder:wms:1.1.13:dataviewer"
contentInstanceId="water_2000_a">
         <VendorSpecific xmlns="http://www.opengis.net/registry">
            <ArcIMS connectionMethod="ServletConnector"
layerId="water_2000_a" isExtractable="true"
mapServiceId="MN_MetroGIS_DataFinder_Demographics" port="80"
host="localhost" layerType="feature" />
            <ExtractDoc
url="http://www.datafinder.org/notices/metc_notice.htm" />
         </VendorSpecific>
         <ContentInstanceMetadata>
```

```
            <alternateIdentifier
authority="http://gcmd.gsfc.nasa.gov/difguide/whatisadif.html">123456</a
lternateIdentifier>
            <authority name="Tanya Mayer, GIS Coordinator"
url="http://www.datafinder.org" />
            <contact>
               <contactInfo xmlns="http://www.opengis.net/iso19119">
                  <phone>
                     <voice>651-602-1604</voice>
                  </phone>
                  <address>
                     <deliveryPoint>Metropolitan Council, 230 East
Fifth Street, St. Paul, Minnesota 55101</deliveryPoint>
                  </address>
                  <onLineResource>
                     <linkage type="simple"
href="mailto://tanya.mayer@metc.state.mn.us"
xlink="http://www.w3.org/1999/xlink" />
                  </onLineResource>
               </contactInfo>
               <roleCode xmlns="http://www.opengis.net/iso19119"
value="pointOfContact" />
            </contact>
            <metadata type="FGDC" format="text/html"
url="http://www.datafinder.org/metadata/census_2000_tiger.htm" />
            <!-- NOTE: the Keywords section is omitted until we
understand
               Stephane Fellah's proposal.  Development and TIEs can
proceed
               with the temporary absence of Keyword information. -->
               <!-- keywords -->
               <!-- /keywords -->
               <!-- This replaces the AuthorityURL/Identifier
combination of
               WMS 1.1 (0 or more)
            -->
         </ContentInstanceMetadata>
         <ContentInstanceGeospatialMetadata>
            <nativeSRS>EPSG:26915</nativeSRS>
            <extent srsName="EPSG:4236">
               <gml:coord xmlns:gml="http://www.opengis.net/gml">
                  <gml:X>-94.062</gml:X>
                  <gml:Y>44.421</gml:Y>
               </gml:coord>
               <gml:coord xmlns:gml="http://www.opengis.net/gml">
                  <gml:X>-92.682</gml:X>
                  <gml:Y>45.465</gml:Y>
               </gml:coord>
            </extent>
            <extent srsName="EPSG:26915">
               <gml:coord xmlns:gml="http://www.opengis.net/gml">
                  <gml:X>415458.1783425122</gml:X>
                  <gml:Y>4918591.134803784</gml:Y>
               </gml:coord>
               <gml:coord xmlns:gml="http://www.opengis.net/gml">
                  <gml:X>525314.7538104662</gml:X>
```

```
                    <gml:Y>5035064.439980954</gml:Y>
                </gml:coord>
            </extent>
        </ContentInstanceGeospatialMetadata>
        <ows:ServiceSpecificMetadata
xmlns="http://www.opengis.net/wms/R7metadata">
            <Style name="red">
                <Title>Red filled polygons</Title>
                <LegendWidth>180</LegendWidth>
                <LegendHeight>50</LegendHeight>
                <LegendFormat>image/gif</LegendFormat>
                <LegendURL>http://host.sub.domain/red.png</LegendURL>
                <ScaleHint max="100" min="10" />
            </Style>
            <Style name="blue">
                <Title>Blue outlines</Title>
                <LegendWidth>180</LegendWidth>
                <LegendHeight>50</LegendHeight>
                <LegendFormat>image/gif</LegendFormat>
                <LegendURL>http://host.sub.domain/blue.png</LegendURL>
                <ScaleHint max="100" min="10" />
            </Style>
        </ows:ServiceSpecificMetadata>
        <parameterBinding>
            <operation>GetMap</operation>
            <parameter type="LayerParameterType">
                <enumeration>
                    <value>water_2000_a</value>
                </enumeration>
            </parameter>
            <parameter type="StylesParameterType">
                <enumeration>
                    <value>blue</value>
                    <value>red</value>
                </enumeration>
            </parameter>
        </parameterBinding>
    </ContentInstance>
</ContentInstanceList>
<PresentationScheme title="DataViewer Menu" id="metrogis_tax_1"
schemeType="ContentInstance">
    <PresentationNode xmlns="urn:oasis:names:tc:ebxml-
regrep:rim:xsd:2.0" title="DataFinder Catalog">
        <PresentationNode title="Demographics">
            <PresentationNode title="Census 2000">
                <Presentation classifiedObject="water_2000_a">
                    <Description>
                        <LocalizedString value="2000 Census Water
Features (TIGER)" />
                    </Description>
                </Presentation>
            </PresentationNode>
        </PresentationNode>
    </PresentationNode>
</PresentationScheme>
</OGC_Capabilities>
```

B.3 Web Feature Service
TBD

B.4 Web Registry Service
TBD

B.5 Web Coverage Service
TBD

B.6 Sensor Collection Service
TBD

B.7 Web Notification Service
TBD

B.8 Sensor Planning Service
TBD

B.9 Image Archive
TBD

B.10 Image Catalog
TBD

B.11 Style/Symbol Management Service
TBD

B.12 LOF Manager
TBD

Bibliography

[1] OGC 02-112r2, OGC Abstract Specification Topic 12: Service Architecture, available at: http://www.opengis.org/techno/abstract/02-112.pdf (contains ISO/DIS 19119: Geographic information – Services)

[2] OGC 02-034, Geography Markup Language version 3 schemas draft, available at: http://member.opengis.org/tc/archive/arch02/02-034.zip

[3] OGC 02-017r1, Web Map Service Implementation Specification Part 2: XML for Requests using HTTP POST http://www.opengis.org/techno/discussions/02-017r1.pdf

[4] OGC 02-019r1, Coverage Portrayal Server Implementation Specification draft, available at: http://member.opengis.org/tc/archive/arch02/02-019r1.pdf

[5] OGC 02-024, Web Coverage Server Implementation Specification draft, available at: http://www.opengis.org/techno/discussions/02-024.pdf

[6] Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000, available at: http://www.w3.org/TR/REC-xml

[7] OGC 99-051, OpenGIS Catalog Interface Implementation Specification, available at: http://www.opengis.org/techno/specs/99-051.pdf

[8] OGC 01-024r1, Web Registry Server Implementation Specification draft, available at: http://member.opengis.org/tc/archive/arch01/01-024r1.doc

[9] OGC 01-065, Web Feature Server Implementation Specification, available at: http://member.opengis.org/tc/archive/arch01/01-065.doc

[10] OGC 01-067, Filter Encoding Implementation Specification, available at: http://member.opengis.org/tc/archive/arch01/01-067.doc

[11] OGC 02-009, Geography Markup Language version 2.1.1, available at: http://www.opengis.net/gml/02-009/GML2-11.pdf

[12] OGC 02-013, Styled Layer Description Implementation Specification, available at: http://member.opengis.org/tc/archive/arch02/02-013.doc