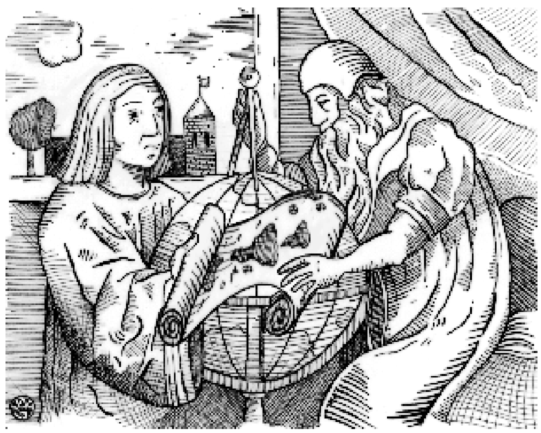
OGC 03-040

# Open Geospatial Consortium Inc.

Date: 2003-09-16
Reference number: **OGC 03-040**
Version: 0.1.3
Category: OGC Reference Model
Editor: George Percivall

# OGC Reference Model





**Warning**

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

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.

# OGC 03-040

# Contents

Revision History ..... vi

Foreword ..... vii

1 Introduction ..... 1
1.1 Purpose and Scope ..... 1
1.2 Who Should Read This Document ..... 1
1.3 References ..... 2
1.4 Key Terms ..... 2
1.5 Reference Model for Open Distributed Processing ..... 3

2 Enterprise View point ..... 4
2.1 Geospatial: A Fundamental Information Ingredient ..... 5
2.2 Geospatial Information Value Chain ..... 5
2.3 Requirements on OGC Technologies ..... 6

3 Information Viewpoint: OpenGIS Information Framework ..... 7
3.1 Geographic features ..... 7
3.2 Spatial-temporal geometry and topology ..... 13
3.3 Spatial Referencing ..... 17
3.4 Coverages including imagery ..... 20
3.5 Portrayal and human interface ..... 23
3.6 Metadata ..... 24
3.7 Service Information Model ..... 26
3.8 Information communities ..... 27

4 Computational Viewpoint: OpenGIS Service Framework ..... 29
4.1 Services, Interfaces and Operations ..... 29
4.2 Publish, Find and Bind Pattern ..... 30
4.3 Service Type Examples ..... 32
4.4 Applying services ..... 38

5 Engineering viewpoint: multi-tier, multi-network architecture ..... 39
5.1 Multi-tier architectures ..... 39
5.2 Bridging multiple networks ..... 45
5.3 Distribution transparencies ..... 47

6 Technology viewpoint: Cross platform implementations ..... 48
6.1 Multi-platform Implementation Approach ..... 48
6.2 Implementation View of the OpenGIS Information Framework ..... 49
6.3 OGC Encodings; including GML ..... 52
6.4 Web Services Platform ..... 58

7 Scenarios Illustrating ORM ..... 67
7.1 Information Producer Scenario ..... 67

**7.2 Information Broker Scenario........................................................................76**
**7.3 Information Consumer Scenario ...................................................................83**

**Annex A: References..............................................................................................89**
**7.1 OpenGIS Abstract Specifications....................................................................89**
**7.2 OpenGIS Implementation Specifications .......................................................89**
**7.3 OpenGIS Recommendation Papers.................................................................90**
**7.4 OpenGIS Discussion Papers ............................................................................90**
**7.5 ISO TC211 References ....................................................................................91**
**7.6 Other References..............................................................................................91**

**Annex A Mapping ORM requirements to OGC services........................................93**

## Figures

Figure 1 - RM-ODP Viewpoints ........ 4
Figure 2 – Representative Geospatial Information Value Chain ........ 6
Figure 3 - Geographic Features ........ 8
Figure 4 - Modeling Geographic Information ........ 9
Figure 5 - Kernel of the General Feature Model ........ 10
Figure 6 - General Feature Model ........ 10
Figure 7 - Geometry Conceptual Schema; basic classes ........ 14
Figure 8 - Topological Class Diagram ........ 15
Figure 9 - Query operator examples ........ 16
Figure 10 - Coordinate Reference System ........ 19
Figure 11 - Coverage: Domain and Range ........ 21
Figure 12 - Earth Imagery ........ 21
Figure 13 - Portrayal model ........ 23
Figure 14 - Dataset Metadata ........ 25
Figure 15 - Service definition relationships. ........ 30
Figure 16 - Publish/Find/Bind Pattern ........ 31
Figure 17 - Service and Interface Types ........ 34
Figure 18 - The OWS Service Framework (OSF) ........ 35
Figure 19 - Decision Support Example ........ 38
Figure 20 - Service Chaining Example ........ 39
Figure 21 - Logical multi-tiered architecture mapped to different physical architectures 40
Figure 22 -Thin vs. thick clients for portraying features over the Internet ........ 41
Figure 23 - Thin vs. thick clients in the World Wide Web environment ........ 42
Figure 24 -Application Services and the OWS Services Framework ........ 43
Figure 25 - Mapping from platform independent UML models ........ 45
Figure 26 - OpenLS system concept ........ 46
Figure 27 - Sensor Collection Service concept ........ 46
Figure 28 - OpenGIS Information Framework ........ 49

Figure 29 - Services Interoperability Stack ........ 59
Figure 30 - Information Production Scenario ........ 68
Figure 31 - Enterprise Viewpoint: Production ........ 69
Figure 32 - Enterprise Viewpoint: Production Detail ........ 70
Figure 33 - Feature Extraction and Update ........ 72
Figure 34 - General Service Model ........ 73
Figure 35 - Production Model ........ 74
Figure 36 - Feature Extraction ........ 75
Figure 37 - NSDI, RSDI and GCCOG as an Enterprise ........ 78
Figure 38 - GCCOG Proposal Evaluation Process ........ 79
Figure 39 - Travel Assistance Consumer Scenario ........ 84
Figure 40 - Details of Personal Navigation Service ........ 85

# Revision History

| | | |
|---|---|---|
| Original Contributors | Kurt Buehler, Sam Bacharach, Carl Reed, Cliff Kottman, Chuck Heazel, John Davidson, Yaser Bisher, Harry Niedzwiadek, John Evans | April 2002 |
| Version 0 | Kurt Buehler and Harry Niedzwiadek | July 2002 |
| Version 0.1 | George Percivall | August 2002 |
| Version 0.1.1 Minor edits | Jeffrey Simon | February 2003 |
| Version 0.1.2 Minor Edits, extend terms and definitions | Carl Reed | March 2003 |
| Version 1.1.3 to update to reflect spec activity since ORM Document Adoption | Carl Reed | September 2003 |
| | | |
| | | |

# Foreword

This document documents the OGC Reference Model (ORM). The ORM describes a framework for the ongoing work of the Open Geospatial Consortium and our specifications and implementing interoperable solutions and applications for geospatial services, data, and applications.

The Open Geospatial Consortium (OGC) is an industry consortium whose members work in a collaborative, consensus process to enhance and enable interoperability for technologies involving spatial information and location. The OGC vision is a world in which everyone benefits from geographic information and services made available across any network, application, or platform. The OGC Mission is to deliver spatial interface and encoding specifications that are openly and publicly available for global use.

In the OGC Specification Program, the OGC Technical Committee and OGC Planning Committee work in a formal consensus process to arrive at approved (or "adopted") OpenGIS Specifications.

The OGC Interoperability Program is a series of hands-on engineering initiatives to accelerate the development and acceptance of OpenGIS Specifications.

The OGC Review Board creates and maintains documentation of procedural and technical baselines.

The OGC Review Board is responsible for the creation and maintenance of the OGC Reference Model (ORM). This document shall serve as the Technical Baseline of the OGC. The ORM document shall be subject to version control and shall be approved by the OGC Technical and Planning Committees for use as the Technical Baseline. The Review Board reviews RFP, RFC, RFT and RFQ documents as they are under development and before they are issued for consistency with procedural and technical baselines and to make change recommendations to the appropriate body. The Review Board shall use the ORM as its primary guidance for solving technical disputes.

# OGC Reference Model

## 1 Introduction

### 1.1 Purpose and Scope

The OGC Reference Model (ORM) provides an architecture framework for the ongoing work of the OGC. Further, the ORM provides a framework for the OGC Technical Baseline. The OGC Technical Baseline consists of the currently approved OpenGIS® Specifications as well as for a number of candidate specifications that are currently in progress.

The ORM has the following purposes:

– Provides a foundation for coordination and understanding (both internal and external to OGC) of ongoing OGC activities and the Technical Baseline;

– Update/Replacement of parts of the 1998 OpenGIS Guide;

– Describes the OGC requirements baseline for geospatial interoperability;

– Describes the OGC architecture framework through a series of non-overlapping viewpoints: including existing and future elements;

– Regularize the development of domain-specific interoperability architectures by providing examples.

The ORM is a living document that will be revised on a regular basis to continually and accurately reflect the ongoing work of the Consortium.

### 1.2 Who Should Read This Document

Any individual who:

– Wishes to better understand the ongoing work and Technical Baseline of the OGC;

– Wishes to implement one or more of the OpenGIS Specifications

– Wants the understanding necessary to make contributions to the OGC process.

### 1.3 References

See Annex A

### 1.4 Key Terms

**Application schema** – set of conceptual schema for data required by one or more applications.

**Client -** A software component that can invoke an operation from a server

**Conceptual** schemas (also called base schemas)

**Coordinate reference system** – coordinate system that has a reference to the Earth.

**Coverage** is a feature that associates positions within a bounded space (its spatiotemporal domain) to feature attribute values (its range)

**DCP** – Distributed Computing Platform

F**eature** – abstraction of a real world phenomenon.

**General feature model** – metamodel of feature types

**Geographic feature** – feature associated with a location relative to the Earth.

**GML** – Geographic Markup Language

**Interface –** named set of operations that characterize the behavior of an entity [6]

**Interoperability** - capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units [ISO 2382-1]

**Map projectio**n – coordinate conversion from a geodetic coordinate system to a planar surface.

**Metadata** – Data about data.

**OGC** – Open Geospatial Consortium

**Operation** – Specification of an interaction that can be requested from an object to effect behavior. [ISO 19119]

**Property -** A facet or attribute or an object referenced by a name

**ORM** – OGC Reference Model

**Service request -** An request by a client of an operation from a service.

**Service** - A collection of operations, accessible through an interface, that allows a user to evoke a behavior of value to the user. [ISO – 19119]

**Service chain -** sequence of services where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action [ISO 19119]

**Spatial Reference System –** As defined in the OpenGIS Abstract Specification Topic 2 and ISO 19111.

**Viewpoint** – form of abstraction achieved using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within a system. [ISO-10746-2]

### 1.5 Reference Model for Open Distributed Processing

The **Reference Model for Open Distributed Processing** (RM-ODP) [30] is an international standard for architecting open, distributed processing systems. It provides an overall conceptual framework for building distributed systems in an incremental manner. The RM-ODP standards have been widely adopted: they constitute the conceptual basis for the ISO 19100 series of geomatics standards (normative references in ISO/DIS 19119), and they also have been employed in the OMG object management architecture.

The application of RM-ODP in the design of ORM is two-fold: 1) a way of thinking about architectural issues in terms of fundamental patterns or organizing principles, and 2) a set of guiding concepts and terminology.

RM-ODP defines standard concepts and terminology for open, distributed processing. In a generic way, the model identifies the top priorities for architectural specifications and provides a minimal set of requirements—plus an object model—to ensure system integrity. Five standard *viewpoints* are defined; the viewpoints address different aspects of the system and enable the 'separation of concerns' (See Table 1)

**Table 1 - RM-ODP viewpoints**

| **Viewpoint Name** | **Definition of RM-ODP Viewpoint** |
|---|---|
| Enterprise | Focuses on the purpose, scope and policies for that system. |
| Information | Focuses on the semantics of information and information processing. |
| Computational | Captures component and interface details without regard to distribution |
| Engineering | Focuses on the mechanisms and functions required to support distributed interaction between objects in the system. |
| Technology | Focuses on the choice of technology. |

A graphical depiction of the relationships between the viewpoints is provided in Figure 1.

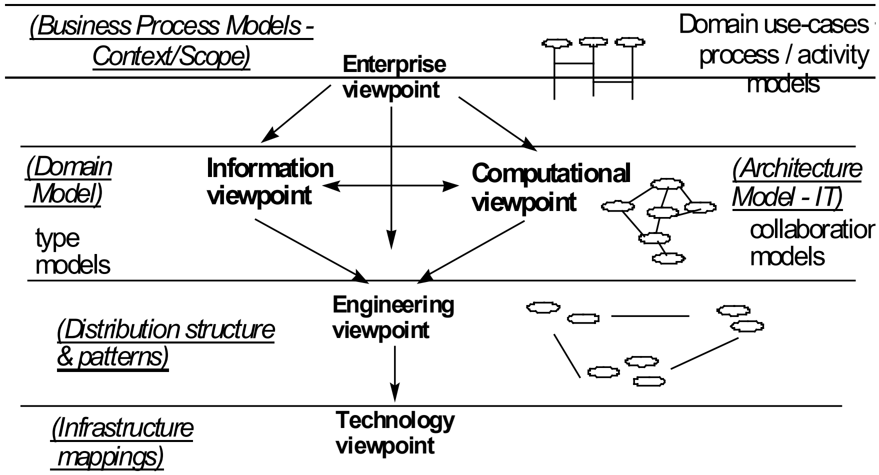



**Figure 1 - RM-ODP Viewpoints**

## 2 Enterprise View point

The enterprise viewpoint of the Open Geospatial Consortium, Inc (OGC) describes its business perspective, purpose, scope and policies. After highlighting the role of geospatial location as a fundamental information ingredient, this section provides a representative value chain of geospatial information within an enterprise or an information community. The section concludes by highlighting the major requirements on OGC technology as derived from the described value chain.

OGC is a not-for-profit trade association dedicated to promoting new technical and commercial approaches to interoperable geoprocessing. The OGC was founded in response to widespread recognition of the problem of non-interoperability in the geospatial industry and the many negative ramifications for industry, government and academia. With greater than 80% of business and government information having some reference to location (Ref. [31] page 54), the vision of OGC is that of a national and global infrastructure in which geospatial or location referenced data[1] and geoprocessing resources move freely, fully integrated with the latest distributed computing technologies, accessible to everyone. In addition, the OGC vision encompasses geo-enabling a wide variety of activities that are currently outside the domain of geoprocessing, opening new markets and giving rise to new kinds of businesses and new benefits to the public.

The core mission of OGC is to deliver spatial interface and encoding specifications that are openly and publicly available for global use. This mission is achieved through organizing interoperability projects, working toward consensus, formalizing OGC specifications, developing strategic business opportunities and standards partnerships, and promoting demand for

[1] We take a very broad view of what is geospatial and location referenced data. Examples range from traditional digital map data to survey data, satellite imagery, aerial imagery, GPS, sensor webs cell phone locations or any tabular database that has some location content such as addresses, phone numbers, IP addresses, landmark names, and place names.

interoperable products. The OGC Reference model described in this document focuses on the **technology aspects** of the OGC mission.

### 2.1 Geospatial: A Fundamental Information Ingredient

Geospatial **location** is integral to all aspects of the OGC Reference Model. Geospatial location is a foundational property for modeling the world in a coherent, intuitive way. Location and time can be exploited as a unifying theme to better understand the context of most real and abstract phenomena. Location is contextually simple and intuitive to most people. For example, people can relate to where they are on a map, follow directions to a place, readily grasping the spatial context of their local environment, and so forth.

Geospatial location is a ubiquitous information ingredient. Whether represented as a map or an image, encoded as an address, zip code, or phone number, described in a text passage as a landmark or event, or any of the many other ways of representing earth features and their properties, geospatial information is pervasive. Consider what could be accomplished if vast stores of unrelated information were related and organized on the basis of their common geospatial and temporal properties. Hidden meanings and understandings would be revealed, affected business processes and services would improve and new, collaborative business processes and services that would greatly enhance an enterprise's performance would be possible.

Geospatial location is also often considered as a valuable organizing principle for architecting and constructing enterprise data stores. Interoperable geospatial technologies can play a foundational role in exploiting these data stores for enterprise missions.

### 2.2 Geospatial Information Value Chain

Figure 2 illustrates a informative **value chain** for geospatial information within an enterprise or an information community. The value chain starts with geospatial information sources entering an interoperable environment, which then pass through geoprocessing chains, creating intermediate **value-add geospatial-based products** along the way. Figure 2 shows several examples of such value-add products including **fusion** (combining, correlating, annotating, and interrelating geospatial information from many sources into a single structure) and **analysis** (operating on geospatial information for the purpose of deriving new information, extracting results or understanding its nature and significance). The last step in the value chain involves creating **finished products** that contain geospatial information, or are derived from geospatial information, for both internal customers and external customers. Typical products provide functions such as visualization and portrayal, reporting, analysis or information transfer and dissemination.

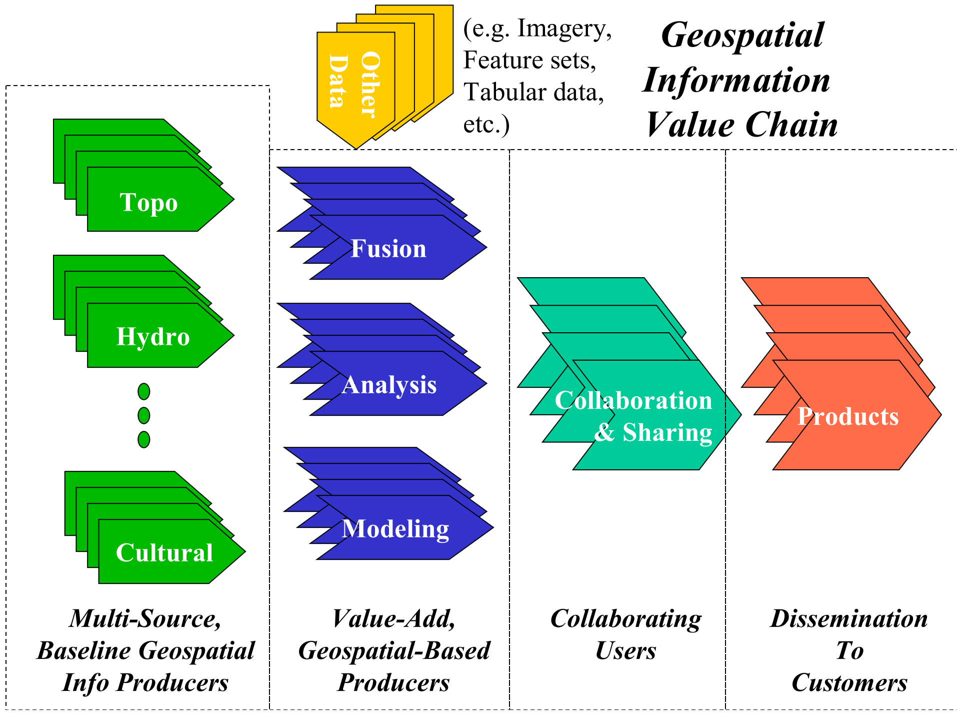



**Figure 2 – Representative Geospatial Information Value Chain**

### 2.3 Requirements on OGC Technologies

This section focuses on extracting key requirements on OGC technologies as derived from the value chain shown above. Before listing these requirements, it is important to note that OGC technologies are not limited to the **specifications** formalized by the consortium; they also include working **implementations** of such specifications. As such, in order to support the geospatial information value chain, OGC technologies must

- Be agile to be able to adapt to changing business rules and operational requirements;
- Support the easy and seamless introduction of new technologies and the evolution of existing ones;
- Provide for robustness and consistent error handling and recovery to support mission-critical systems development;
- Accommodate authentication, security and privacy features and support asset protection;
- Be platform independent (e.g., DCP, hardware, OS, programming language, encodings, etc);
- Support implementations of N-tiered, component architectures;

- Support standard interfaces and metadata while accommodating the use of other complementary standards and specifications in environments where OpenGIS specifications are implemented;
- Support interoperability by specifying interface definitions, service descriptions and protocols for software collaboration and negotiation;
- Accommodate independently developed implementations of a service and many independently provided instantiations of different types of services;
- Accommodate a wide range of data policies (e.g., data access and data use policies);
- Be vendor and data neutral;
- Be data content format independent.

# 3 Information Viewpoint: OpenGIS Information Framework

The information viewpoint is concerned with the semantics of information and information processing. The information viewpoint defines conceptual schemas for geospatial information and methods for defining application schemas. The conceptual, or base, schemas are formal descriptions of the model of any geospatial information. Application schemas are information models for a specific information community. Applications schemas are built from the conceptual schemas.

The Information viewpoint begins with a discussion of geographic features as the basic concept of modeling geospatial information, followed by several sections presenting the conceptual schemas, and closes with a single section on application schemas.

## 3.1 Geographic features

The starting point for modeling of geographic information is the geographic feature. A **feature** is an abstraction of a real world phenomenon. A **geographic feature** is a feature associated with a location relative to the Earth. A digital representation of the real world can be thought of as a set of features.

Geographic features occur at two levels: **feature instances** and **feature types** (See Figure 3). At the instance level, a geographic feature is represented as a discrete phenomenon that is associated with its geographic and temporal coordinates. These individual feature instances are grouped into classes with common characteristics — feature types. Geographic information is subjectively perceived and its content depends upon the needs of particular applications. The needs of particular applications determine the way instances are grouped into types within a particular classification scheme.

EXAMPLE The phenomenon 'Eiffel Tower' may be classified with other similar phenomena into a feature type 'tower'.

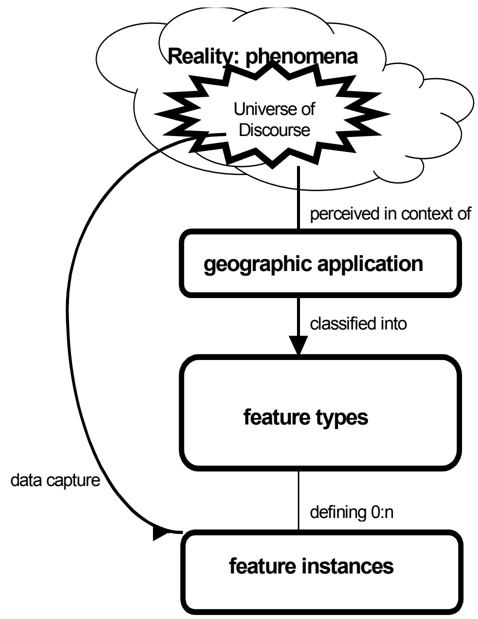



**Figure 3 - Geographic Features**

Figure 4 shows the OpenGIS approach to modeling geographic features. The OpenGIS Information Framework provides **conceptual schemas** to define abstract feature types and provides the process for domain experts to develop **application schemas** that are used to capture data about feature instances. The process for defining an application schema is described in Section 3.8.

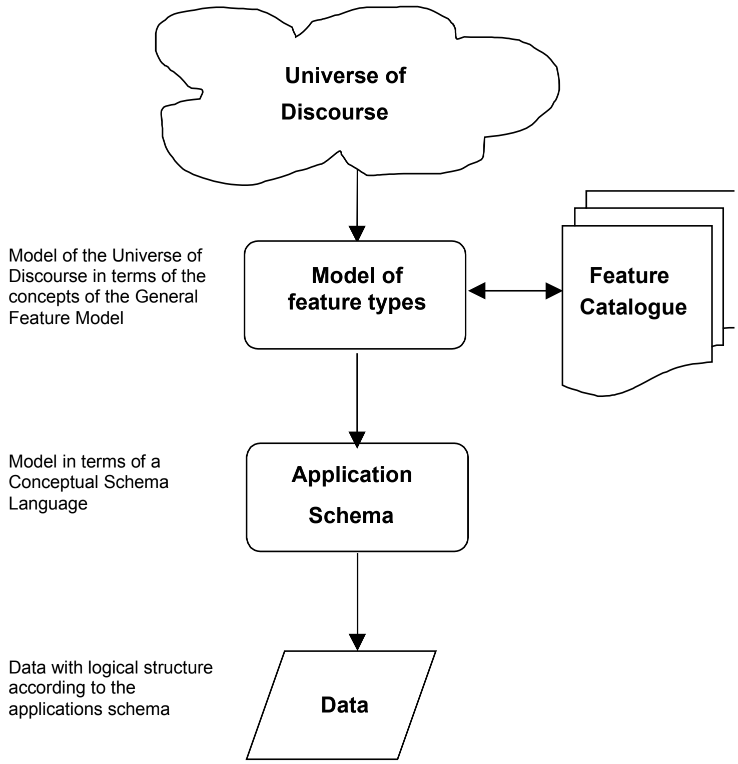



**Figure 4 - Modeling Geographic Information**

#### 3.1.1 General Feature Model

The things we want to classify we call features; the relations between feature types are feature association types and inheritance. Feature types have properties that are feature attributes, feature operations and feature association roles. These concepts are expressed in the **general feature model**. The general feature model is a metamodel of feature types. (See Figure 5 and Figure 6).

Any feature may have a number of **properties** that may be operations, attributes or associations. Any feature may have a number of **attributes**, some of which may be geometric and spatial. A feature is not defined in terms of a single geometry, but rather as a conceptually meaningful object within a particular domain of discourse, one or more of whose properties may be geometric.

**BASELINE**: A general feature model has not been adopted by OGC. Potential sources for inputs to such a model include the ISO 19109 Geographic Information – Rules for Application Schema and the conceptual model implied in the Geography Markup Language.

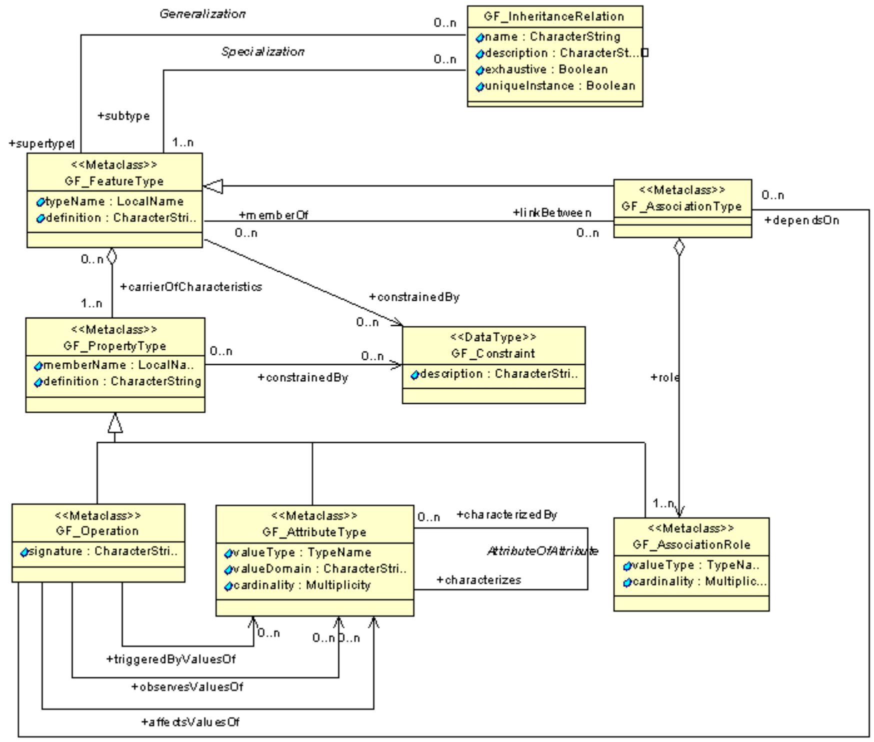



**Figure 5 - Kernel of the General Feature Model**

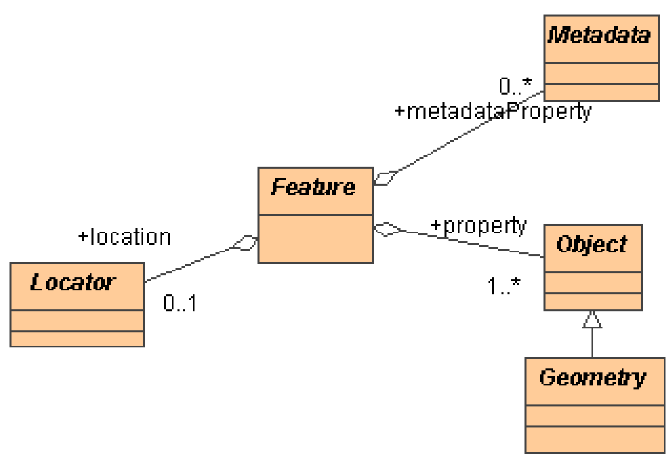



**Figure 6 - General Feature Model**[2]

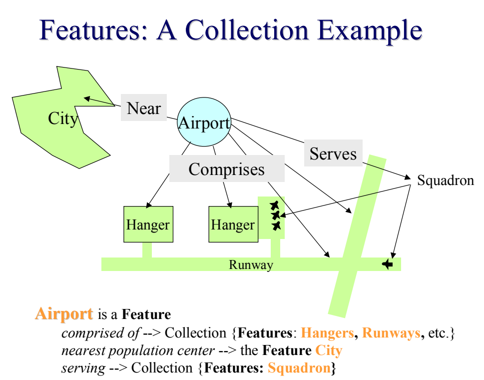
---

2 Figure needs to be updated based on GML3 and ISO 19109.

#### 3.1.2 Characteristics of Feature

The basic characteristics of *Features* are as follows:

- Within an information community or enterprise, there should be only one *Feature* per real-world entity. The granularity is user-determined. By 'one feature' we mean that there is a central object in a feature database that can handle all the responsibility of cascading information about a given 'real world entity', including updates, creation or deletion of attribution, metadata, and history. By 'real world entity' we mean any thing or phenomena about which the data store has a reason to capture and maintain information.
- *Features* have a unique, persistent ID, which may not be human-settable. This ID can be used to locate the *Feature*.
- *Features* are not classed, but they have Product View, i.e., application-oriented views that are classed. Real world entities do not fall into neat categories, and are often unique, through the auspices of human inventiveness. Specific applications, such as low-level flight, will require that these entities/objects assume the characteristics of particular application-oriented Product Views, such as a vertical obstruction. This means that somewhere (possibly distributed) there is a set of Product Views of multiple classes associated with this Feature, each with named attributes for what it means to be in their respective classes of application objects.
- *Features* have descriptions, which themselves have no uniqueness constraints.
- They are comprised of one or more attributes (properties) with name, type, value, and [optional description]. Attributes can be null. (More on Attributes, below.) The semantics of an attribute require that it must be uniquely identified by its name. One should be able to ask for an attribute by name and discover the following:
- Data type [e.g., integer, real, geometry, raster, references, etc.]. The value returned might be the root value of the attribute, or a derived value, depending on the access privileges of the user making the query.
- Collection type (applicable to *Features* that are Collections). For complex features represented by Collections (see Collections, below), the components of one Feature may be Features in their own right (see figure). Features that are Collections will always have one or more child Features. Parent Features contain References to their children. In the figure, the parent Feature *Airport* <u>comprises</u> Feature children that are *Hangers* and *Runways*, <u>serves</u> Feature children in the *Squadron*, and is <u>near</u> the Feature child that is a population center called *City*. The terms "<u>comprises</u>, <u>serves</u> and <u>near"</u> are examples of Collection types (i.e., types of object relationships that pertain to the Collection).



- Description – semantic description of the attribute
- Security level
- Accuracy
- Authority/source
- Currency (date/time).
- Type-dependent metadata about an attribute (e.g., Coordinate Reference System for the geometry)
- "Full content" consisting of an array of tuples: <value, accuracy, authority (source), timestamp [, security class]>. Need to tolerate attribute values that don't agree, but are both correct, because they apply in different contexts, e.g. when the height of a smokestack varies depending upon usage. This is the situation when objects/attributes reside across multiple systems.
- A *Feature* must always record the most accurate or most detailed value of each attribute (**root value**), and is responsible (albeit indirectly) for deriving the application-specific versions of each attribute for each of its Product Views. Any pair of Product Views for a *Feature* that have the same attribute, must have consistent values. In general, we envision that attribute duplication/replication is kept to a minimum between a *Feature* and its associated Product Views.

#### 3.1.2.1 (Feature) Collection

A collection is a special category of feature that represents a collection of features that have common metadata and formal relationships. Collections possess all the characteristics of a feature, i.e., they are complex features. Collections always contain feature references. I'm not sure how to fix this without changing the meaning, but defining a *collection* in the context of being a category of feature by saying "..a collection of features" isn't a clear way to convey its meaning.

#### 3.1.2.2 Event (Feature)

An event is a special category of feature or collection with an enumerated temporal domain.

### 3.1.3 Continuous and Discrete Features

Geographic phenomena fall into two broad categories, discrete and continuous. Discrete phenomena are recognizable objects that have relatively well-defined boundaries or spatial extent, examples include buildings, streams, and measurement stations. Continuous phenomena vary over space and have no specific extent. Examples include temperature, soil composition, and elevation. A value or description of a continuous phenomenon is only meaningful at a particular position in space (and possibly time). Temperature, for example, takes on specific values only at defined locations, whether measured or interpolated from other locations.

These concepts are not mutually exclusive. In fact, many components of the landscape may be viewed alternatively as discrete or continuous. For example, a stream is a discrete entity, but its flow rate and water quality index vary from one position to another. Similarly, a highway can be thought of as a feature or as a collection of observations measuring accidents or traffic flow, and an agricultural field is both a spatial object and a set of measurements of crop yield through time.

Historically, geographic information has been treated in terms of two fundamental types called **vector** data and **raster** data.

"Vector data" deals with discrete phenomena, each of which is conceived of as a feature. The spatial characteristics of a discrete real world phenomenon are represented by a set of one or more geometric primitives (points, curves, surfaces, or solids). Other characteristics of the phenomenon are recorded as feature attributes. Usually, a single feature is associated with a single set of attribute values.

"Raster data", on the other hand, deals with real world phenomena that vary continuously over space. It contains a set of values, each associated with one of the elements in a regular array of points or cells. It is usually associated with a method for interpolating values at spatial positions between the points or within the cells. Since this data structure is not the only one that can be used to represent phenomena that vary continuously over space, OGC uses the term "**coverage**" to refer to any data representation that assigns values directly to spatial position. A coverage is a function from a spatiotemporal domain to an attribute domain. A coverage associates a position within a spatiotemporal domain to a record of values of defined data types.

### 3.2 Spatial-temporal geometry and topology

Standardized conceptual schemas for spatial and temporal characteristics increase the ability to share geographic information among applications. These schemas are used by geographic information system and software developers and users of geographic information to provide consistently understandable spatial data structures.

Geometry provides the means for the quantitative description, by means of coordinates and mathematical functions, of the spatial characteristics of features, including dimension, position, size, shape, and orientation. The mathematical functions used for describing the geometry of an object depend on the type of coordinate reference system used to define the spatial position. Geometry is the only aspect of geographic information that changes when the information is transformed from one geodetic reference system or coordinate system to another.

Topology deals with the characteristics of geometric figures that remain invariant if the space is deformed elastically and continuously – for example, when geographic data is transformed from one coordinate system to another. Within the context of geographic information, topology is commonly used to describe the connectivity of an n-dimensional graph, a property that is invariant under continuous transformation of the graph. Computational topology provides information about the connectivity of geometric primitives that can be derived from the underlying geometry.

This section provides an overview and references to the conceptual schema for geometry, topology and temporal reference systems.

#### 3.2.1 Geometry

A **geometric object** is a combination of a coordinate geometry and a coordinate reference system. In general, a geometric object is a set of geometric points, represented by **direct positions**. A direct position holds the coordinates for a position within some coordinate reference system.

Figure 7 shows the basic classes defined in the conceptual geometry schema. Any object that inherits the semantics of the GM_Object acts as a set of direct positions. Objects under GM_Primitive will be open, that is, they will not contain their boundary points; curves will not contain their end points, surfaces will not contain their boundary curves, and solids will not contain their bounding surfaces. Objects under GM_Complex will be closed, that is, they will contain their boundary points.

**BASELINE**: The conceptual schema for geometry has been adopted in OpenGIS Abstract Specification Topic 1 - Feature Geometry. Topic 1 is identical with ISO 19107, “Geographic information — Spatial schema.”

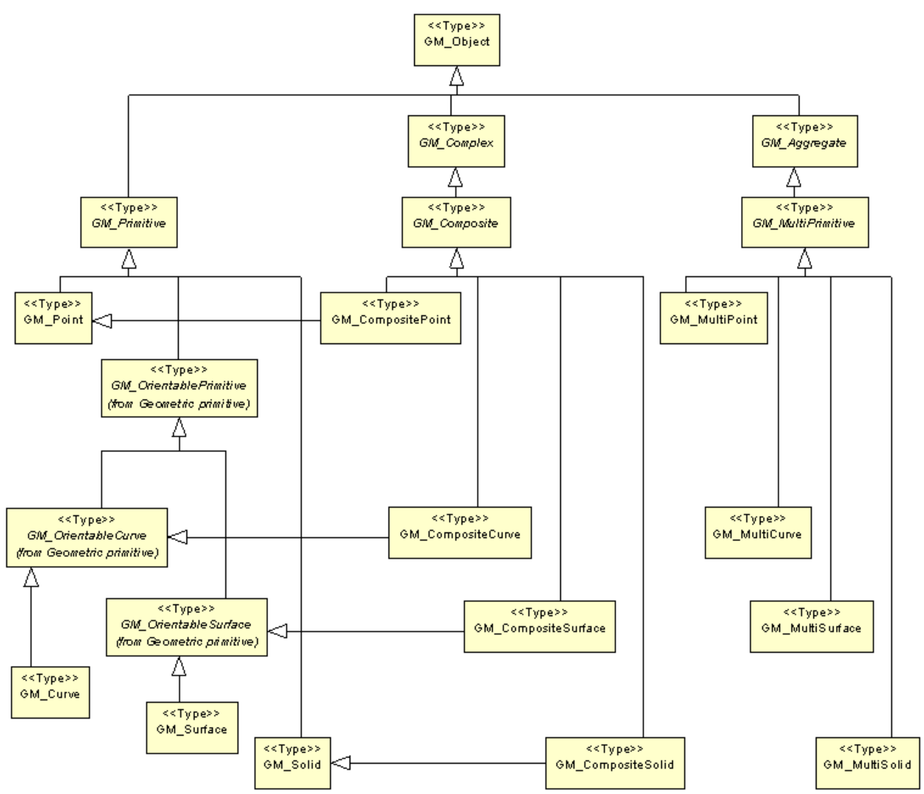



**Figure 7 - Geometry Conceptual Schema; basic classes**

### 3.2.2 Topology

The most productive use of topology is to accelerate computational geometry. Geometric calculations such as containment (point-in-polygon), adjacency, boundary, and network tracking are computationally intensive. For this reason, combinatorial structures known as topological complexes are constructed to convert computational geometry algorithms into combinatorial algorithms. Another purpose is, within the geographic information domain, to relate feature instances independently of their geometry.

A **topological complex** consists of collections of **topological primitives** of all kinds up to the dimension of the complex. Thus, a 2-dimensional complex must contain faces, edges, and nodes, while a 1-dimensional complex or graph contains only edges and nodes. Figure 8 gives an overview of the class structure of the basic topological packages. The root class of the diagram is TP_Object. Under this, there are TP_Primitive, and TP_Complex.

**BASELINE** The conceptual schema for topology has been adopted in OpenGIS Abstract Specification, Topic 1 - Feature Geometry. Topic 1 is identical with ISO 19107, "Geographic information — Spatial schema."

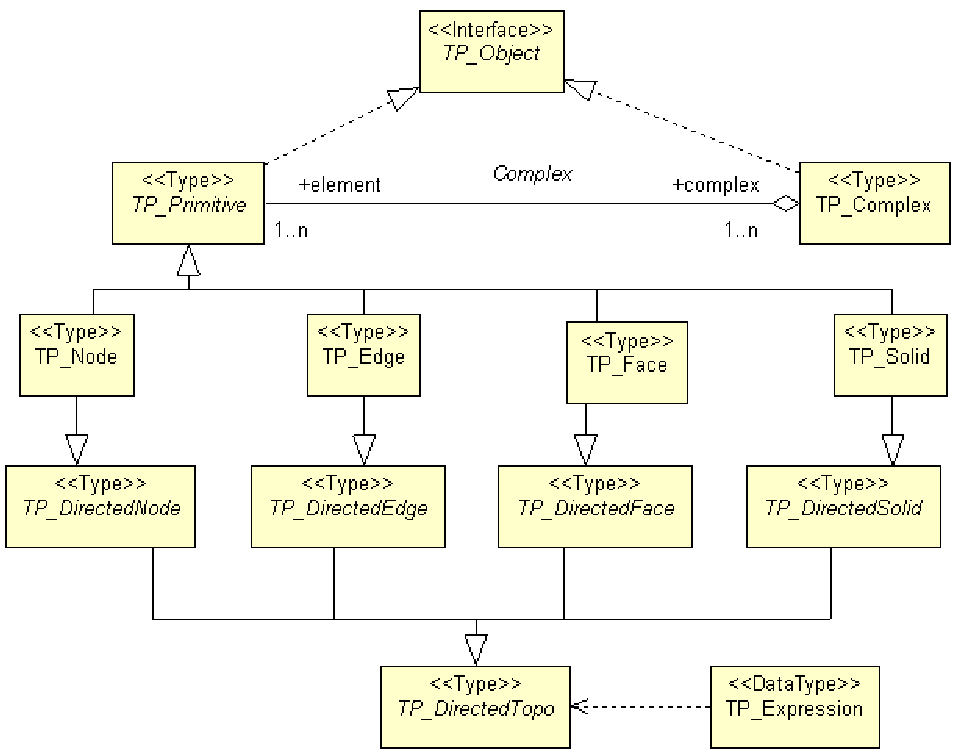



**Figure 8 - Topological Class Diagram**

### 3.2.3 Query Operators

Query operators are a mechanism for characterizing topological relations between different features. The operators are meant mainly for query evaluation and are defined in such a manner as to allow a variety of implementations to be assured of equivalent results against datasets with equivalent information content. Typical names for these query operators include “contains,” "intersects," and "equals" operations (See Figure 9).

BASELINE The conceptual schema for query operators has been adopted in OpenGIS Abstract Specification, Topic 1 - Feature Geometry. Topic 1 is identical with ISO 19107, “Geographic information — Spatial schema.”

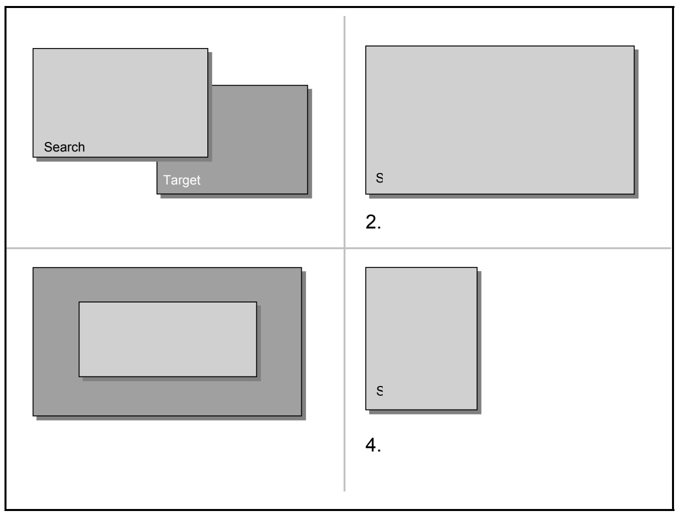



**Figure 9 - Query operator examples**

### 3.2.4 Temporal

A conceptual temporal schema defines the concepts needed to describe the temporal characteristics of geographic information as they are abstracted from the real world. Temporal characteristics of geographic information include feature attributes, feature operations, feature associations, and metadata elements that take a value in the temporal domain.

A temporal schema contains two packages. The **temporal objects** package defines objects for use as values for the temporal characteristics of features and data sets, e.g., instant and period. The temporal position of an object is specified in relation to a temporal reference system. The **temporal reference system** package provides elements for describing temporal reference systems, e.g., calendars and clocks

A value in the time domain is a temporal position measured relative to a temporal reference system. ISO 8601 specifies the use of the Gregorian calendar and 24 hour local or Coordinated Universal Time (UTC) for information interchange. This is the primary temporal reference system for use with geographic information. A different temporal reference system may be appropriate for some applications of geographic information.

BASELINE: A conceptual temporal schema has not been adopted by OGC. Potential sources for inputs to such a model include the ISO 19108, Geographic information — Temporal. ISO and the conceptual model implied in the Geography Markup Language. Both ISO 19108 and GML utilize ISO 8601, Data elements and interchange formats - Information interchange - Representation of dates and times

#### 3.2.5 Spatio-temporal Schema

Historically, temporal characteristics of features have been treated as thematic feature attributes. For example, a feature "Building" may have an attribute "date of construction". However, there is increasing interest in describing the behavior of features as a function of time. This can be supported to a limited extent when time is treated independently of space. For example, the path followed by a moving object can be represented as a set of features called "way point", each of which is represented as a point and has an attribute that provides the time at which the object was at that spatial position.

Behavior in time may be described more easily if the temporal dimension is combined with the spatial dimensions, so that a feature can be represented as a spatiotemporal object. For example, the path of a moving object could be represented as a curve described by coordinates in x, y and t.

BASELINE: A spatio-temporal conceptual schema has yet to be developed. However, much of the foundation work for a spatio-temporal schema can be found in OGC Recommendation Paper titled "Observations and Measurements" (version 0.9.2, February 2003).

### 3.3 Spatial Referencing

This section addresses how spatial data is referenced, including the following topics:

- Terminology with spatial reference
- Coordinate reference systems
- Coordinate transformations
- Location organizer folder

#### 3.3.1 Terminology with spatial reference

Many terms refer to locations near the surface of the earth, e.g., identifiers and place names. Spatial referencing with identifiers is when the **identifier** uniquely indicates a location, a postal code. Place names may be ambiguous, e.g., Springfield, requiring additional information to be resolved into a specific location. Gazetteers and geocoding are used to resolve the ambiguity.

A **gazetteer** retrieves the geometries for one or more features, given their associated well-known feature identifiers (text strings). The identifiers are any words or terms that describe the features, which are well known to the gazetteer, such as a set of place names and/or landmarks. Each instance of a gazetteer has an associated vocabulary of identifiers. A gazetteer may apply to a given region, such as a country, or some other specialized grouping of features. A gazetteer may be thought of as a special case of a geocoder.

A **geocoder** transforms a description of a feature location, such as a place name, street address or postal code, into a normalized description of the location, which includes a coordinate geometry. A geocoder Service receives a description of a feature location as input and provides a normalized address with geometry as output. The feature location descriptions are any words, codes or terms that describe the features, and that are well-known to the Geocoder Service, such as a street addressing or postal coding scheme.

BASELINE: A conceptual model for terminology with spatial reference has not been adopted by OGC. Potential sources for such a model include ISO 19112 "Spatial referencing by geographic identifiers" and the conceptual model implied in the "Gazetteer Service Draft Candidate Implementation Specification 0.0.9" (OGC Document 02-076r3), the "Geocoder Service Draft Candidate Implementation Specification 0.7.6" (OGC Document 01-026r1) and the OpenGIS Location Services Interface Specification 0.0 (OGC Document 03-06r1). This later document has a full geocoder and reverse geocoder interface definition.

#### 3.3.2 Coordinate Reference Systems

Coordinates are unambiguous only when the coordinate reference system to which those coordinates are related has been fully defined. A **coordinate reference system** is a coordinate system that has a reference to the Earth. A coordinate reference system consists of a coordinate system and a datum. Types of coordinate reference systems include: geocentric, geographic (including an ellipsoid), projected, engineering, image, vertical, temporal

**Coordinates** are a tuple of ordered scalar values that define the position of a single point feature in a coordinate reference system. The tuple is composed of one, two or three "ordinates". The ordinates must be mutually independent and their number must be equal to the dimension of the coordinate space; for example, a tuple of coordinates may not contain two heights.

A **coordinate system** is composed of a set of coordinate axes with known metric. The concept "metric of a coordinate space" consists of the set of mathematical rules that defines the relationships between the coordinate values and the invariant spatial quantities between points; for example, the mathematical rules (formulae) required for calculating angles and distances between points from coordinate values and vice versa.

The **datum** defines the origin, orientation and scale of the coordinate system and ties it to the earth, ensuring that the abstract mathematical concept "coordinate system" can be applied to the practical problem of describing positions of features on or near the earth's surface by means of coordinates.

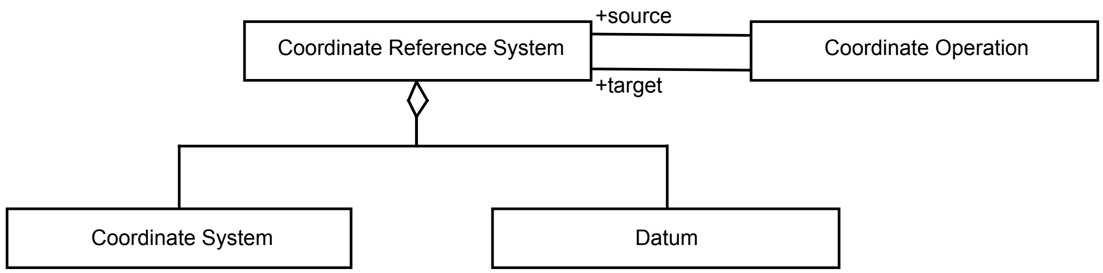



**Figure 10 - Coordinate Reference System**

BASELINE The conceptual schema for Coordinate Reference Systems has been adopted as OpenGIS Abstract Specification, Topic 2: Spatial Referencing by Coordinates (Version: 1.0.1, 2001-11-13, OGC 01-063r1). Topic 2 supplements and corrects ISO DIS 19111: Geographic Information – Spatial Referencing by Coordinates.

### 3.3.3 Coordinate Transformations

In addition to describing a coordinate reference system, OGC Topic 2 provides the description of a coordinate transformation or coordinate conversion between two different coordinate reference systems (See Figure 10). With such information, geographic data referred to different coordinate reference systems can be merged together for integrated manipulation.

**Coordinate conversion** is a mathematical operation on coordinates that does not include any change of Datum. The best-known example of a coordinate conversion is a map projection. The parameters describing coordinate conversions are defined rather than empirically derived.

**Coordinate transformation** is a mathematical operation on coordinates that includes a change of Datum. The parameters of a coordinate transformation are empirically derived from a dataset containing the coordinates of a series of points in both coordinate reference systems. This computational process is usually ‘overdetermined’, allowing derivation of error (or accuracy) estimates for the transformation. Also, the stochastic nature of the parameters may result in multiple (different) instantiations of the same coordinate transformation.

A **map projection** is a coordinate conversion from a geodetic coordinate system to a planar surface, converting geodetic latitude and longitude to plane (map) coordinates. The result is a two-dimensional coordinate system called a projected coordinate reference system.

BASELINE Prior to the adoption of the revised Topic 2 in December of 2001, OGC adopted the OpenGIS® Coordinate Transformation Services Implementation Specification. The CTS Implementation Specification was developed according to the prior version of Topic 2. Efforts to align Topic 6 and the CTS IS are underway. Recently, there has been considerable work by the CRS Working group to align all CRS references across all OGC specifications. Part of this work resulted in the OGC Recommendation Paper “Recommended XML Encoding of CRS” (OGC document 03-010r7, May 2003).

### 3.3.4 Location Organizer Folder (LOF)

Spatial referencing can be accomplished by aggregating information items that share a common location in space and time. OGC has defined the **Location Organizer Folder** (LOF) as a general, multi-source information container model for handling sets of inter-related spatio-temporal information. A LOF can contain such things as images, maps, features, cables, and any other information elements (e.g. audio, video, etc). LOFs also contain relationships (links) between any LOF information elements, or between LOF elements and external resources (e.g. a link to a Web site, a spatial query for a specific database, etc.).

BASELINE: Work was performed during the GeoSpatial Fusion Initiative that resulted in the OGC Discussion Paper titled "Location Organizer Folder" (OGC Document01-037, March 2001),

## 3.4 Coverages including imagery

### 3.4.1 Introduction

A coverage is a feature that associates positions within a bounded space (its spatiotemporal domain) to feature attribute values (its range) (See Figure 11). Examples include a raster image, a polygon overlay, or a digital elevation matrix. A coverage can represent one feature or a collection of features to model spatial relationships between, and the spatial distribution of, earth phenomena.

The spatio-temporal **domain** of a coverage is a set of geometric objects described in terms of direct positions (See Section 3.2). Commonly used spatio-temporal domains include point sets, grids, collections of closed rectangles, and other collections of geometric objects. The geometric objects may exhaustively partition the spatio-temporal domain, and thereby form a tessellation such as a grid or a TIN.

The **range** of a coverage is a set of feature attribute values. Coverages often model many associated functions sharing the same spatio-temporal domain. Therefore, the value set is represented as a collection of records with a common schema. The attributes of a coverage, i.e., its range, are homogeneous across its domain.

EXAMPLE A coverage might assign to each direct position in a county the temperature, pressure, humidity, and wind velocity at noon, today, at that point. The coverage maps every direct position in the county to a record of 4 fields.

BASELINE: The conceptual schema for Coverages has been adopted in OpenGIS Abstract Specification, Topic 6: The Coverage Type and its Subtypes. Topic 6 incorporates and extends ISO 19123 Geographic information — Schema for coverage geometry and functions. An update of Topic 6 incorporating ISO 19123 has been approved by the TC.

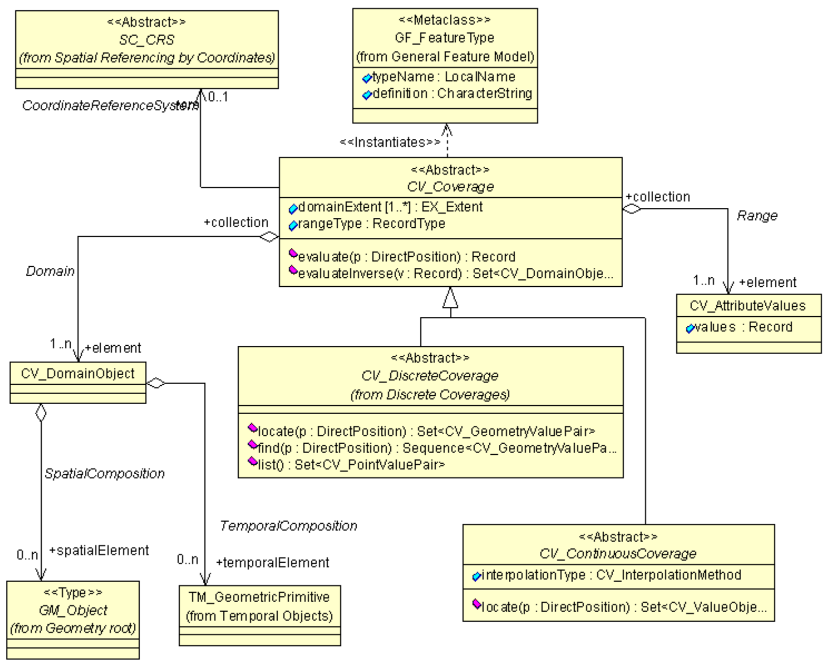



**Figure 11 - Coverage: Domain and Range**

### 3.4.2 Imagery – a type of Coverage

Imagery is a common way of collecting information associated with a coverage, by which the value of a continuous phenomenon is usually sampled at regular but discrete locations, i.e. pixels (See Figure 12).

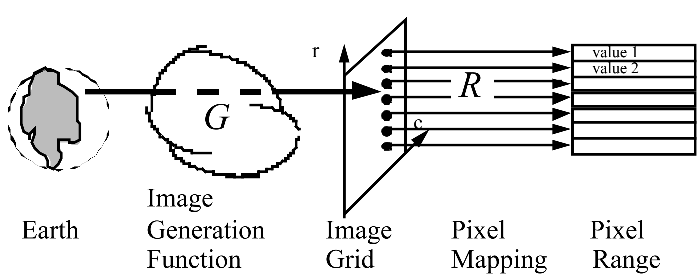



**Figure 12 - Earth Imagery**

Imagery as gridded data can be classified depending upon spatial and attribute properties.

Based on its spatial properties, gridded data can be classified into two subclasses, georectified and ungeorectified gridded data. Any cell in a **georectified** gridded data can be uniquely geolocated, given the cell spacing, grid origin and orientation. **Ungeorectified** gridded data are irregularly spaced in any geographic/map projection coordinate system. Therefore, the location of one cell in an ungeorectified gridded data cannot be determined based on another cell's location. Ungeorectified data can be georeferenced or ungeoreferenced subclasses, depending on whether information is provided with a data set that allows determination of the geolocation of a cell. One approach to rectifying imagery utilizes a **sensor description**.

Based on its attribute properties, gridded data can be classified into two subclasses: **imagery** and **thematic**. Imagery data is gridded data whose attribute values are a numerical representation of the physical parameter measured by an instrument from which the image is transmitted, for example calibrated radiance values. Thematic data has values of a geographic feature, for example, landcover classification.

**BASELINE** Imagery is defined conceptually in adopted OpenGIS Abstract Specification, Topic 7: The Earth Imagery Case and in the review summary for ISO 19124, Geographic information - Imagery and gridded data components. Several current projects in OGC are further developing conceptual models for earth imagery. A conceptual model for sensor descriptions is under development in ISO 19129, which is closely coordinated with OGC SensorML.

#### 3.4.3 Observations and Measurements

Many methods exist for structuring the domain portion of coverages, e.g., encoding methods for gridded data. Much less work has been done to structure the range side of a coverage function. An approach to coverage attributes regarding observations and measurements is under development in the OGC Interoperability Program.

A **measurement** is an observation event whose value property is a value of some natural phenomenon. A measurement usually refers to the measuring device and procedure used to determine the value, such as a sensor or observer, analytical procedure, simulation or other numerical process. In particular, a measurement feature binds the result to the (spatiotemporal) location where the measurement was made. An observed value may be semantically typed according to the subject or observable.

A measurement has a single sensor and observable, and a single result. The description of the sensor provides important metadata to support the interpretation of the result. The sensor may be a "sensor package" (such as a weather station, or multi-band radiance sensor) measuring an "aggregate observable" and producing a resulting "aggregate value" (the corresponding set of weather parameters, or a radiance spectrum). When associated with a measurement, however, the sensor, observable and result are each single logical entities.

It is through its association with a Measurement feature that a Value is bound to a feature of interest or a geospatial location, to a time instant or period, and to the `Sensor Instance` responsible for the measurement. The Measurement might be considered to be the *mapping* of a value and (spatiotemporal) location.

Measurements may be aggregated into a **collection** where certain information items may be used repetitively. Frequently, a single sensor is associated with Measurements in a temporal coverage or time-series. A spatial array of similar sensors sampled at a single instant or measuring a time-invariant phenomenon can produce a spatial coverage.

The name of the phenomenon being sampled or *observable* provides an important classifier for a measurement. This ensures that the results of measurements may be compared, on condition that the values are available in the same reference system.

**Baseline:** The current approved OGC position on observations and measurements is documented in the OGC Recommendation Paper "Observations and Measurements", OGC Document 03-022r3.

### 3.5 Portrayal and human interface

**Portrayal** is the presentation of information to humans, e.g., a map. A **map** is a two-dimensional visual portrayal of geospatial data; a map is not the data itself. Two or more maps with the same geographic extent and coordinate reference system can be accurately layered to produce a composite map. Information types associated with geospatial data visualization are shown in the context of the portrayal process (See Figure 13).

1. Image or picture of the data, e.g., a map to be displayed.
2. Display elements, e.g., lexical description of graphics to be drawn onto the target display.

**BASELINE** Currently there is no stand-alone conceptual model adopted by OGC for portrayal. Potential inputs to such a model include the OGC Web Map Service (WMS) Implementation Specification and the Portrayal Work done in the OGC OpenLS Interoperability initiative. In particular: WMS version 1.1.1, Section 4. Extending the portrayal model to include a perspective view could build on the OGC Web Terrain Service (WTS) discussion paper. Another supporting document in this area is the OpenGIS Style Layer Descriptor Implementation Specification, Version1.0.

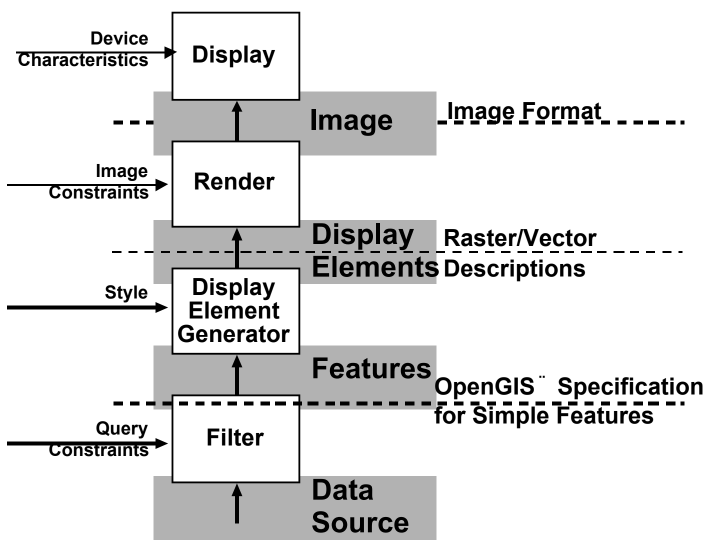



**Figure 13 - Portrayal model**

#### 3.5.1 Symbology

The creation of display elements requires two inputs; features and style (See Figure 13). Style for portrayal requires **symbology**, a methodology for describing symbols and mapping of the schema to an application schema.

Symbology needs to be present to portray a dataset containing geographic data. Symbology makes it possible to portray the same dataset in different ways without altering the dataset itself. There are two basic ways to style a data set. The simplest one is to color all features the same way, e.g., for a hydrology data set color the insides of all polygons in a light blue. This type of styling requires no knowledge of the attributes or "feature types" of the underlying data, only a language with which to describe these styles. A more complicated requirement is a feature-centered rule of styling features of the data differently depending on some attribute. For example, in a roads data set, style highways with a three-pixel red line; and style four-lane roads in a two-pixel black line.

**BASELINE** Currently there is no conceptual model adopted by OGC for symbology. Potential inputs to such a model include ISO 19117 "Geographic information — Portrayal" and the conceptual model implied in the "OpenGIS Styled Layer Descriptor Implementation Specification, Version 1.0".

#### 3.5.2 Human Interaction with Geospatial Information

Maps are a significant method for human interaction with geospatial information, but there are many areas that need further development. Challenges regarding human interaction with geospatial information include:

- Harnessing geospatial information volume and complexity; including 3-D visualization and immersion, scientific visualization.
- Geospatial for everyone; universal access and usability
- Geospatial everywhere; mobile information, mobile enhancement of human perception
- Collaborative Work with GI

### 3.6 Metadata

Metadata is data about data. This section addresses dataset metadata, service metadata and the registry information model.

#### 3.6.1 Data set metadata

Metadata elements and schema are used by data producers to characterize their geographic data. Metadata enables the use of geographic data in the most efficient way by knowing its basic characteristics. Metadata facilitates data discovery, retrieval and reuse. Metadata enable users to determine whether geographic data in a holding? will be of use to them.

Metadata is applicable to independent datasets, aggregations of datasets, individual geographic features, and the various classes of objects that compose a feature.

The conceptual metadata schema is intended to be used by information system analysts, program planners, and developers of geographic information systems, as well as others in order to understand the basic principles and the overall requirements for standardization of geographic information.

Metadata entity set information consists of the entity (UML class) MD_Metadata (see Figure 14). The MD_Metadata entity contains both mandatory and optional metadata elements (UML attributes). The MD_Metadata entity is an aggregate of entities shown in Figure 14.

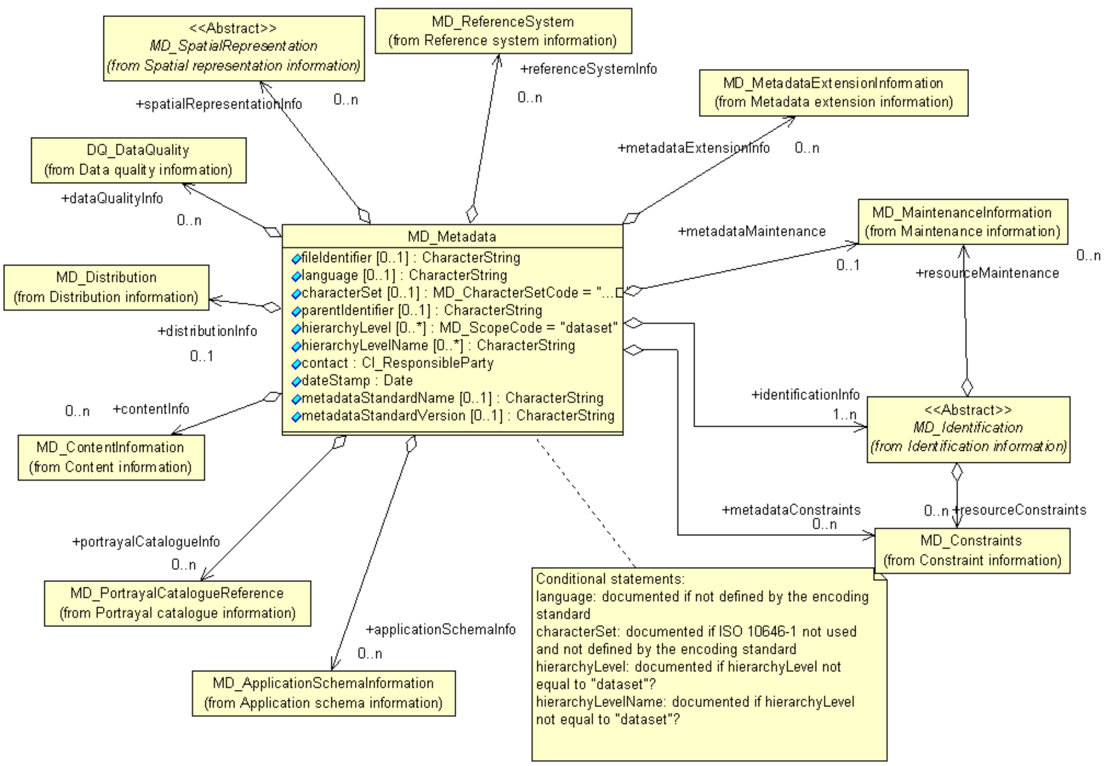



**Figure 14 - Dataset Metadata**

The conceptual schema for dataset metadata defines an extensive set of metadata elements; typically only a subset of the full number of elements is used. A subset of the elements known as the **core metadata elements** required to identify a dataset is defined, and typically used for catalogue purposes.

**BASELINE** The abstract specification for metadata has been adopted as ISO 19115, Geographic Information – Metadata. ISO 19115 was adopted as a replacement for OGC Abstract Specification Topics 9 and 11. In June 2001, a motion to include material in addition to ISO 19115 was adopted as document "01-111 Metadata AS." FGDC in conjunction with ANSI INCITS L1 are planning the migration of the FGDC Content Standard for Geospatial Metadata to be a profile of ISO 19115.

#### 3.6.2 Service Metadata

The most basic operation all OGC services must provide is the ability to describe themselves. This "Get Capabilities" operation is common to all OWS1 services. The result of invoking this operation on a service is a message containing a "capabilities document" describing the service. A **capabilities document**, or service profile, provides a high-level description of a service instance and its provider. Service profiles consist of three types of information: a human readable description of the service, a specification of the functionalities that are provided by the service and a set of functional attributes that provide additional information and requirements about the service that assist when reasoning, for example, about several services with similar capabilities.

**BASELINE** The conceptual schema for metadata for services has been adopted as OpenGIS Abstract Specification, Topic 12: OpenGIS Service Architecture. Topic 12 incorporates ISO 19115 Geographic information — Services. Further work on service metadata is being conducted in the OGC Web Services initiative (see OWS1 Web Services Architecture OGC document 02-022r1, Version: 0.2, 15 April 2002).

#### 3.6.3 Registry Information Model

Dataset metadata and service metadata are examples of a more general class of information. Adopting a broader perspective, all metadata and data types are regarded as **registry objects**. Every registered resource is a registry object. The registry information model is a conceptual metamodel that specifies the structure of metadata within a registry who's main purpose is to provide a formal structure representing registry objects and their interrelationships. The importance of the metamodel lies in its specification of a common model for understanding, sharing, and reusing of the contents of registry implementations.

The OGC registry information model (ogcRIM) is based on the ebXML registry information model (ebRIM, v 2.1), and thus makes the following assumptions: 1) all access to registry content is performed through the interfaces defined for the Registry Service; 2) the information model provides a basis for interoperability by constraining the behavior of the registry (i.e. it helps the implementer to understand what to build by defining the kind of metadata being manipulated and how they are interrelated); 3) the registry makes use of a repository for storing and retrieving persistent information and shared resources; but no assumptions are made concerning the exact nature of the repository or its location.

**BASELINE** A draft registry Information model was developed – in the OGC Web Services Interoperability Initiatives.

### 3.7 Service Information Model

OGC Web Based services 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.

A number of current limitations in the use of GetCapabilities 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.

The current work 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.

**BASELINE: OGC Discussion Paper "OWS 1.2 Service Information Model" (OGC Document 03-026, January 2003).**

### 3.8 Information communities

An **information community** is a collection of people (a government agency or group of agencies, a profession, a group of researchers in the same discipline, corporate partners cooperating on a project, etc.) who, at least part of the time, share a common digital geographic information language and common spatial feature definitions. This implies a common worldview as well as common abstractions, feature representations, and metadata. The feature collections that conform to the information community's standard language, definitions, and representations belong to that information community.

Information communities develop application schema, product specifications and framework data themes as defined in this section.

#### 3.8.1 Application Schema

An application schema provides the formal description of the data structure and content required by one or more information communities. Any description is always an abstraction, always partial, and always just one of many possible views. An application schema contains the descriptions of both geographic data and other related data.

An **application schema** is a set of conceptual schema for data required by one or more applications. An application schema contains selected parts of the base schemas presented in the prior sections of the ORM Information Viewpoint. Designers of application schemas may extend or restrict the types defined in the base schemas to define appropriate types for an application domain.

The purpose of an application schema is twofold:

— to provide a computer-readable data description defining the data structure, which makes it possible to apply automated mechanisms for data management;

— to achieve a common and correct understanding of the data, by documenting the data content of the particular application field, thereby making it possible to unambiguously retrieve information from the data.

**BASELINE** A conceptual model for application schemas has not been adopted by OGC. A source for such a model could be ISO 19110, "Geographic information - Feature cataloguing methodology."

#### 3.8.2 Feature catalogue

The application schema contains instances of types defined in the general feature model. For example, the general feature model defines the concept of feature type while an application schema defines specific feature types such as Road or Lake. Feature types are defined for an information community in a feature catalogue.

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

**BASELINE** Currently there is no conceptual model adopted by OGC for feature cataloguing. Potential inputs to such a model include ISO 19110, Geographic information - Feature cataloguing methodology.

An example of a feature catalogue is the Spatial Data Standard for Facilities, Infrastructure, and Environment (SDSFIE), standardized by ANSI as INCITS 353. NCITS 353 defines a catalog of geographic features supporting comprehensive master planning, environmental planning, and site planning, engineering, and lifecycle maintenance for facilities/installations, infrastructure, and environmental applications.

#### 3.8.3 Product specifications

A product specification is a description of an information community and the specification for mapping the information community to a dataset. It contains normative references to specific clauses in the base standards required to completely specify a data set. A product specification can be considered to document an application schema.

A product specification should specify the application and spatial schema, the metadata, quality information, reference system, feature coding catalogue (if applicable) and encoding (for transfer data sets). A product specification specifies a narrow class of instances of data sets. Some optionality may be permitted, but the range of optionality should be limited.

**BASELINE** Currently there is no conceptual model adopted by OGC for product specifications. A project is underway in ISO TC211 to define a product specification methodology.

#### 3.8.4 NSDI Framework Datasets

Product specifications for the National Spatial Data Infrastructure (NSDI) are under development by ANSI and FGDC. Framework data themes are to be standardized in seven areas: cadastral, digital ortho imagery, elevation bathymetric, geodetic control, governmental units, hydrography and transportation.

A Framework data content standard includes a conceptual schema expressed in UML. For a NSDI theme, the conceptual schema will specify, as appropriate, the feature types, attribute types, attribute domain, feature relationships, spatial representation, data organization, and metadata that define the information content of a data set. A Framework data content standard includes a format-independent specification of a conceptual model that can be implemented with one or more logical and physical models. The standards will include implementation profiles derived from the conceptual schema as annexes.

# 4 Computational Viewpoint: OpenGIS Service Framework

The computational viewpoint is concerned with the functional decomposition of the system into a set of services that interact at interfaces. This viewpoint captures the details of these components and interfaces without regard to distribution. Accordingly, this section first defines the core concepts of services, interfaces and operations (and the relationships amongst these concepts), then describes the Publish/Find/Bind pattern which represents the interactions among OGC services. The computational viewpoint of the ORM also includes examples of service classification and a description of the OGC Service Framework. Example service applications where services are combined or chained in support of decision-making are included at the end of this section.

## 4.1 Services, Interfaces and Operations

Key definitions for the Service Framework are:

- A **Service** as a distinct part of the functionality that is provided by an entity through interfaces,
- An **Interface** as a named set of operations that characterize the behavior of an entity,
- An **Operation** as a specification of a transformation or query that an object may be called to execute. Each operation has a name and a list of parameters.

The terms are related to each other as depicted in Figure 15.

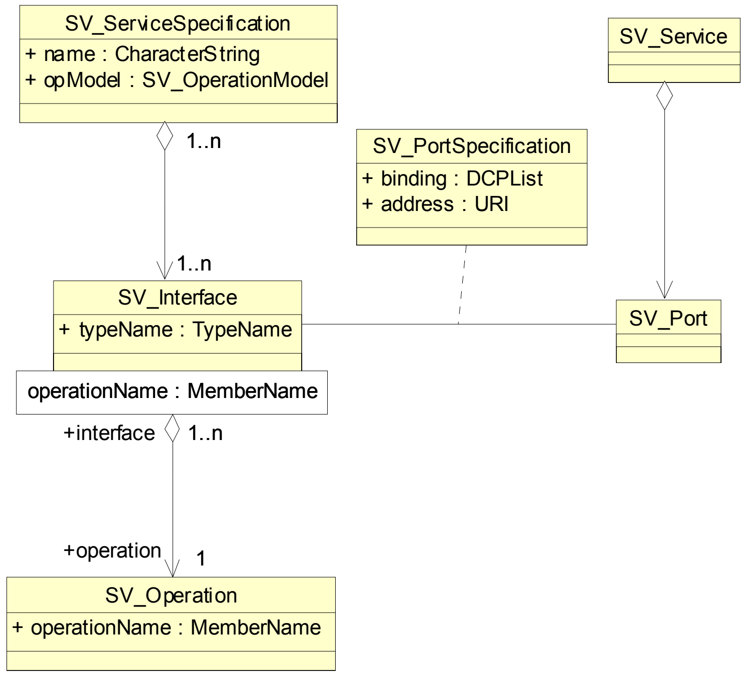



**Figure 15 - Service definition relationships.**

A service may be expressed at various levels of granularity. A coarse-grained collaboration may be refined to produce a service that has a finer granularity. This is accomplished by expanding one or more operations from a high level collaboration into distinct lower level services, one for each operation.

An instance of a service may be associated with a specific instance of a dataset, or it may be a service that can be used to operate on multiple, unspecified datasets. The first case is referred to as a **tightly coupled data and service**. The second case is referred to as **loosely coupled service**. Service operations can be associated with data classes (data type) or with instances (data set).

**BASELINE**: A computational model for service definitions has been adopted in OpenGIS Abstract Specification, Topic 12 - OpenGIS Service Architecture. Topic 12 is identical with ISO 19119, Geographic information — Services.

### 4.2 Publish, Find and Bind Pattern

The architecture is based on the publish/find/bind pattern shown in Figure 16, and supports the dynamic binding between service providers and requestors since sites and applications are frequently changing in a distributed environment.

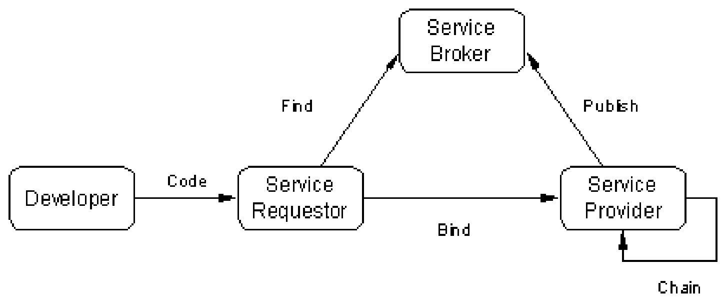



**Figure 16 - Publish/Find/Bind Pattern**

In Figure 16, there are three essential roles:

- **Service provider**: publishes services to a broker (registry) and delivers services to service requestors.

- **Service requestor**: performs service discovery operations on the service broker to find the service providers it needs and then accesses service providers for provision of the desired service.

- **Service broker**: helps service providers and service requestors to find each other by acting as a registry or clearinghouse of services.

As shown, there are three essential kinds of operations performed by services:

- **Publish**: used to advertise data and services to a broker (such as registry, catalog or clearinghouse). A service provider contacts the service broker to publish (or unpublish) a service. A service provider typically publishes to the broker metadata describing its capabilities and network address.

- **Find**: used by service requestors to locate specific service types or instances. Service requestors describe the kinds of services they're looking for to the broker and the broker responds by delivering the results that match the request. Service requestors typically use metadata published to the broker to find service providers of interest.

- **Bind**: used when a service requestor and a service provider negotiate, as appropriate, so the requestor can access and invoke services of the provider. A service requestor typically uses service metadata provided by the broker to bind to a service provider. The service requestor can either use a proxy generator to generate the code that can bind to the service, or can use the service description to manually implement the binding before accessing that service.

Figure 16 also shows that services can be chained with various degrees of transparency to achieve larger tasks required by a service requestor.

**BASELINE** A publish-find-bind pattern is under development in the OGC Web Services Interoperability Initiative. The pattern follows the ODP Trading function [32] .

### 4.3 Service Type Examples

A type is used to specify a characteristic of an entity. That entity can be a service, an interface, a binding, a parameter in the interface signature, or a relation. An entity is of a particular type if its properties satisfy that type. Types must match before some action can occur. Therefore, type-matching is almost always a precondition for some action to occur. Entities, such as a service, can assume multiple types at several levels of abstraction.

A service type is at least an interface signature type, but it can also include a set of service properties that contain information about computational aspects (such as the content, behavior and environment of the service) as well as describing the technology, engineering, information, and enterprise aspects of the service (such as defined by a service taxonomy). The next section summarizes three identified service classification schemes (semantic, interface and capability classification schemes), followed by a description of the OGC service framework.

#### 4.3.1 Service Classification Schemes

Three schemes for classification of services are provided below:

- semantic classification scheme (using a service taxonomy),
- interface classification scheme (using the set of operations at an interface),
- capability classification scheme (using associated service metadata schema about the provider of the service, its content, content type or other property types that provide additional semantic details).

##### 4.3.1.1 Semantic Classification Scheme

Using a semantic classification scheme, each service type is identified by a unique name from some classification scheme. This hierarchy provides at least part of the basis for deciding if a service of one type may be substituted for a service of another type. Classification schemes—or taxonomies— are used to facilitate discovery such as this enabling browsing by category (i.e. “yellow pages” functionality). The OGC service taxonomy groups services that are semantically similar, so as to facilitate browsing and discovery according to desired functionality (See Table 2). The OGC service taxonomy is a specialization of the ISO Open Systems Environment taxonomy of service types.

**Table 2 – OGC Service Taxonomy - Top-level categories**

| Service category | Description |
|---|---|
| Human Interaction | Services for managing user interfaces, graphics, multimedia, and presenting compound documents. |
| Information Management | Services for managing the development, manipulation, and storage of metadata, conceptual schemas, and datasets. |
| Workflow | Services that support specific tasks or work-related activities. |
| Processing | Services that perform large-scale computations; a processing service does not include capabilities for providing persistent storage of data or transfer of data over networks.<br><br>Sub-categories of processing:<br>- Geographic processing services – spatial<br>- Geographic processing services – thematic<br>- Geographic processing services – temporal<br>- Geographic processing services – metadata |
| Communication | Services that encode and transfer data across networks. |
| System Management | Services for managing system components, applications, and networks (including access control). |

BASELINE: The OGC service taxonomy has been adopted in OpenGIS Abstract Specification, Topic 12 - OpenGIS Service Architecture. Topic 12 is identical with ISO 19119, Geographic information — Services.

#### 4.3.1.2 Interface Classification Scheme

Each service type is identified by the set of operations and the signature (e.g., inputs, outputs, exceptions) of each operation in the interface. Services specialize? one another through extension (or restriction) of their interfaces and the operations of the interfaces and the signature (e.g., inputs, outputs, exceptions) of individual operations. Services organized according to an interface inheritance hierarchy can provide at least part of the basis for deciding if a service of one type may be composed with another to conduct some interaction.

Figure 17 illustrates the basic service model for inheritance hierarchies of service by interfaces and operations.

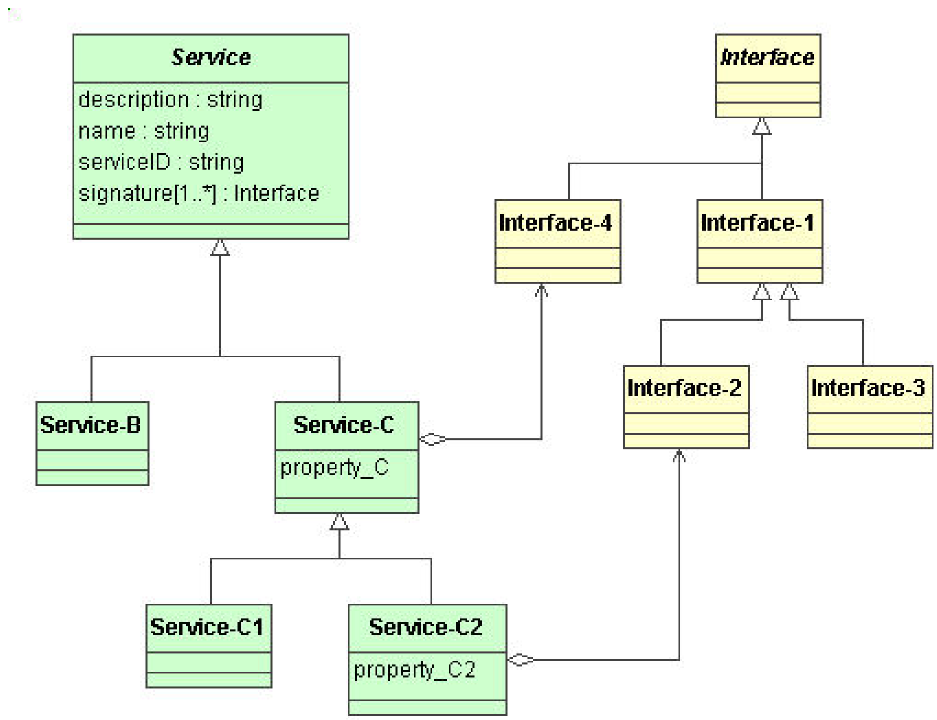



**Figure 17 - Service and Interface Types**

BASELINE A interface inheritance scheme is under development in the OGC Interoperability Program. The pattern follows the ODP Trading function [32] .

#### 4.3.1.3 Capability Classification Scheme

Service types may also be characterized by the kind of content they operate on (e.g., features, coverages, pictures), content categories (e.g., addresses, streets, parcels, etc) as well as other metadata such as descriptions of the service provider, region of validity, temporal range (epoch) of validity, types of data content, sources of content, quality of service, etc. The schema used to describe the capabilities and especially the content of a service could, when taken as a whole, be said to define the service type.

BASELINE A capability classification scheme is under development in the OGC Interoperability Program. It is important to note that OGC services appears to be rather unique among Web Services initiatives in emphasizing the importance of service content and content type as distinct from the services and service operations themselves.

### 4.3.2 OWS Service Framework

The OWS Service Framework (OSF) identifies services, interfaces and exchange protocols that can be utilized by any application (See Figure 18). OpenGIS Services are implementations of services that conform to OpenGIS Implementation Specifications. Compliant applications, called OpenGIS Applications, can then "plug into" the framework to join the operational environment.

By building applications to common interfaces, each application can be built without a-priori or run-time dependencies on other applications or services. Applications and services can be added, modified, or replaced without impacting other applications. In addition, operational workflows can be changed on-the-fly, allowing rapid response to time-critical situations. This loosely coupled, standards-based approach to development results in very agile systems—systems that can be flexibly adapted to changing requirements and technologies

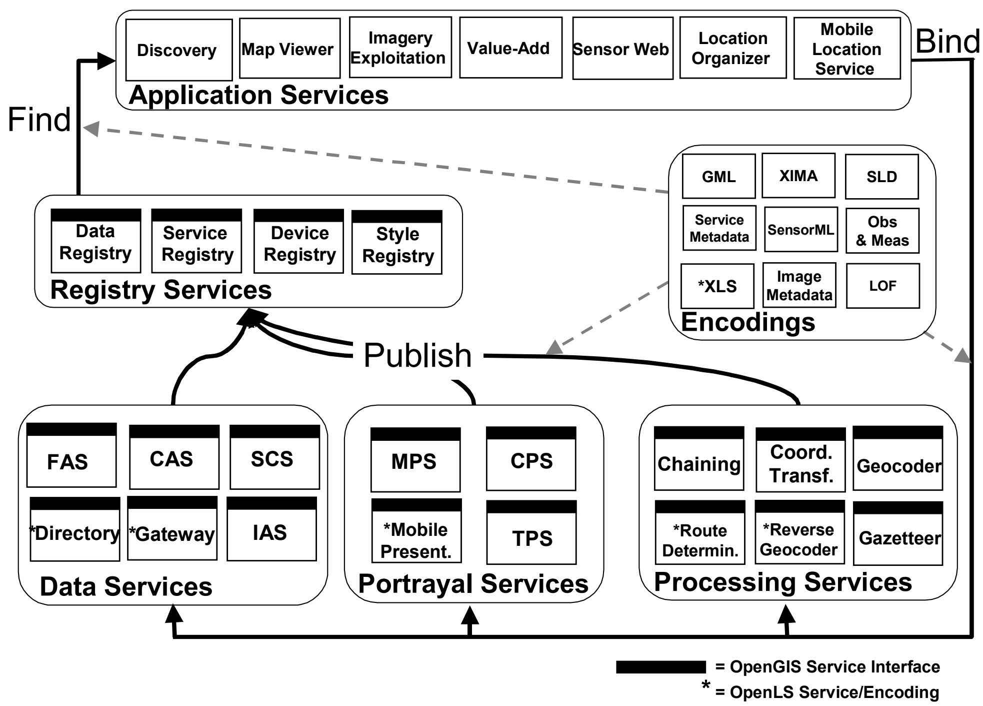



**Figure 18 - The OWS Service Framework (OSF)**

The OSF is designed to meet the following purposes:

- Provide a framework for coordinated development of new and extended services
- Enable interoperable services through standard interfaces and encodings
- Support publishing, discovery and binding of services through service metadata
- Allow separation of data instances from service instances
- Enable use of a provider's service on another provider's data
- Define a framework that can be implemented in multiple ways

The OSF is a profile of the OGC services taxonomy (See Section 4.3.1.1). The OSF categorizes services into five categories that correspond to the OGC services taxonomy top-level domains as shown in Table 3.

**Table 3 - OSF and ISO 19119 Service Categories**

| OSF Service Categories | OGC Service Taxonomy Categories |
|---|---|
| Application Services | Geographic Human Interaction |
| Registry Services | Geographic Information Management |
| Data Services | Geographic Information Management |
| Portrayal Services | Geographic Human Interaction |
| Processing Services | Geographic Processing Interaction |

BASELINE The OWS Service Framework is under development in the OGC Interoperability Program. See in particular the OWS1 Web Services Architecture. Does the OSF apply to OLS?

#### 4.3.2.1 Application Services

OSF services are accessible from Application Services operating on user terminals (e.g., desktop, notebook, handset, etc.) or servers that have network connectivity. Users may use Application Services to access Registry, Portrayal, Processing and Data Services, depending upon the requirements and designed implementation of the application. Application Services commonly, but not necessarily, provide user-oriented displays of geospatial content and support user interaction at the user terminal. Examples of OSF applications services include:

- Discovery Application Services
- Map Viewer Application Services
- Value-Add Application Services
- Imagery Exploitation Application Services
- Sensor Web Application Services
- Location Organizer (LO) Application Services
- Mobile Location Services

#### 4.3.2.2 Registry Services

OSF registry services provide a common mechanism to classify, register, describe, search, maintain and access information about resources available on a network. Resources are network addressable instances of typed data or services. Types of registries are differentiated by their roles; registries for cataloging data types, online data instances, service types and online service instances are a few examples. ?

#### 4.3.2.3 Processing Services

OSF processing services operate on geospatial data and provide “value-add” services for applications. They can transform, combine, or create data. Processing Services can be tightly or loosely coupled with other services, such as Data and Portrayal Services. Processing Services can be sequenced into a “value-chain” of services to perform specialized processing in support of information production workflows and decision support. Examples of OSF processing services include:

- Chaining Services
- Coordinate Transformation Services (CTS)
- Geocoder Services
- Gazetteer Services
- Geoparser Services
- Reverse Geocoder Services
- Route Determination Services

#### 4.3.2.4 Portrayal Services

OSF portrayal services provide visualization of geospatial information. Portrayal Services are components that, given one or more inputs, produce rendered outputs (e.g., cartographically portrayed maps, perspective views of terrain, annotated images, views of dynamically changing features in space and time, etc.). Portrayal Services can be tightly or loosely coupled with other services such as Data and Processing Services and transform, combine, or create portrayed outputs. Portrayal Services may use styling rules specified during configuration or dynamically at runtime by Application Services. Portrayal Services can be sequenced into a “value-chain” of services to perform specialized processing in support of information production workflows and decision support. Examples of OSF portrayal services include:

- Map Portrayal Services (MPS)
- Coverage Portrayal Services (CPS)
- Mobile Presentation Services

#### 4.3.2.5 Data Services

OSF data services provide access to collections of data in repositories and databases. Resources accessible by Data Services can generally be referenced by a name (identity, address, etc). Given a name, Data Services can then find the resource. Data Services usually maintain indexes to help speed up the process of finding items by name or by other attributes of the item. The sections below describe the current OSF set of Data Services. Examples of OSF data services include:

- Feature Access Services (FAS)
- Coverage Access Services (CAS)
- Sensor Collection Service (SCS)

- Image Archive Services (IAS)

## 4.4 Applying services

This section of the computational viewpoint of the ORM includes examples of service applications implemented by combining or chaining several services.

### 4.4.1 Decision Support Example

The first service application example shows how OGC services can be combined to support decision makers, geospatial specialists, remote sensing experts, domain professionals and incident support teams in emergency or recovery response cases. OGC services provide these players with the infrastructure to publish, find and access multiple information sources hosted by various organizations (such as universities, NGOs, as well as federal, state, and local governments).

Figure 19 shows a web browser accessing a Map Viewer service that is retrieving data from several Web Mapping and Web Feature servers serving different layers of a geographic area. This approach can be used to get and overlay quick maps, access and visualize imagery and other data, and present information for decision. Adding a registry interface will also allow the client to register and discover services at runtime.

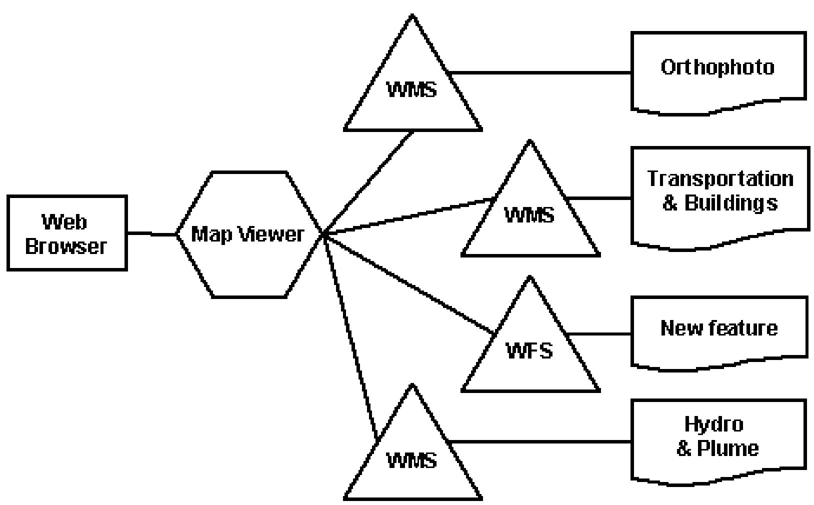



**Figure 19 - Decision Support Example**

### 4.4.2 Service Chaining

The service chaining example portrayed in Figure 20 takes the previous one step further to show how service chaining can help in cases where remotely sensed data fetched from a Web

Coverage Service (WCS) is not directly viewable by a thin client. In this case, a third party Coverage Portrayal Service (CPS) is chained to the WCS to convert data to a map usable by the client. The resultant map is then overlayed with data fetched from other servers for reference and orientation.

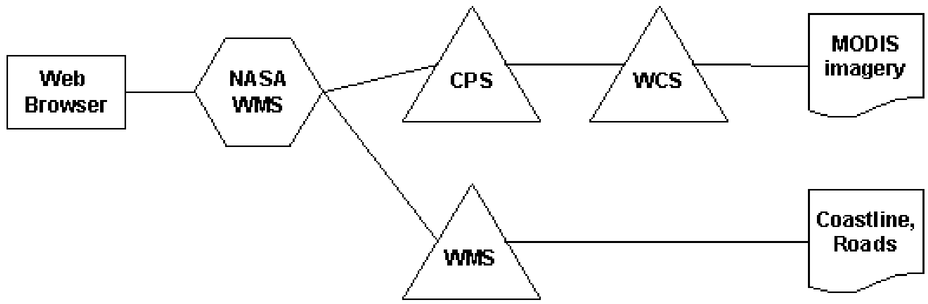



**Figure 20 - Service Chaining Example**

## 5 Engineering viewpoint: multi-tier, multi-network architecture

The Enterprise, Information, and Computation viewpoints describe a system in terms of its purposes, its content, and its functions. The Engineering viewpoint relates these to specific components linked by a communications network. This viewpoint is concerned primarily with the interaction between distinct computational objects[3]: its chief concerns are communication, computing systems, software processes and the clustering of computational functions at physical nodes of a communications network. The engineering viewpoint also provides terms for assessing the "transparency" of a system of networked components – that is, how well each piece works without detailed knowledge of the computational infrastructure. The engineering viewpoint can be described in terms of UML collaboration diagrams and deployment diagrams.[4]

### 5.1 Multi-tier architectures

The engineering viewpoint describes how a system assigns functions and information to various components, or "tiers," along a network. Computational functions, data, and metadata may be found on the server side in one or more intermediate "middleware" components, or on the client side[5]. Figure 21 shows several categories of services arrayed in a logical, 4-tier architecture, and mapped to different physical architectures.

---

[3] http://www.cs.tcd.ie/synapses/public/deliverables/part1.pdf

[4] http://www.informatics.sintef.no/~jol/ODP/rmodp_overview.html

[5] http://www.imn.htwk-leipzig.de/~kudrass/Publikationen/OOPSLA99.pdf

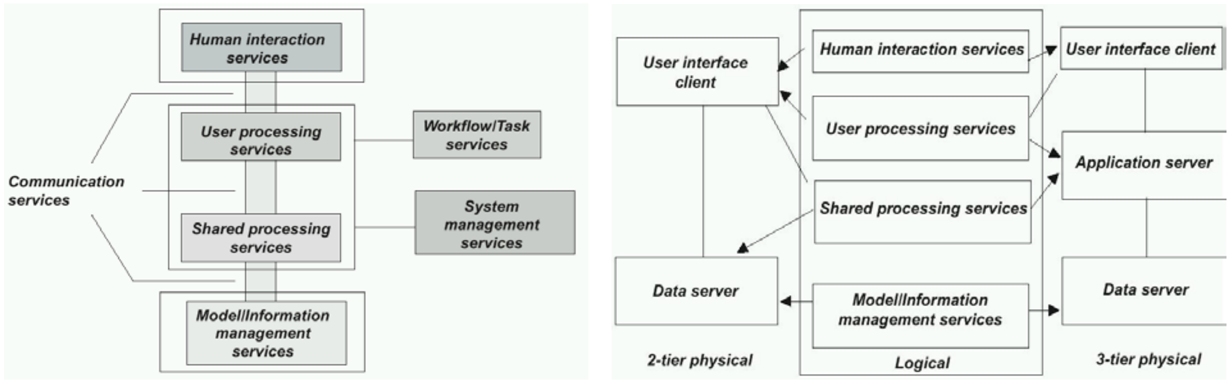



**Figure 21 - Logical multi-tiered architecture mapped to different physical architectures**

### 5.1.1 Thin and thick clients

The engineering viewpoint helps to articulate a key distinction among distributed systems:

- **Thin clients** rely on invoking the services of other components (servers, middleware) for most of the computation they need to function in the system; they also rely on other components to manage most of the data and metadata they need.

- **Thick clients** handle much of the necessary computation and data/metadata management themselves; and rather than invoking the processing services of other components, they obtain their inputs through low-level data-access requests.

A thick client requires less functionality on the part of the server and other components; but a thin client is easier to build or to embed into general-purpose software components. The distinction often has quite tangible implications: thin clients are typically simple software with limited functions and flexibility, and smaller RAM and CPU requirements, often suitable for handheld or mobile devices. Thick clients usually require a significant portion of (at least) a microcomputer's resources, but provide greater flexibility and capacity to decode, transform, render, and interact with retrieved data.

### 5.1.2 A case in point: Web mapping architectures

Web Mapping is one of the key areas in which OGC has explored and discussed thin and thick clients. The Essential Model of Interactive Portrayal[6] compares thin and thick clients (and intermediate "medium" clients), for the special case of portraying geospatial features over the Internet:

[6] http://member.opengis.org/tc/archive/arch98/98-061.pdf

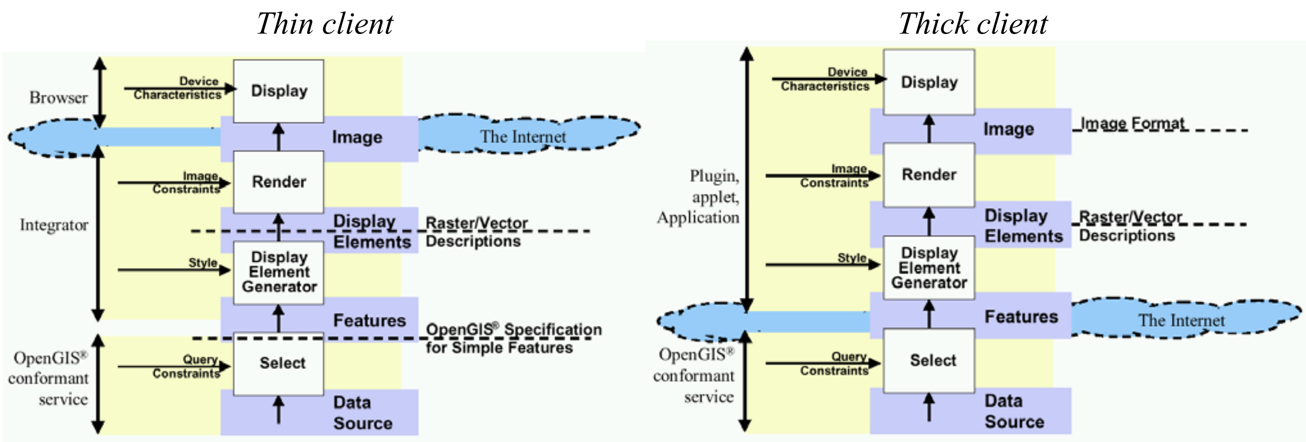



**Figure 22 -Thin vs. thick clients for portraying features over the Internet**

The Essential Model suggests that in a World Wide Web environment, a thin client may be an unadorned Web browser with no need for Java applets or plug-ins. Thus, systems that produce GIF or JPEG files with pictures of maps can be thought of as thin client systems. Such thin clients allow only simple user input in the form of single-clicks on the map or HTML form controls outside the map image. Thick clients (usually applets, plug-ins, or standalone applications) move some or all of the feature-rendering functionality into the client side, and may allow more complex user input.

The Web Map Server Interfaces Implementation Specification, v1.0[7] discusses Web mapping architectures based on thin, thick, and medium clients. [8] It also distinguishes the **picture case** (which transfers pictures to the client, encoded as JPEG, GIF, etc.), the **graphic element case** (which transfers pre-styled, pre-projected picture elements encoded as SVG, PostScript, etc.), and the **data case** (which transfers unretouched geospatial features).

Experience has shown that most Web mapping architectures based on thin clients and/or the picture case rely on a server-side **Viewer Client Generator** to process client requests, maintain or transfer state between requests, and return responses as HTML pages.

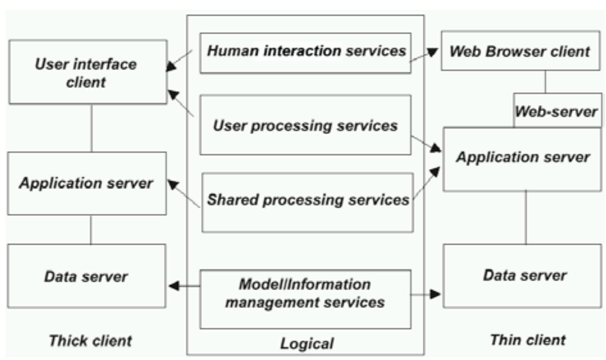



**Figure 23 - Thin vs. thick clients in the World Wide Web environment**

### 5.1.3 Multi-tiers for OSF

The Open Location Services (OLS) and OpenGIS Web Services (OWS) initiatives also defined **server-side client applications,** as “the main server-side components of client applications”. The OWS Services Framework (See section 4.3.2) describes how these components run on the server side of the network, drawing on “user application logic” (business logic) to invoke Registry, Processing, Portrayal, and Data services, and to

[7] http://www.opengis.org/techno/specs/00-028.pdf

[8] The specification warns that the marketing literature of the day (1999) had already encumbered the terms “thick client” and “thin client” by very imprecise definitions. It suggests that a better way to describe and compare Web mapping systems is the kind of information that crosses the boundary between a client computer and a Web server (and additionally, how that information is packaged) – hence the three “cases” that follow.

interact with client-side components through a Web/Portal Server. These components generalize the "viewer client generators" of Web mapping to support thin (small, simple) clients running on mobile devices such as cell phones. Server-side client applications fit into a larger architecture of services, depicted below.[9]

OpenGIS Services are accessible from Application Services operating on user terminals (e.g., desktop, notebook, handset, etc.) or servers that have network connectivity and that utilize OpenGIS service interfaces and encoding specifications (Figure 24). Users may use Application Services to access Registry, Portrayal, Processing and Data Services, depending upon the requirements and designed implementation of the application. Application Services commonly, but not necessarily, provide user-oriented displays of geospatial content and support user interaction at the user terminal. Application Services may be realized as marked-up text (e.g., HTML or XML) transferred across a network from a server, software modules (e.g., Java classes or ActiveX components) transferred across a network and executed on a local system, or as executable code resident on a local system. OpenGIS® Application Services may also support privacy and access controls based on authenticated user identity, however, such controls will typically be provided by an authentication server or some other access control mechanism.

Figure 24 illustrates the distinction between client-side and server-side Application Services.

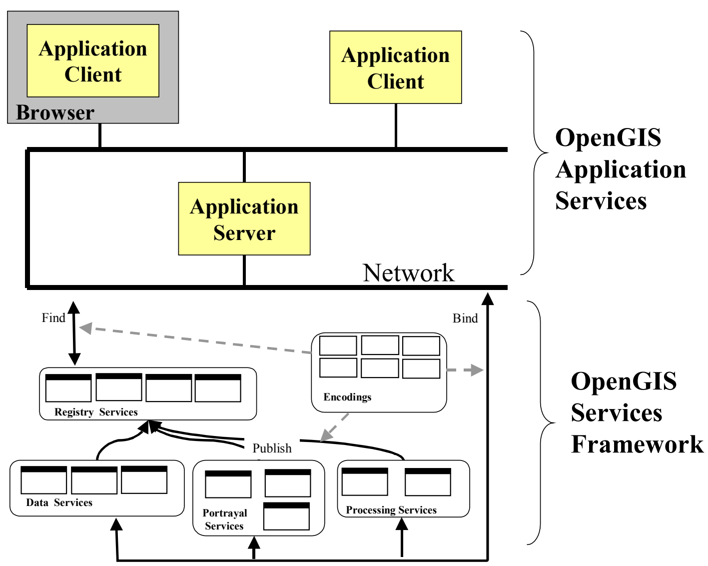



**Figure 24 -Application Services and the OWS Services Framework**

[9] Need to make a distinction between "service types in the computational view" and "service aggregation in the engineering view. "Applications services" is used in the computation view as a class of service types. "Application services" is used in the engineering view as a grouping of services on the users terminals.

Client-side Application Services should:

- Provide the means to find geospatial-based services and data resources through search and discovery mechanisms of Registry Services;
- Provide access to geospatial data (e.g., geographic features and images) and other geospatial-based Application Services and Data Services;
- Integrate with a range of deployment platforms from Web browser-based to desktop to wireless handsets;
- Portray geospatial information in graphic, image, and/or text form;
- Support user interaction via keyboard, cursor or other human-machine interfaces

Application Services should be able to execute not only on the user's desktop (or handset), but also on a server on the network. Examples of server-side Application Services include compute-intensive (and/or I/O-intensive), server-based applications like those required for Image Processing or Route Determination.

Server-side Application Services:

- Implement user application logic (business logic) that utilizes supporting OpenGIS Framework Services such as Registry, Processing, Portrayal, and Data Services.
- Interact with client-side Application Services through an appropriate network protocol depending on the platform being used.
- May be deployed as components of Web Portals and web-accessible business services that add-value to underlying OpenGIS Framework Services.

The above discussion of client-side and server-side Application Services notwithstanding, the OPS Services Framework does not distinguish the myriad options for deploying applications on a network. Instead, any user-facing software component that performs a service that satisfies user-requirements, whether it executes on the client or on the server side of a network connection, is simply an Application Service. The Application Services described below categorize applications by logical function and not physical deployment. Implementations of OpenGIS Application Services are, through standardized interfaces and encodings, freely able to mix and match the capabilities of OpenGIS Services Framework into physical implementations to meet market or application-specific requirements.

#### 5.1.4 Specifying tiers independently

Regardless of how exactly one defines thin vs. thick clients, the multi-tier view afforded by the Engineering viewpoint allows a large specification to be broken into different parts, each addressing a different part of a total system specification (user interface, service invocation, and data/metadata transfer), which can each be mapped into specific technologies (See Figure 25).

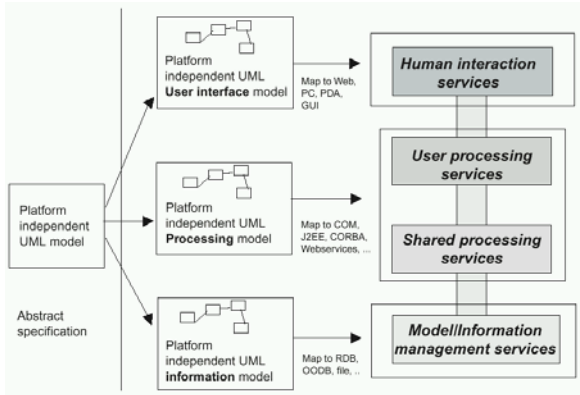



**Figure 25 - Mapping from platform independent UML models**

## 5.2 Bridging multiple networks

The Engineering viewpoint also provides a way to describe systems whose components bridge more than one communications network. In OGC, the Open Location Services initiative has explored this topic, as well as the OWS Sensor Web Enablement (SWE) effort.

### 5.2.1 Open Location Services (OLS)

The Open Location Services (OLS) initiative[10] introduced **gateway services,** which link location application services (accessed via the Internet or the Web) with mobile wireless-IP platforms, in support of small form factor mobile terminals:

[10] http://www.openls.org/docs/001027-openls-rft.pdf

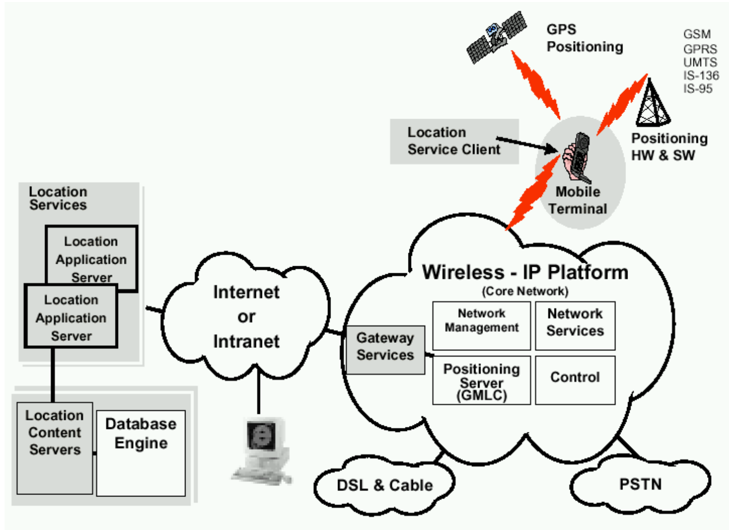



**Figure 26 - OpenLS system concept**

BASELINE The engineering viewpoint for OLS, including definition of components that provide access to services on multiple networks, was developed in the OGC Interoperability Program. The result of that work is an adopted OpenGIS specification titled "Location Service Core Interface Specifications, 03-06r1 and 03-07r1.

### 5.2.2 Sensor Web Enablement (SWE)

The OpenGIS Web Services (OWS) initiative defines a Sensor Web Enablement (SWE) thread to link environmental sensors to the World Wide Web. A Sensor Collection Service (SCS)[11] server gathers readings from in-situ environmental sensors via a private network (cellular, microwave, etc.), and provides summaries or interpretations of those readings to SCS clients over the Web, as depicted below.

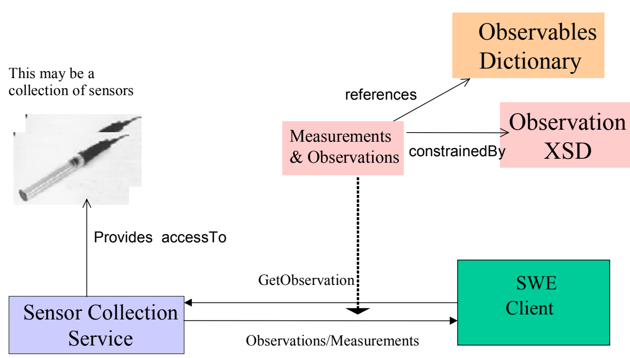



**Figure 27 - Sensor Collection Service concept**

BASELINE The engineering viewpoint for SWE, including definition of components that provide access to sensors on other networks, was developed as part of the OGC Web Services Initiative. This work

[11] http://member.opengis.org/tc/archive/arch02/02-028.doc

has resulted in two documents, both discussion papers: Sensor Collection Service and Sensor Markup Language (SensorML)..

### 5.3 Distribution transparencies

Finally, the Engineering viewpoint provides terms to describe how a system hides complexities associated with system distribution from applications where they are irrelevant to their purpose. To describe how components can work together without a detailed knowledge of network addresses, topology, security, etc., the Engineering viewpoint defines the following eight "transparencies"[12]:

- **access** transparency hides differences in data representation or access mechanisms
- **failure** transparency hides failures and recoveries.
- **location** transparency hides the physical location of an object.
- **migration** transparency hides from an object knowledge about any relocation of itself.
- **relocation** transparency hides relocation of an interface from other interfaces bound to it.
- **replication** transparency conceals behavior associated with the replication of objects.
- **persistence** transparency conceals the deactivation and reactivation of objects.
- **transaction** transparency hides problems of coordination between the activities of a group of objects.

RM-ODP also describes a system's approach to node and cluster management, replication, transactions, and security.[13]

Some of these transparencies are provided by the underlying infrastructure (e.g. the Internet's Domain Name System service provides location transparency), and need not be a significant architectural concern. Others, however, (such as relocation transparency) remain difficult research and engineering problems. Most of those are generic IT challenges, but a few (e.g., replication transparency with complex spatial relationships[14]) have aspects peculiar to geospatial information and geoprocessing.

BASELINE Thus far, few OGC documents have articulated this aspect of the Engineering viewpoint. But as geoprocessing services move from stand-alone, two-tier client-server models to collaborative "computational objects," these transparencies will likely provide a useful structure for organizing the resulting complexity.

---

[12] http://www.dstc.edu.au/Research/Projects/EDOC/ODPEL.html

[13] http://www.firstmark.ca/ContentDocs/architectural%20docs/theopengrouparchframework/rmodp_togaf.htm

[14] Replication transparency masks the fact that multiple copies of a service may be supplied in order to provide reliability and availability. Replication transparency is handled to the extent that different calling semantics can be transparently implemented in the framework, dependent on the policies desired. The supported semantics are shallow copy (copy of one object without its associated objects), partial copy (copy of one object with its direct associated objects) and deep copy (copy of one object and transitively a copy of all its associated objects until no more objects are reachable). [http://www.opengis.org/techno/abstract/02-112.pdf]

# 6 Technology viewpoint: Cross platform implementations

The technology viewpoint is concerned with the underlying infrastructure in a distributed system. It describes the hardware and software components used in a distributed system. The infrastructure, which may be provided by a Distributed Computing Platform (DCP), allows objects to interoperate across computer networks, hardware platforms, operating systems and programming languages.

This section provides an overview of OGC's multiple DCP implementation approach, followed by a list of the current encodings used within OGC, including XML, imagery and well-known binary and text encodings. The Technology viewpoint also summarizes the technologies chosen for the Web Services and the OLS platform implementations.

## 6.1 Multi-platform Implementation Approach

Multiple platform-specific specifications are necessary because of the variety of DCP's and the differences in the way in which they support the functional requirements. It is assumed that one conceptual specification will be the basis for multiple platform-specific implementation specifications.

Development of specifications may proceed from conceptual to implementation or from implementation to conceptual. In either case, a specification shall not be considered complete until it has a conceptual model and at least one implementation.

BASELINE: The multiple DCP approach has been adopted in OpenGIS Abstract Specification, Topic 12 - OpenGIS Service Architecture. Topic 12 is identical with ISO 19119, Geographic information — Services.

### 6.1.1 DCP Independent Application Objects

Recently, the OGC has investigated to definition and implementation of DCP independent application objects. This work was done in the GeoSpatial Objects One initiative. The result of this initiative are documented in the OGC Discussion Paper "GO-1 Application Objects Draft Interoperability Program Report (OGC Document 03-064r1, June 2003). This document is a draft of the OpenGIS" Application Objects Implementation Specification, hereinafter "AOS". The AOS is one of a family of specifications that make up the OGC Geographic Objects activity. The Geographic Objects Initiative was established to develop an open set of common, lightweight, language-independent abstractions for describing, managing, rendering, and manipulating geometric and geographic objects within an application programming environment. This document defines that set of vendor-neutral, Object-Oriented geometric and geographic object abstractions. It provides both an abstract object specification (in UML) and a JAVA specific profile to that that specification. The language-specific binding specifications themselves serve as open Application Programmer Interface (API) specifications for these Application Objects.

## 6.2 Implementation View of the OpenGIS Information Framework

The Implementation View captures how information must be represented within a working enterprise (i.e., how it is encoded for runtime use). The Implementation View must be consistent with the Information View, such that the information modeling concepts constituting the Information View (i.e., class, types, elements, relations, and properties) are preserved in the Implementation View. For example, if the Abstract View maintains that a “road” is a type of geospatial feature in the class “transportation”, with certain properties (e.g., width, surface material, etc), the Implementation View must then represent the transportation feature (road) and its properties in a well-known, standard, runtime form to meet the processing needs that are required for the full life-cycle of this information. Specifically, the basic information type (e.g., vector), descriptive framework (i.e., feature dictionary) and schema (e.g., an application schema represented in GML) of the road must employ “well-known standards” at all points of interoperability (interfaces) that roads will pass through within an interoperable environment.

The Implementation View is comprised of two basic sets of information modeling constructs: *Descriptive Components* and *Runtime Components*.

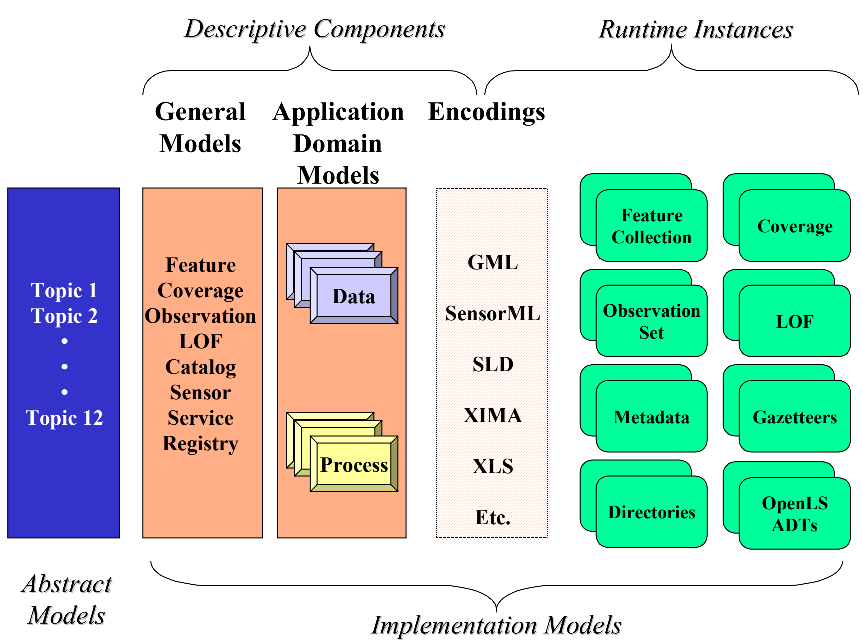



**Figure 28 - OpenGIS Information Framework**

BASELINE This is new information that needs to be elaborated in a separate OGC specification.

#### 6.2.1 Descriptive (Model) Components

*Descriptive Components* represent the means for characterizing and encoding geospatial information and related geoprocessing support information. They are the foundational building blocks that define how geospatial information and related geoprocessing support information is to be represented (i.e., semantics, typing framework, language and schema).

##### 6.2.1.1 General Models

*General Models* define the basic models for how geospatial information is to be characterized and encoded. To date, OGC has defined several types of General Models:

- (Simple) Feature Model – The general, descriptive model for how earth features may be represented as vector objects (i.e., points, lines and polygons).
- Coverage Model – The basic model for how earth information may be represented as raster or grid coverages (e.g., an image or digital terrain model).
- Observation Model – The general model for representing observations of earth phenomena; general observation model for describing well-known observations.
- Location Organizer Folder (LOF) – The general, multi-source information container model for handling sets of inter-related spatiotemporal information. A LOF can contain such things as images, maps, features, cables, and any other information elements (e.g., audio, video, etc). LOFs also contain relationships (links) between any LOF information elements, or between LOF elements and external resources (e.g., a link to a Web site, a spatial query for a specific database, etc.). LOFs are equivalent to a digital version of a “shoebox”.
- Registry Model – The general model for online registries.
- Sensor Model – The general model for sensor phenomena; the general sensor model for describing well-known sensors.
- Service Model – The general model for online services.
- Data Catalog Model – The general model for representing online data catalogs that pertain to enterprise data stores.

The following additional General Models would be beneficial for enhanced interoperability:

- Dictionary Model – The general model for representing online dictionaries that pertain to well-known types of classification schemes and dictionaries.
- Directory Model – The general model for representing online, well-known types of directories (e.g. Yellow Pages).
- Gazetteer Model – The general model for representing online, well-known types of gazetteers (e.g. NIMA Gazetteer).

#### 6.2.1.2 Application Domain Models [15]

**Application Domain Models** are application-oriented models that characterize information and service resources within a domain. They are often based upon a General Model and must always be consistent with the Abstract Model. There are two subclasses of these models: Data Domain Models and Process Domain Models.

**Data Domain Models** characterize the application schemas for well-known, domain-specific data resources.

- Feature Domain Models – the definition (typing framework and properties) of a domain-specific application schema for a well-known class of geospatial features, in vector form (i.e., points, lines and polygons). For example: Transportation, Hydrographic, Electric Utility, etc.
- Coverage Domain Model - the definition of a domain-specific application schema for a well-known geospatial coverage. For example: DTED.
- Sensor Domain Model - the definition of a specific sensor type in accordance with the general sensor model.
- Observation Domain Model – the definition of a specific observation type in accordance with the general observation model.

**Process Domain Models** characterize well-known, domain-specific business processes. These models capture business rules, policies, tasks, and procedures in the form of processing chains. The IT industry is still fleshing out the basic standards for this area. Much progress has been made over the past year, but it would be prudent to wait to see what the industry leaders produce in the next year or two. Specifically, Microsoft, IBM and others are now collaborating on a standard methodology for online workflow and service chaining. When this standard stabilizes and emerges, it would then make sense to start testing this technology and adapt it in a wide range of workflows. When that happens, many Process Domain Models will result.

### 6.2.2 Modeling Languages

**Modeling Languages** are well-known 'languages' to encode the semantics, syntax and schema of geospatial and geoprocessing-related information resources. They apply to all Application Domain Models and Runtime (Model) Components. (Strictly speaking, the General Models should be language-independent. However, for convenience, one may choose to define a General Model with a particular language, such as GML. However, this is not recommended if more than one encoding technology is to be employed.)

The suitability of an information modeling language for meeting the runtime needs of an environment depends upon:

[15] Need to determine if Application Domain Models and Application Schema are the same concept.

1) The ability to adequately and accurately represent all required classes, types, elements, relationships and properties for the required information.

2) Adequate tools to build, validate, publish, and maintain the models in runtime form.

3) The ability to support efficient, interoperable runtime implementations.

4) The OpenGIS Service Framework must support it.

The languages and associated technologies that a functioning enterprise depends upon are part of the **Implementation Profiles** for an environment. An Implementation Profile contains an interoperable set of implementation technologies. In addition to modeling languages, Implementation Profiles also contain inter-process communication protocols and other dependent infrastructure technologies that the framework employs. The framework may support one or more Implementation Profiles.

To date, OGC has defined (or inherited for use) the following methods of encoding, all which are based upon XML:

- Geographic Markup Language – the language for describing and encoding geospatial information.
- Sensor Model Language – the language for describing and encoding sensors (in situ, satellite and airborne).
- Styled Layer Descriptors (SLD) – a map-styling language for producing georeferenced maps with user-defined styling
- XML for Imagery and Map Annotations (XIMA) – the means to encode annotations on imagery, maps, and other geospatial data.
- XML for Location Services (XLS) – the encoding method for OpenLS-based Abstract Data Types.
- Web Services Description Language – the language for describing and encoding services; primarily used to populate Service Registries (Publishing Services), service discovery (Discovery Services) and service access.

### 6.3 OGC Encodings; including GML

Encodings describe specialized vocabularies for the transfer of specific kinds of data packages as messages between application clients and services, and between services. This section first lists the XML-based encodings, followed by the Imagery and well-known binary and text encodings used within the OSF.

#### 6.3.1 Style Layer Descriptor

The Styled Layer Descriptor encoding specifies the format of a map-styling language for producing geo-referenced maps with user-defined styling. SLD addresses the need for geospatial consumers (either humans or machines) to control the visual portrayal of the data with which they work. The ability for a human or machine client to define the styling rules requires a styling language that the client and server can both understand. SLD is defined using XML Schema and can be used to portray the output of Web Map Servers, Web Feature Servers and Web Coverage Servers.

Defining this language, called the ***StyledLayerDescriptor (SLD)***, is the main focus of this specification, and it can be used to portray the output of Web Map Servers, Web Feature Servers and Web Coverage Servers. In many cases, however, the client needs some information about the data residing on the remote server before he, she or it can make a sensible request.

BASELINE SLD Version 1 is an adopted OGC specification.

#### 6.3.2 Geography Markup Language encodings

The Geography Markup Language is an XML grammar written in XML Schema for the modeling, transport, and storage of geographic information., including both the geometry and properties of geographic features. GML utilizes the OpenGIS Abstract Specification geometry model. GML includes the ability to handle complex properties.

Below are several OGC encoding specifications that are application profiles of GML.

##### 6.3.2.1 GML3

This version (3.0) of GML addresses the following needs that were not addressed or adequately met by the previous version:

- Represent geospatial phenomena in addition to simple 2D linear features, including features with complex, non-linear, 3D geometry, features with 2D topology, features with temporal properties, dynamic features, coverages, and observations;
- Provide more explicit support for properties of features and other objects whose value is complex
- Represent spatial and temporal reference systems, units of measure and standards information;
- Use reference system, units and standards information in the representation of geospatial phenomena, observations, and values;
- Represent default styles for feature and coverage visualization;
- Conform with other standards, including
  - ISO DIS 19107 Geographic Information – Spatial Schema
  - ISO DIS 19108 Geographic Information – Temporal Schema

- ISO DIS 19118 Geographic Information – Encoding
- ISO DIS 19123 Geographic Information – Coverages

BASELINE The old GML baseline was GML2.1.1 and is still in use. GML3 was approved in February 2003. Considerable work was done in version 3.0 to harmonize GML with other OGC and ISO specifications, such as CRS, ISO 19107, and ISO 19118.

#### 6.3.2.2 XML for Image and Map Annotation (XIMA)

The XML for Image and Map Annotation (XIMA) defines an XML vocabulary to encode annotations on imagery, maps, and other geospatial data. This vocabulary draws on the Geography Markup Language (GML) to express the positions of these annotations in geographic (real world) or image-pixel coordinates, and to associate each annotation with the geospatial resource(s) it describes. The XIMA encoding is useful for any activity that requires linking or tagging geospatial data in order to present and discuss it with others, to make joint decisions, or to communicate spatially.

BASELINE XIMA is a discussion paper. It uses GML 2.xx and has not been updated to reflect the changes in GML 3.0.

#### 6.3.2.3 Location Organizer Folder (LOF)

The Location Organizer Folder (LOF) is a GML application schema that provides a structure for organizing the information related to a particular event or events of interest. It may be used in various analysis applications, like disaster analysis, Intelligence analysis, etc. It is spatially enabled, and capable of managing disparate types of information. There may be a variety of services external to the LOF that operate on it to add, modify and manipulate resources in or referenced by it. This includes search and discovery, parsing different resources and extracting useful information, assigning spatial attributes, relating (linking) resources of interest, and so on.

BASELINE: LOF is a discussion paper. It uses GML 2.xx and has not been updated to reflect the changes in GML 3.0.

### 6.3.3 Other XML Encodings

#### 6.3.3.1 Recommended XML Encoding for Coordination Reference Systems

This Recommendation Paper provides an XML encoding for commonly used coordinate reference systems.

#### 6.3.3.2 Web Map Context Document Specification

WMS 1.1.1 specifies how individual map servers describe and provide their map content. The present Context specification states how a specific grouping of one or more maps from one or more map servers can be described in a portable, platform-independent format for storage in a repository or for transmission between clients. This description is known as a "Web Map Context Document," or simply a "Context." Presently, context documents are primarily designed for WMS bindings. However, extensibility is envisioned for binding to other services.

A Context document includes information about the server(s) providing layer(s) in the overall map, the bounding box and map projection shared by all the maps, sufficient operational metadata for Client software to reproduce the map, and ancillary metadata used to annotate or describe the maps and their provenance for the benefit of human viewers.

A Context document is structured using eXtensible Markup Language (XML). Annex A of this specification contains the XML Schema against which Context XML can be validated.

BASELINE: The specification and use of context documents is described in OpenGIS Specification “Web Map Context Document” (OGC Document, 03-036r2, May 2003)

#### 6.3.3.3 Service Metadata

Service Metadata is a vocabulary comprised of several parts for describing different aspects of a service. The first unit describes the service interface in sufficient detail so that an automated process can read the description and invoke an operation that the service advertises. A second unit describes the data content of the service (or the data it operates on) in a way that enables service requestors to dynamically compose requests for service. This content unit description is optional, depending on whether the service contains or operates on data content. Additional description units provide information specific to particular types of services as well as specific instances of services.

There are certain basic parts of service metadata that need to be managed independently to support specialized registry functionality and/or useful descriptive granularity. The parts of the Service Metadata vocabulary are:

1. Service Type (or Operation Signature)
2. Service Instance (advertising a collection of ServiceOffers)
3. Service Metadata (ISO 19119 subset, as properties of ServiceOffers)
4. Content Type (or Data Type or Data Product Specification at a more specific level)
5. Content Instance (or Data Instance, including ISO 19115 subset)
6. Content Organization

7. Common

It should be emphasized that not every service need provide every one of these capabilities components. The only essential components (in the case of no inherent service content) are 2) Service Instance and 3) Service Metadata, which in turn would make reference to externally maintained Service Type components.

BASELINE OGC Discussion Paper titled “OGC Web Services Registry Service” (OGC Document 03-024, January 2003)

#### 6.3.3.4 Image Metadata

Image Metadata is an XML encoding used to adequately describe all types of images handled by OpenGIS Framework services. Image Metadata is used for publishing and discovery of types of original and derived images, image identifications, dates, spatial extents and other information that could be used to find and retrieve images from an archive. This same metadata may also include image geometry model parameters (although they might not be queried, just returned) as expressed with SensorML.

BASELINE TBD

#### 6.3.3.5 SensorML

The Sensor Model Language defines an XML schema for describing the geometric, dynamic, and observational characteristics of sensor types and instances. Sensors are devices for the measurement of physical quantities. There are a great variety of sensor types, from simple visual thermometers to complex electron microscopes and earth observing satellites.

The purpose of the sensor description is threefold: (1) to provide general information about the sensor in support of data discovery, (2) to support the processing and analysis of the sensor measurements, and (3) to support the geolocation of the measured data.

The Sensor Model Language is a human-readable, XML-based language that can be easily parsed by a wide variety of existing tools. The current standard calls for keywords in the English language, although consideration for internationalization of keywords should be considered if deemed beneficial.

BASELINE SensorML is a discussion paper

### 6.3.4 Imagery and Gridded Data Encodings

Many encodings for imagery and gridded data exist. The following encodings are referenced by OGC Implementation specifications

#### 6.3.4.1 JPEG

JPEG (Joint Photographic Experts Group) for continuous tone pictures: JPEG makes use of continuous-tone digital images much more economical by drastically reducing the volume required for storage and the bandwidth required for transmission. It enables interchange of images by providing a common coded representation of compressed image data.

#### 6.3.4.2 DIGEST (Digital Geographic Exchange Standard)

The DIGEST standard supports images and gridded data in alignment with the ISO/IEC 1/SC 24 BIIF standard. DIGEST Annex D, known as the Image Interchange Format, is an encapsulation of the NATO Secondary Imagery Format (NSIF), which allows for the standard exchange of image, graphic and text data.

#### 6.3.4.3 EOSDIS/HDF

HDF-EOS is an extension of NCSA (National Center for Supercomputing Applications) Hierarchical? Data Format. HDF-EOS data format adds mechanisms for storing geo-referencing and temporal information, data organization, and metadata storage. HDF-EOS contains Grid, Point and Swath structures. HDF supports multiple data types, its portability, its ease of use and implementation, the availability of free software and documents and the availability of software tools for manipulating and visualizing data in HDF.

#### 6.3.4.4 GeoTIFF

The GeoTIFF data interchange standard for raster geographic images is an extension of the TIFF format to support a geodetically sound raster data georeferencing capability. The use of private tags in TIFF has lead to the extension GeoTIFF, where private tags are introduced to support some geo-metadata and geo-referencing. The GeoTIFF format is quite common since people that have access to software that can view TIFF images can at least view the pixels from a GeoTIFF file ignoring metadata, registration, etc. The geographic content supported in GeoTIFF tag structure includes its cartographic projection, datum, ground pixel dimension, and other geographic variables. The aim of GeoTIFF is to allow a means for tying a raster image to a known model space or map projection, and for describing those projections.

GeoTIFF is not intended to become a replacement for existing geographic data interchange or metadata standards, such as FGDC, USGS or ISO standards. Rather, it aims to augment an existing popular raster-data format to support georeferencing and encoding information that is compatible, in principle, with these standards, the National Spatial Data Infrastructure (NSDI) and other emerging international standards.

#### 6.3.5 Well-Known Binary and Text Encodings

This section summarizes the Well-Known Binary Representation for Geometry (WKBGeometry) and the Well-Known Text Representation of Spatial Reference Systems as described in ISO 19125-1, Geographic information- Simple feature access.

##### 6.3.5.1 6.2.3.1 Well-Known Binary Representation for Geometry (WKBGeometry)

WKBGeometry provides a portable representation of a Geometry value as a contiguous stream of bytes. The Well-known Binary Representation for Geometry is obtained by serializing a geometric object as a sequence of numeric types drawn from the set {Unsigned Integer, Double} and then serializing each numeric type as a sequence of bytes using one of two well defined, standard, binary representations for numeric types (NDR, XDR). The specific binary encoding (NDR or XDR) used for a geometry representation is described by a one byte tag that precedes the serialized bytes. The only difference between the two encodings of geometry is one of byte order. The XDR encoding is Big Endian (Unsigned Integer represented using most significant byte first, Double represented using first byte as sign bit), and the NDR encoding is Little Endian (Unsigned Integer represented using least significant byte first, Double represented using last byte as sign bit).

##### 6.3.5.2 6.2.3.2 Well-Known Text Representation of Spatial Reference Systems

The Well-Known Text Representation of Spatial Reference Systems provides a standard textual representation for spatial reference system information. The definitions of the well-known text representations are modeled after the POSC/EPSG coordinate data model.

### 6.4 Web Services Platform

This section presents the technologies and standards chosen for the OGC Web Services interoperability platform. This section first depicts a layered architecture of web technologies and standards on which services can be implemented and deployed, then lists the services that were implemented using these technologies and standards in the first phase of OGC's OWS testbed.

#### 6.4.1 Web Services Stack

Figure 29 depicts a layered architecture of technology and standards on which web services can be implemented and deployed. The lowest levels of the stack enable connectivity of software components by enabling them to bind, send and receive messages. Higher levels in the stack enable interoperability and, via publish-find-bind mechanisms, allow software components to transparently work together in more integrated and dynamic ways.

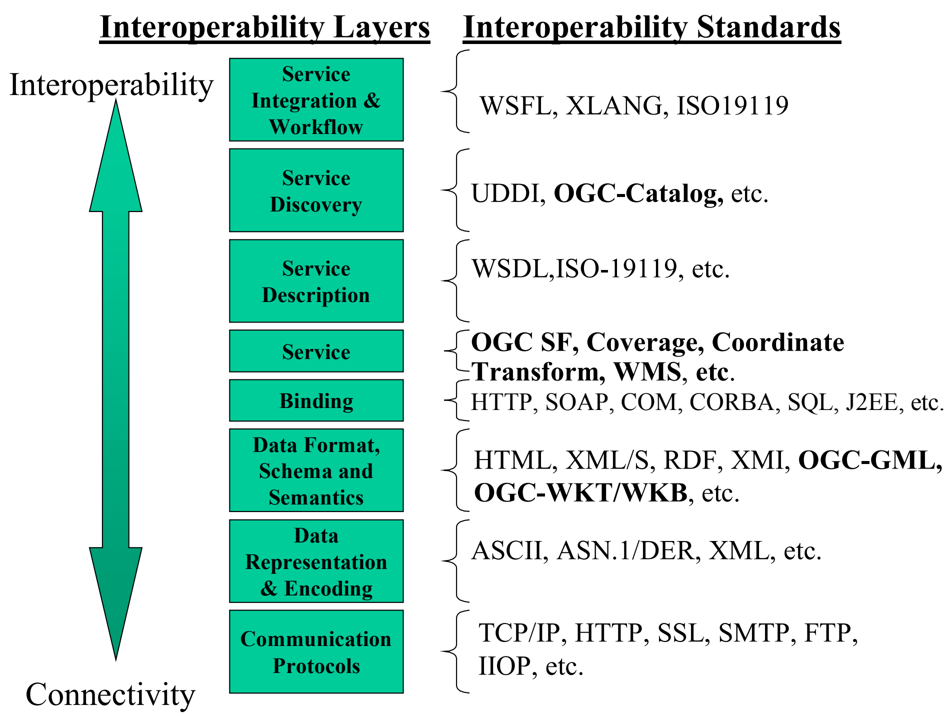



**Figure 29 - Services Interoperability Stack**

#### 6.4.1.1 HTTP as Distributed Computing Platform

OGC has defined a suite of Web Service interfaces that have explicit bindings for HTTP. Specifically, there are two HTTP bindings for invoking operations of a service (i.e., Sending a message): GET and POST. Thus the Online Resource for each operation supported by a service instance is an HTTP Uniform Resource Locator (URL). Only the parameters comprosing the service request itself are mandated by OGC Web Service Specifications for HTTP.

TBD- point to list of implementation specs in annex.

#### 6.4.1.2 WSDL as Service Description Language

Web Services Description Language (WSDL) provides a model and an XML format for describing Web services. WSDL enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as "how" and "where" that functionality is offered.

This specification defines a language for describing the abstract functionality of a service as well as a framework for describing the concrete details of a service description. The *WSDL Version 1.2 Part 2: Message Patterns* specification [*WSDL 1.2 Message Patterns*] defines define the sequence, direction, and cardinality of abstract messages sent or received by an operation. The *WSDL Version 1.2 Part 3: Bindings* specification [*WSDL 1.2 Bindings*] defines a language for describing such concrete details for SOAP 1.2 [*SOAP 1.2 Part 1: Messaging Framework*], HTTP [*IETF RFC 2616*] and MIME [*IETF RFC 2045*].

A service provider uses WSDL documents to publish Web services, to find published services and to bind to services dynamically. WSDL was used in the first phase of OWS in the OperationSignatures portion of a service XML Capabilities document.

WSDL was designed so that, given an understanding of the semantics of a given operation, a programmer could discover and build an interface to a particular service instance. WSDL would allow automatic generation of "Stub" code for the interface, and basically a programmer would be required to extend this according to an understanding of the semantics of the content, and context for operations. Further knowledge of the content is required to "bind" to the service with a meaningful query.

BASELINE: As part of the OGC Web Services 2 initiative, experiments were performed to evaluate the usefulness of WSDL for describing OGC Web Service interfaces. Further WSDL experiments will be performed as part of OWS 3. Also, a WSDL profile for the WFS specification will be part of the next release of WFS.

#### 6.4.1.3 SOAP as an OGC Web Services Enabler

SOAP is an XML-based remote procedure call protocol that allows messages to be exchanged between different services. SOAP is used for the bind operation described above. As part of OGC Web Service, a SOAP implementation experiment for WMS was conducted. The SOAP experiment had three primary goals:

- Use the Unified Modeling Language (UML) to define a platform independent model of Web Map Server interfaces
- Use the UML WMS model to define the appropriate XML Schema and Web Service Definition Language (WSDL) files that allow OWS services to be invoked using standard protocols such as HTTP GET, HTTP POST and SOAP
- Use standard COTS Web Services toolkits to invoke OGC Web Services across the Internet.

This work is in progress. The report documents a large number of issues that impact the use of SOAP for OGC Web Services.

BASELINE: OGC Discussion Paper "OWS 1.2 SOAP Experiment Report" (OGC Document 03-14, January 2003)

#### 6.4.2 OGC Registry Model and Service Metadata

The OGC registry information model (ogcRIM) is based on the ebXML registry information model (ebRIM, v 2.1), and thus makes the following assumptions: 1) all access to registry content is performed through the interfaces defined for the Registry Service; 2) the information model provides a basis for interoperability by constraining the behavior of the registry (i.e. it helps the implementor to undestand what to build by defining the kind of metadata being manipulated and how they are interrelated); 3) the registry makes use of a repository for storing and retrieving persistent information and shared resources; but no assumptions are made concerning the exact nature of the repository or its location.

Adopting a broader perspective, a registry service can be viewed as a type management system that fulfils a basic repository function in open, distributed systems. It can

- provide persistent type information for use in service discovery;
- facilitate dynamic (i.e. late or run-time) binding to service instances;
- provide support for run-time type checking for safety;
- be linked to—or federated with—other registries.

Every registry instance implements an information model that provides a high-level schema that defines what types of objects are stored in the registry and how they are organized. Prominent models within the web services realm include the ebXML registry information model (ebRIM) and the UDDI data model [11]. The APIs associated with both models support multiple query patterns: browse and drill-down (by category), or filtered queries against specified registry objects. However, the ebRIM is more general and extensible—it draws on the ISO 11179 set of standards to provide comprehensive facilities for managing metadata. The UDDI model focuses on business entities and associated service descriptions (even though `tModel` objects can accommodate many kinds of technical specifications).

The `Service` offer is the root object for OGC service-related metadata; other objects may be associated with it to provide additional information. For example, every OGC service description is classified according to the OWS service taxonomy in order to distinguish different kinds of services; and every service offer is associated with a capability profile that conveys non-operational service properties (e.g. supported query languages, quality of service, vendor-specific extensions). The operational characteristics of a service are described through *binding* specifications (e.g. WSDL or DAML-S documents). Many data access services are associated with descriptions of available datasets. Figure xx informally illustrates how the various service metadata instances are related in a registry.

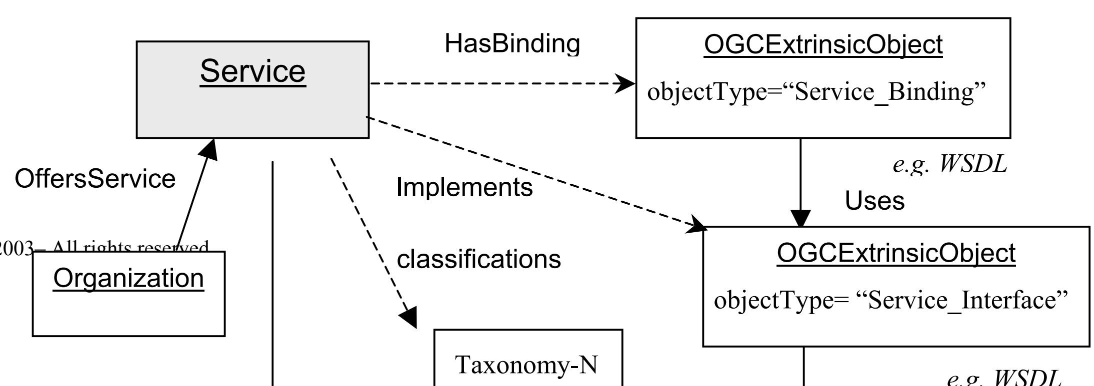


**Figure xx — Service-related metadata in the registry**

A `Service` object is at the hub of the associations portrayed in Figure 16—it conveys basic metadata about a service instance, including version, status, and expiration date. A registry may be capable of 'harvesting' an updated capability profile for a service; it will attempt to do so when a service offer expires. If the attempt fails, the 'status' attribute of the service offer is changed to "Withdrawn".

A service offer may be classified using any number of classification schemes. A `Classification` instance would then associate a `Service` object with a node belonging to some `ClassificationScheme`. For example, a service would be classified as an OGC Sensor Collection Service by associating it with the corresponding classification node from the OGC service taxonomy.

Employing a generic registry information model obviates the need for special-purpose registries. A registry instance could conceivably catalog many kinds of resources: service offers, application schemas, taxonomies, interface types, UML models, even software components. But a registration authority (RA) is free to impose a policy that restricts what type of resources can be registered (i.e. only schemas, or only service offers). However, if such a publication restriction is imposed, the registry must be able to communicate with other registries that can provide the necessary type descriptions.

Clause 7.4 of ISO 19119 (Geographic Information – Services) deals with service-related metadata. The metadata elements for a service are intended to provide sufficient information to allow a client to invoke the service based on the metadata content. Furthermore, a service instance (i.e. a software entity comprised of one or more server processes) may be tightly coupled with a dataset instance; in this case the service metadata shall describe both the service and the geographic dataset. Mirroring such an intimate association in a registry facilitates a wide range of discovery scenarios, including searching for appropriate services and searching for desired datasets

#### 6.4.3 Web Mapping Service (WMS)

The WMS specification standardizes the way in which clients request maps. Clients request maps from a WMS instance in terms of named layers and provide parameters such as the size of the returned map as well as the spatial reference system to be used in drawing the map.

This specification defines three WMS operations: `GetCapabilities` returns service-level metadata, which is a description of the service's information content and acceptable request parameters; `GetMap` returns a map image whose geospatial and dimensional parameters are well-defined; `GetFeatureInfo` (optional) returns information about particular features shown on a map.

Need to standardize the font for operations; in this section, we have both Courier New 12pt, Times New Roman 12 pt and Times New Roman italicized 12 pt.

BASELINE WMS is an adopted OGC Implementation Specification

#### 6.4.4 Web Feature Service (WFS)

The Web Feature Service (WFS) supports INSERT, UPDATE, DELETE, QUERY and DISCOVERY of geographic features. WFS delivers GML representations of simple geospatial features in response to queries from HTTP clients. Clients access geographic feature data through WFS by submitting a request for just those features that are needed for an application.

A WFS can either be a basic WFS (a READ-ONLY WFS), which implements the `GetCapabilities, DescribeFeatureType` and `GetFeature` interfaces, or a transaction WFS, which, in addition to supporting all the interfaces of a basic WFS, implements the Transaction interface (and optionally the LockFeature interface).

BASELINE WFS is an adopted OGC Implementation Specification

#### 6.4.5 Web Coverage Service (WCS)

The Web Coverage Service supports the networked interchange of geospatial data as "coverages" containing values or properties of geographic locations. Unlike the Web Map Service, which returns static maps (server-rendered as pictures), the Web Coverage Service provides access to intact (unrendered) geospatial information, as needed for client-side rendering, multi-valued coverages, and input into scientific models and other clients beyond simple viewers.

The Web Coverage Service consists of three operations: *GetCapabilities* returns a description of the service with elements to describe multidimensional data collections from which a coverage may be requested; *GetCoverage* returns values or properties of geographic locations, bundled in a well-known coverage format; *DescribeCoverageType* returns a description of the structure of the coverages which the WCS returns.

BASELINE WCS is an adopted OpenGIS Specification

#### 6.4.6 Coverage Portrayal Service (CPS)

The Coverage Portrayal Service defines a standard interface for producing visual pictures from coverage data. CPS extends the WMS interface and uses the Styled Layer Descriptor (SLD) language to support rendering of WCS coverages. In addition to request parameters, which serve to qualify the request action, additional information has to be provided to the CPS for it to be able to do its work. SLD is used to express additional information about:

1. where to get the coverage that the client is interested in,
2. what part or parts of the coverage data to work with, and
3. what the client wants the CPS to do with that data.

Consequently, the CPS integrates into the OGC architecture by implementing two standard OGC interfaces, the WMS interface and the WCS interface.

#### 6.4.7 Sensor Collection Service (SCS)

The basic function of the *Sensor Collection Service* (SCS) is to provide a web-enabled interface to a *sensor*, *collection of sensors* or *sensor proxy*. The Sensor Collection Service provides a standard interface for clients to collect and access sensor observations and manipulate them in different ways. SCS instances are collection points on the web for disparate types and instances of sensors. SCS instances deliver sensor observation values (e.g., temperature, ppm, chemical type) in response to queries form HTTP clients.

Several interfaces are provided by a Sensor Collection Service (SCS). Observed Values may be encapsulated within Coverages or Measurement Features, which are available through the `GetCoverage` and `GetFeature` interfaces (respectively). Observed Values may also be available directly through the `GetObservation` interface, in which case the meaning of the measurements must be known from the context. Thus, data provided by a SCS might be in the form of instance documents describing Measurements (i.e., timing, location) or might include only Observed Values (i.e., no explicit timing or location in the data document). As with other OGC Web services, the SCS also implements the `GetCapabilities` interface.

BASELINE SCS is an OGC discussion paper

#### 6.4.8 Geocoder Service

Geocoding is the process of linking words, terms and codes found in a text string to their applicable geospatial features, with known positions (i.e., usually a point with x, y coordinates but more generally any geometry). The most commonly known type of geocoding is converting a street address to a geographic location. The Geocoder Service Interface allows for a request providing an address or set of addresses and returns them along with the corresponding geometry (usually a point relative to a requested spatial reference system.) The request is "sent" to a Geocoder Service, which processes the request and returns the resulting geographic feature representing position.

BASELINE Geocoder is an OGC discussion paper

#### 6.4.9 Gazetteer Service

The Gazetteer Service is a network-accessible service that retrieves the known geometries for one or more features, given their associated well-known feature identifiers (text strings), which are specified at run-time through a query (filter) request. The identifiers are any words or terms that describe the features. These features are well known to the Gazetteer Service, such as a set of place names and/or landmarks. Each instance of a Gazetteer Service has an associated vocabulary of identifiers. Thus, a Gazetteer Service may apply to a given region, such as a country, or some other specialized grouping of features. The returned geometries are expressed in an OGC Spatial Reference System according to the ISO feature model and could be encoded in GML.

A Gazetteer Service may be thought of as a special case of a Geocoder Service that is an authority for place names and their associated geometries. This is expected to be consistent with the definition of a gazetteer in ISO CD 19112 - "Geographic Information - Spatial referencing by geographic identifiers". This specification adopts a data model based on CD 19112, with derivation and mappings explicitly noted. This has been extended with optional elements whose contents have application defined semantics to support the Alexandria Digital Library Gazetteer Content Model, without losing explicit semantics derived from the ISO model.

A Gazetteer Service exposes both a potentially structured vocabulary and a feature collection, and thus may support, with the same data structure, all the capabilities of a hierarchical vocabulary and a Feature Access Service. However, this version of the specification is limited to the latter case. Thus, it only applies to Gazetteer Services that deliver features that instantiates a well-defined OGC GML schema, in response to any set of geographical queries, as defined in the OGC Filter Encoding Spec. It is the responsibility of the service to advertise which geographical queries it supports.

BASELINE Gazetteer is an OGC discussion paper

#### 6.4.10 Geoparser Service

Geoparsing refers to the capability to process a textual document and identify key words and phrases that have a spatial context. A Geoparsing Interface implementing this specification works in the context of two bodies of information: a reserved vocabulary (usually of place names) and a text source (e.g., a newspaper, or voice track.) The Geoparser Service returns all occurrences of the use (in the text source) of any word in the reserved vocabulary. Each occasion establishes a geolink between the source and the location associated with the reserved word. For example, your home might be geolinked with your water bill through their common address.

BASELINE Geoparser is an OGC discussion paper

# 7 Scenarios Illustrating ORM[16]

The following scenarios represent typical applications of ORM for specific environments. The first, *Information Producer Scenario*, explores the application of OpenGIS Information and Services Framework to a production enterprise. The second, *Information Broker Scenario*, shows how ORM might be applied in implementing an information community. The third, *Information Consumer Scenario*, applies ORM to an open location services application for wireless mobile subscribers.

## 7.1 Information Producer Scenario

The first scenario is centered on the production of a standard product called Foundation Feature Information (FFI), from an assortment of sources, at a Civilian Mapping Agency.

This formal scenario name is the "Base Map Production Unit (BMPU) of the Average Civilian Mapping Agency (ACMA)". The scenario is called the *Production Scenario* for short. The motto of the ACMA is "FFI Production R Us."

From the Enterprise Viewpoint, this scenario encompasses the central reason for the existence of the Civilian Mapping Agency: the production of timely, accurate, and useful Feature Foundation Information to include elevation information, imagery information, and feature information. A variety of input, resources, and control parameters make the production goals possible.

The scenario focus is on three major steps: (1) the establishment of image control so that stereoscopic and photogrammetric technology can support the discovery of features and the extraction of feature geometry, attributes, and relationships in support of FFI production, (2) the extraction and/or update of new features, and (3) the fusion of new features into legacy FFI to create new or updated information conforming to FFI specifications.

### 7.1.1 Enterprise Viewpoint

An abstract enterprise view of the Production Scenario is shown at Figure 30. This scenario involves the production of Foundation Feature Information (FFI) in a Civilian Mapping Agency (CMA). The scenario places its focus on the high-level activities that convert the input (consisting of imagery, legacy features, and other sources) into managed, stored, Foundation Feature Information.

---

[16] The scenarios need to be updated to utilize the architecture defined in the ORM viewpoints.

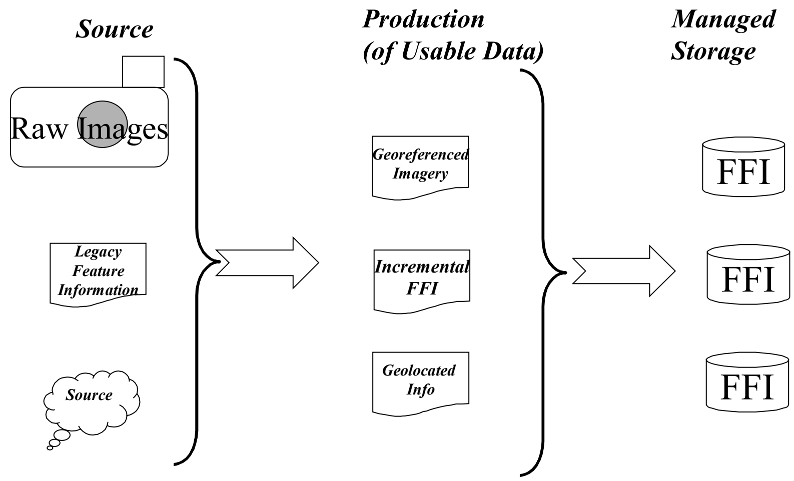



**Figure 30 - Information Production Scenario**

Figure 31 summarizes the Enterprise Viewpoint for Information Production at a high level of abstraction.

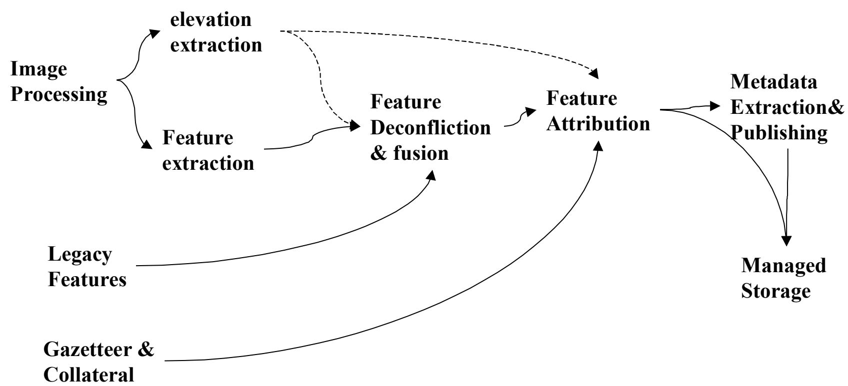



**Figure 31 - Enterprise Viewpoint: Production**

The intermediate processes that arbitrate? this high level view are finer-grained input-to-output process and services. These finer-grained processes refine the sources through the creation and finishing of intermediate and usable information sets that may include imagery, feature data, and other geolocated information ready for inclusion and management within Foundation Feature Information. A finer-grained enterprise view is provided at Figure 32. This view reflects the concerns of the enterprise owner or manager, namely, componentization, the interoperability of components, and reduction of risks that accompany common operations and common data types.

Figure 32 summarizes the Production Scenario from the Enterprise Viewpoint.

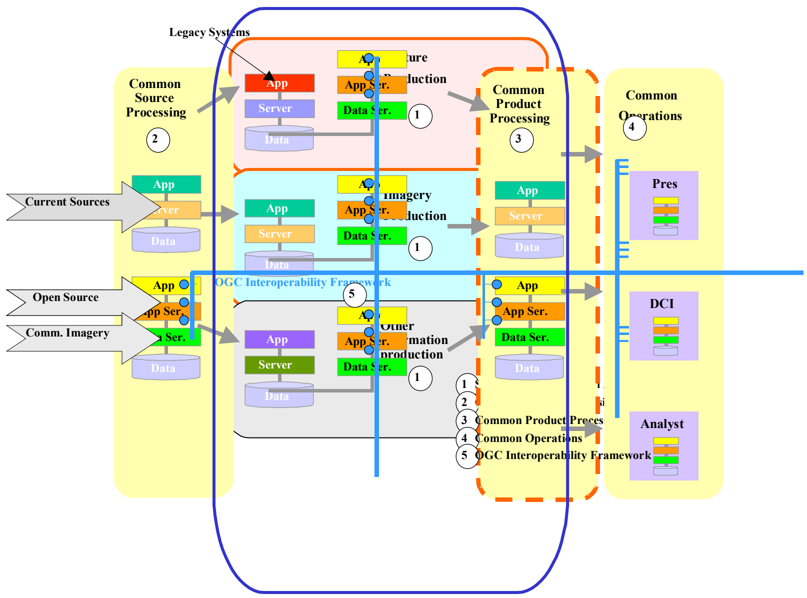



**Figure 32 - Enterprise Viewpoint: Production Detail**

#### 7.1.1.1 Requirements

The fundamental requirement in the scenario is the on-time delivery of FFI into managed FFI Data Stores. Specific requirements that spin from this include:

- There must exist a detailed product specification for FFI that includes
    - A feature capture specification including
        - Rules of Inclusion:
            - Explicit and comprehensive instructions for the decision to declare a new instance of a feature type within the FFI world view
            - Measurement thresholds, material types, shape types
        - Rules of Representation:
            - Type/classification/taxonomy
            - Type names

        - Application schema
        - Attributes and the procedure for the assignment of attributes
        - Relationships, and the procedure for the capture and representation of relationships to other objects
        - Geometry types; geometry collection process
        - Topology types, collection processes?
        - Metadata (source lineage, quality)
    - Update Rules for change of:
        - Type, attribute, geometry, topology, metadata, relationship, portrayal, quality, and authority
- Sufficiently accurate, and sufficiently high-resolution stereoscopic imagery covering a sufficiently large footprint including the area of interest to support the collection of FFI.
- Access to legacy FFI
- The ability to manage raw and processed imagery and FFI

#### 7.1.1.2 Behavior Walkthrough

There are three major steps in the Production Scenario. These three steps can be chained to form a single, more coarse-grained, step that will be discussed later.

The first step involves the activities that lead from uncontrolled imagery to sufficiently georeferenced imagery. "Sufficiently georeferenced" implies the achievement of ground-to-image functions possessing the positional accuracy and quality needed to be in conformance with FFI production standards. The first-step activities include the observation and measurement of control points and ground points, establishing a sufficiently rich bundle of co-linearity and co-planarity equations, including parameters that specify correction models for camera motion and lens errors and other sources of error. The step concludes with the exercise of a bundle adjustment to georeference the images. That is, for each image, j, there is created a function $f_j$ that maps from ground coordinates to image coordinates. We assume that the union of the stereo footprints covers the area of interest.

The second major step is to exploit the imagery (and possibly other sources) to extract (or update) features within the FFI worldview. Included here are activities associated with the creation of a stereoscopic view of the area of interest, the overlaying of legacy information, the human cognition of instances of features belonging to the FFI world view, and the steps that populate the newly discovered features with geometry, topology, attributes, and relations, in accordance with FFI specifications. Sources other than imagery (such as gazetteers) may contribute to this step as well. The second step ends when all of the observed instances of FFI features in the area of interest have been captured, and are ready to include in a new or updated FFI product.

The third step is the iterated application of a service that we call "Add Next Feature." Using the concept of this service, we may simplify the second step to the special case where imagery is used to discover a single new feature. In the third step, this feature with all its attributes is integrated or "fused" into a pre-existing FFI base. Iterating this step over all the features discovered in step 2 results in FFI ready for managed storage.

The situation is summarized in Figure 33.

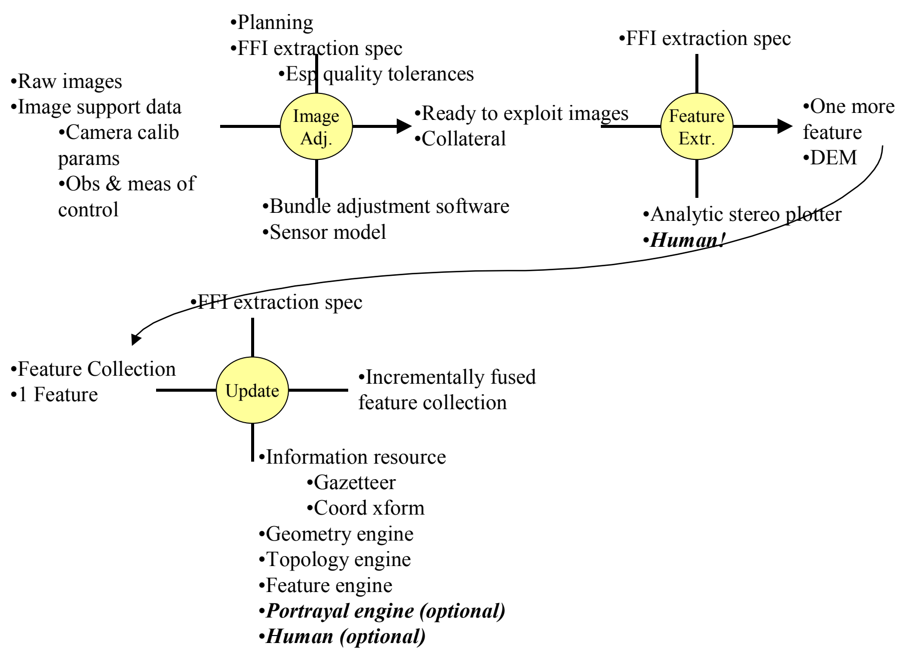



**Figure 33 - Feature Extraction and Update**

### 7.1.2 Information Viewpoint

The information viewpoint of the Production Scenario is best understood by building a service model for each of the fundamental services in the scenario. The General Service Model is provided at Figure 34. Using this model, each service partitions information into a family of "information types." Each information type is characterized by it semantics and its data structure. Complex information types may be built in hierarchies from more primitive information types. The terms "Class," "Type," "Subtype," "Element," "Relationship", and "Property" may be used to characterize information types in order of complexity from most complex to most simple.

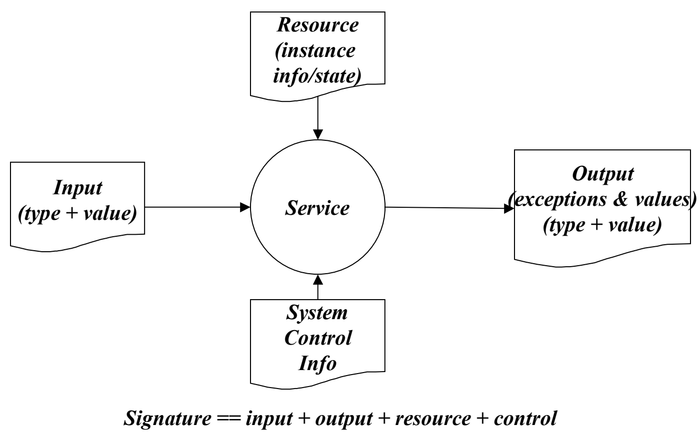



**Figure 34 - General Service Model**

Information types can be organized into four categories for each service. There are information types in support of:

- Service Input
- Service Output
- Service Resources, and
- Service Control

Figure 35 illustrates these information types for Feature Production.

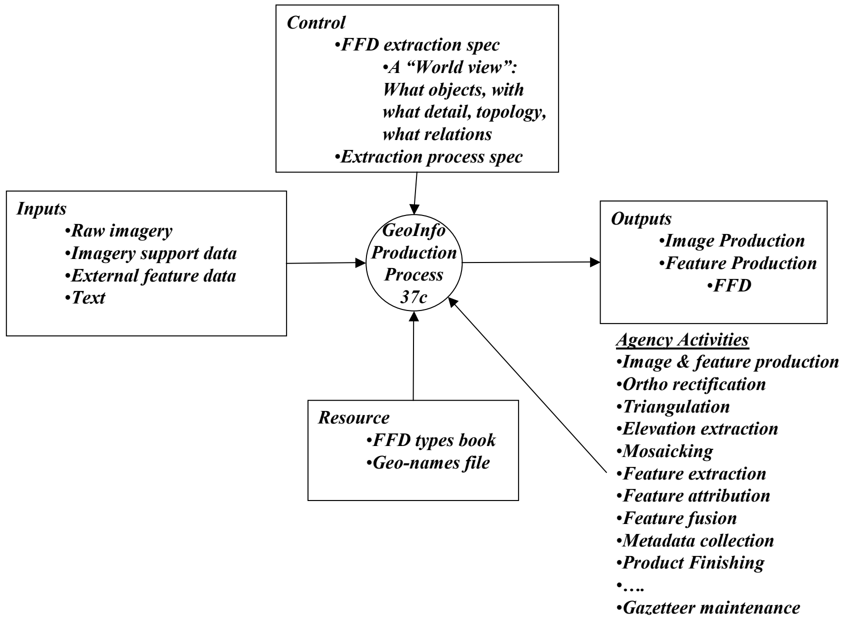



**Figure 35 - Production Model**

Especially important to GIS are information types in support of the OGC Feature Model, Image Model, and Coverage Model.

#### 7.1.3 Computational Viewpoint

Like the other viewpoints, the computational viewpoint may be presented at multiple levels of granularity. For example, one may conceptualize a computation at the level of:

- Instance
- Service
- Interface
- Operation
- Parameter (Message), and
- Type

Figure 36 shows simplified computational views for two operations: Generate DEM (Digital Elevation Model) from imagery source, and Geometry Extraction using a digital stereoscopic environment. In both operations, the inputs are shown at the left, the outputs at the right, the control at the top, and the resources at the bottom. One way to distinguish between "control" and "resource" is that "control" usually persists and may be used in following operations, while "resources" (like manual labor hours or hourly use of a scarce device) are consumed in the course of the operation and must be replenished.

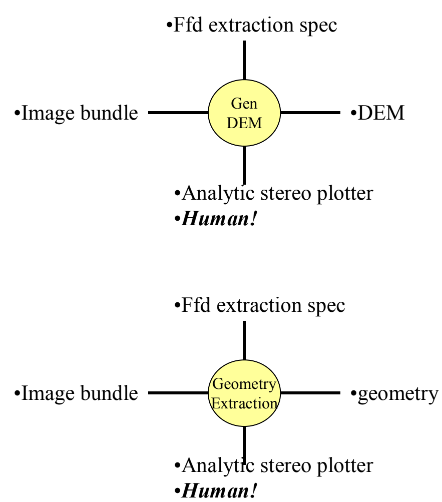



**Figure 36 - Feature Extraction**

The following outline represents the rest of the Computational Viewpoint.

- *Adjustment are italics intentional?*
  - *Establish control*
    - *Georeferenced ground points, in-flight GPS, and/or sensor ephemeris*
  - *Triangulate*
    - *Ground coordinates = f(image coordinates, sensor parameters, exterior orientation parameters)*

      - *Distortion removal*
        - *Radiometric*
        - *Geometric*
- *Info Extraction*
  - *Elevation (DEM) extraction*
  - *Mosaicking*
  - *Feature creation*
    - *Type extraction*
    - *Geometry extraction*
    - *Attribute extraction*
    - *Relations extraction*
    - *Topology extraction*
  - *Lookup name (gazetteer)*
  - *Identify position*
- *Update*
  - *Identify name*
  - *Identify position*
  - *Transform coordinates*
  - *Update geometry, topology, feature*
  - *Portray*

### 7.2 Information Broker Scenario

The Greater Chitroit Council of Governments (GCCOG) is cooperating with the Greater Chitroit Chamber of Commerce and Board of Realtors to solicit plans from major developers for a large multi-purpose development according to the GCCOG Long Range

Development Plan. The intentions are to increase development of light industrial, retail, and single family housing in areas of the region that are in need. Several alternatives are proposed.

The use case starts when the GCCOG receives a Regional development plan from a major developer with alternative locations for implementing the plan. GCCOG evaluates the plan against a set of econometric, traffic, and environmental models. To evaluate the proposals, the GCCOG has adopted an econometric model and a traffic prediction model. Each member of the GCCOG will use visualization and portrayal tools and searches to compare the outcomes of various alternatives. Similarly, the GCCOG will be using environmental parameter data and environmental models to determine impacts of the development. Additionally, the plan is evaluated against GCCOG long-range development plan.

The GCCOG has retained GSTIM, Inc, to develop a State Spatial Data Infrastructure that is interoperable with the National Spatial Data Infrastructure. This RSDI will link local and state information providers together (like the participants in the NSDI) in an environment where data and services can be shared.

### 7.2.1 Enterprise Viewpoint

#### 7.2.1.1 Requirements

Figure 37, shows the relationship between NSDI, RSDI and GCOOC.

1. The RSDI must seamlessly supply inputs to the quantitative modeling:

   - Access to data available in the NSDI (connecting to an existing national infrastructure):
     - Federal, State, and Local DOT transportation data
     - Federal, State, and Local EPA environmental parameters
     - Federal, State, and Local demographic data
   - A set of distributed data to be available in the RSDI:
     - Allow local entities to publish their data holdings: detailed local transportation, environmental, and demographic data; a geospatial representation of the current long-range development plan, etc.
     - Enable the use of community domain models to support unambiguous information sharing

- o Market Research data from Private sector under a regional licensing arrangement with Market Cranium. This requires authentication and payment.

2. The RSDI must supply access to tools for qualitative analysis, such as two- and three-dimensional portrayals of plan alternatives and model outputs.

3. The RSDI must supply dynamic discovery capability in support of the qualitative environmental analyses.

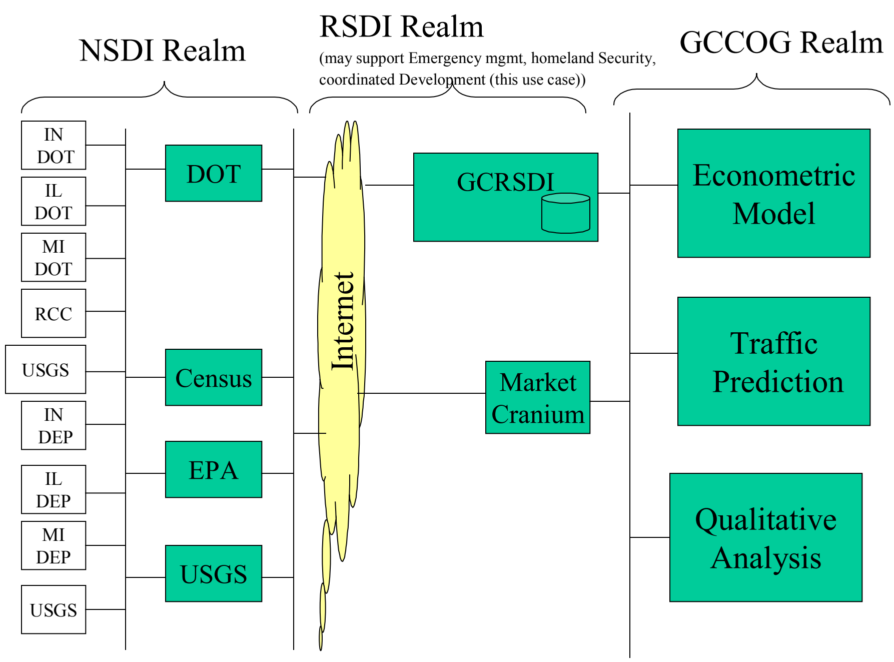



**Figure 37 - NSDI, RSDI and GCCOG as an Enterprise**

#### 7.2.1.2 Pre-Conditions

1) Local entities publish their information with the RSDI.

2) GCCOG applications can seamlessly access RSDI and NSDI

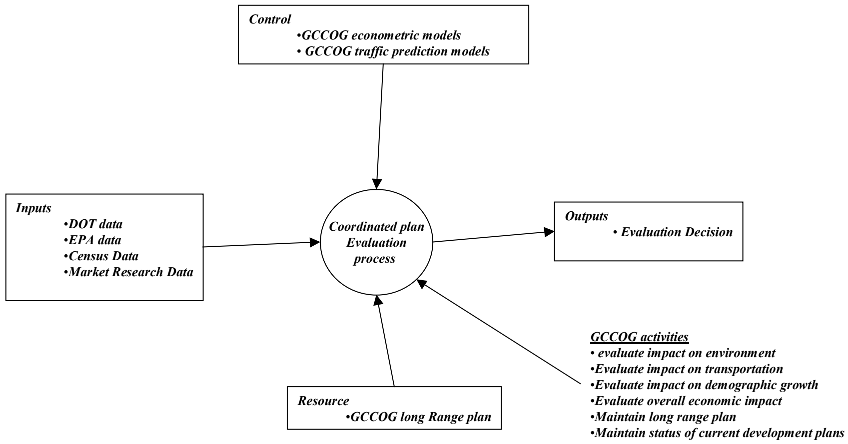



**Figure 38 - GCCOG Proposal Evaluation Process**

#### 7.2.1.3 Behavior Walkthrough

An overview of the GCOOG proposal evaluation process is shown in Figure 38. The GCCOG planning team runs the Growth Model for each development alternative by providing various parameters derived from examining the plan detail:

1) The system searches NSDI to locate and access regional scale demographic and transportation infrastructure data for input to the growth model.

2) The system searches RSDI to locate and access local scale demographic, historical traffic flow, and other data for input to the growth model.

3) The system accesses the market research data for input to the econometric model micro charging for the data delivered.

4) The system accesses elements of the long range plan representation to provide the remaining set of model inputs.

5) The results (the spatial and temporal distribution of population, tax revenue, employment, traffic flow, and infrastructure development costs) are stored back into the RSDI along with the associated model run metadata.

GCCOG members perform Qualitative Environmental Analysis by browsing multi-media views of expected impacts on land, air, and water resources, and residential and recreational spaces:

1) Users search NSDI to collect updated environmental parameters from EPA, and stock orthophotos from USGS.

2) Users search RSDI for high-resolution processed imagery, imagery support, detailed terrain data, and other inputs to realistic 3D landscape views.

3) Users search RSDI for outputs from prior runs of the growth model, including forecasts of housing densities, traffic levels, property values, etc.

4) Users search RSDI for links among various related views.

5) Users apply a variety of tools to visualize expected impacts on water and air quality, wildlife habitat, recreational opportunities, etc.

6) Users establish additional links between views, and publish these back to RSDI.

7) Users search RSDI to find visualization and analysis services that they can apply to relevant datasets.

8) RSDI provides the user with access to these visualization and analysis services to generate additional views that will assist in the final decision.

#### 7.2.1.4 Post-Conditions

GCCIG either accepts the plan, rejects it, or requests from the developer, changes in the proposal.

### 7.2.2 Information Viewpoint

The GCCOG's decision will require many different kinds of feature or coverage information. Clear and complete content models for these data need to be shared between suppliers of related data, and available to GCCOG users. Examples of feature and coverage information content follow:

1) From the Greater Chitroit Council of Governments:

   a) Regional development plan

   b) Anticipated regional land use over time

2) From providers of Commercial Market research:

   a) Industrial and commercial facilities

b) Growth rate, income levels, buying power, etc.

3) From the Regional Chamber of Commerce

a) Long Range Plan

b) Status of current development projects

4) From the Federal, State, and Local Census:

a) Demographics, Socio-economic indicators, Housing

5) From Federal, State, and Local DOTs:

a) Transportation status and forecasts

6) From Federal, State, and Local Environmental agencies:

a) Environmental and public health data

Imagery content needed:

? Aerial Imagery, Orthophotos, etc. for showing land use and land use change.

? National Elevation Database for terrain views and visual impact analysis.

? Metadata content models are needed for the Regional and National Clearinghouses of Spatial Data.

The RSDI would support richer information sharing than NSDI, thanks to more detailed data and metadata models shared defined by consensus among RSDI stakeholders. (Thanks to tighter coupling, and shared agreements groups in the Chitroit region have been able to define a consensus that describes much more than the “Lowest common denominator” model available at the loosely-coupled NSDI.) RSDI content models would be profiles or extensions of NSDI models.

#### 7.2.3 Computational Viewpoint

In this view, we map OGC services to the services required by this scenario. Choosing among these regional planning options requires the following services:

- Data Services
    - Read features and coverages in RSDI and NSDI (representing status, forecasts, and context).

        - Rich query capabilities would facilitate access to large, complex data collections, e.g., detailed Census data tables.
        - Gazetteer services would serve to translate place names into geospatial queries.
    - Add / Update / Delete features and coverages in RSDI (for authorized persons or groups).
- Application Clients
    - 2D maps, 3D terrain views, animations, charts, alerts
    - The user would need access to many visualization parameters, so as to "explore" various multimedia representations of data and forecasts
- Processing Services
    - Coordinate systems transformation
    - Econometric models and a traffic prediction models
    - Feature extraction from an image
    - Cross-tabulation between parameters (e.g., job impacts by skill level) to create derived features and coverages
    - Other manipulations, usually performed on the fly; and possibly chained
    - As a service: Invoke the growth / traffic / environmental models using user-supplied parameters.
    - As software applet or standalone application.
- Encodings
    - Into different data structures (e.g., bundle features as a coverage)
    - Into different encoding (e.g., GML)
- Data Services and Registry Services
    - Search RSDI and NSDI for data
    - Search RSDI and NSDI for services that treat these data
    - Invoke (bind to) these services as needed
    - Propagate search requests from RSDI to NSDI
    - Add / Update / Delete metadata in RSDI for new or changed data and services

## 7.3 Information Consumer Scenario

The following *consumer travel assistance scenario* features a typical application involving OGC's OpenLS services.

### 7.3.1 Enterprise Viewpoint

#### 7.3.1.1 Requirements

1) Must provide the means for a subscriber to invoke and interact with a "Personal Navigation Service".

2) Must provide the means for the Personal Navigation Service to acquire the position of a mobile terminal off the network (use OpenLS Gateway Service).

3) Must provide the means for a subscriber to determine an address for a dental office, given only a phone number (use OpenLS Directory Service).

4) Must provide the means for a subscriber to determine the location of the dental office, given its address (use OpenLS Geocoder Service).

5) Must provide the means for a provider to determine the traffic conditions for the area of interest (use Traffic Service - future OpenLS).

6) Must provide the means for a provider to determine the weather conditions for the area of interest (use WFS-based Weather Data Service).

7) Must provide the means for a provider to determine the road construction conditions for the area of interest (use WFS-based Road Construction Data Service).

8) Must provide the means for a subscriber to determine the route from their current position to the dental office (use OpenLS Route Determination Service).

9) Must provide the means for a subscriber to receive travel advisories that impact their route (use Travel Advisory Service - future OpenLS).

10) Must provide the means for a subscriber to view their route on a map, the maneuvers associated with the route, turn-by-turn directions, and travel advisories (use OpenLS Presentation Service).

11) Must provide the means to track a subscriber's position in order to determine their in-transit status, and changing traffic conditions (use Tracking Service – future OpenLS).

#### 7.3.1.2 Pre-Conditions

A consumer is traveling to a dental appointment and the wireless service provider is offering travel assistance through a Personal Navigation Service. At the onset, the subscriber only knows the phone number of the dentist. The consumer is equipped with a location-enabled cell phone, sees that a major storm is approaching, and wants to obtain routing information from the provider. The routing information is updated to reflect current traffic and weather conditions.

#### 7.3.1.3 Behavior Walkthrough

The flow of events for this scenario is captured in the following figures and descriptive text. Figure 39 illustrates a high-level functional view of the operational environment for the wireless service provider and subscriber. Figure 40 provides a more detailed view of the components supporting the Personal Navigation Service.

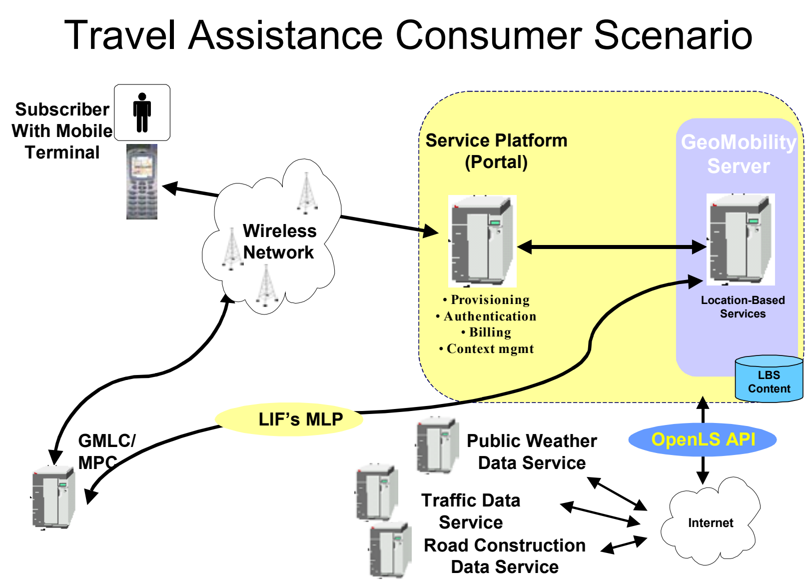



**Figure 39 - Travel Assistance Consumer Scenario**

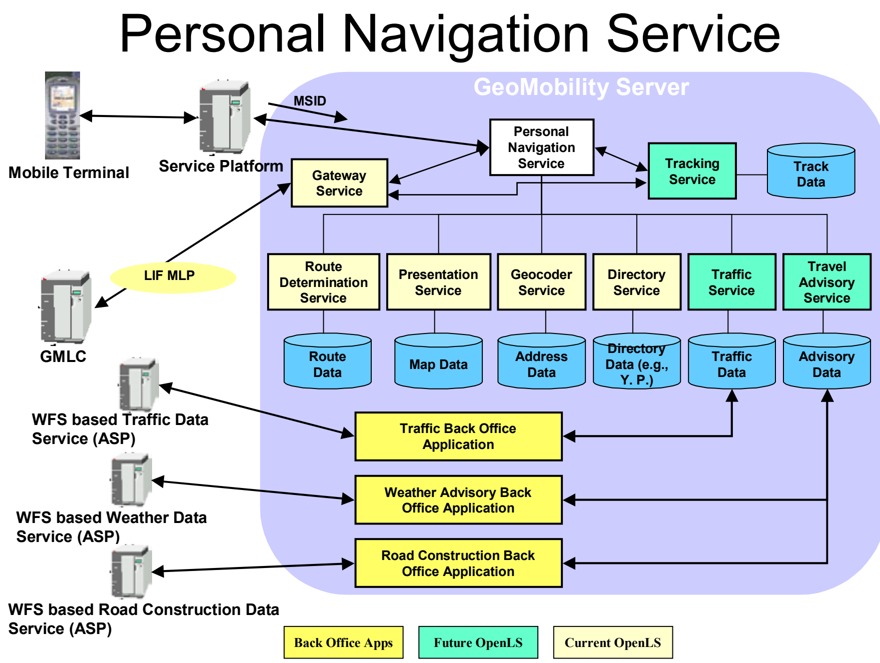



**Figure 40 - Details of Personal Navigation Service**

## Scenario Walkthrough:

OGC Information Framework components are presented in *italics* below. Service (Computational) components are presented in **bold**.

The scenario involves a wireless service provider offering location-based services, including route determination and display, route guidance (maneuvers and advisories), traffic reporting, and weather reporting. The subscriber seeks travel advice from the provider via a **Personal Navigation Service** available through their cell phone. This is an application service that utilizes OpenLS Core Services (Gateway, Directory, Geocoder, Presentation, and Route Determination), and other services, as portrayed in Figure 40.

1) **Personal Navigation Service** invokes a **Gateway Service** (LIF MLP) for *subscriber's current position*, given *mobile subscriber ID* (MSID).

2) **Personal Navigation Service** allows the subscriber to designate their destination by entering the phone number of the dentist. Then the service invokes a **Directory Service** to determine *subscriber's destination address*, the dental office.

3) **Personal Navigation Service** invokes a **Geocoder Service** to resolve *subscriber's destination address* into *subscriber's destination location* (coordinates).

4) **Personal Navigation Service** invokes a query against a **Traffic Service**, which in turn relies on a **Traffic Data Service** provided by an ASP for *current traffic conditions*.

5) Service provider queries a **Public Weather Data Service** provided by an ASP to obtain *weather data*, which is required in support of the **Travel Advisory Service.**

6) **Personal Navigation Service** queries a **Travel Advisory Service** for current and predicted *weather conditions* for a *region encompassing consumer's current position and destination location*.

7) Service provider queries **DOT Road Construction Data Service** to obtain *road construction data*, which is also required to support the **Travel Advisory Service.**

8) **Personal Navigation Service** queries a **Travel Advisory Service** that reports *advisories* based on *road construction data* for *region encompassing subscriber's current position and destination location*.

9) **Personal Navigation Service** applies *traffic conditions*, *road construction*, and *meteorological information* to a **Route Determination Service**, which returns the *best route,* with *advisories* and *maneuvers*.

**10)** **Personal Navigation Service** invokes **Presentation Service** to present the *route* overlayed on a *map* on the Subscriber's mobile terminal.

**11)** Subscriber proceeds along *route*, and the **service** issues *maneuvers* and *travel advisories* through the **Presentation Service**.

**12)** **Personal Navigation Service** tracks *subscriber's current location*, via a **Tracking Service**, and periodically issues near-real-time *travel advisories* based on changing *traffic conditions* and *weather*, through the **Travel Advisory Service**.

#### 7.3.1.4 Post-Conditions

The subscriber obtains visual and audible cues to negotiate an optimal route to the dental office. The cues include navigational guidance along route optimizes for traffic and weather conditions, as well as travel advisories.

### 7.3.2 Information Viewpoint

1) Subscriber’s Current Position (OpenLS Position ADT)

2) Subscriber’s Destination Address (OpenLS Address ADT)

3) Subscriber's Destination Location (OpenLS Location ADT)

4) Region encompassing consumer's current location and destination location (OpenLS AOI ADT)

5) Current traffic conditions (Future: OpenLS Traffic ADT)

6) Weather Data (GML)

7) Road Construction Data (GML)

8) Advisories (Future: OpenLS Travel Advisories ADT)

9) Best Route (OpenLS Route Summary ADT & Route Geometry ADT)

10) Maneuvers (OpenLS Route Maneuvers List ADT)

11) Map for the area of interest (OpenLS Map ADT)

#### 7.3.3 Computational Viewpoint

In this view, we map OGC services (**Error! Reference source not found.**) to the services required by this scenario. The scenario requires the following services:

1) Application services
   a) Personal Navigation Service
2) Data services
   a) OpenLS Directory Service
   b) OpenLS Gateway Service
   c) OpenLS Travel Advisory Service (future)
   d) OpenLS Traffic Service (future)
   e) Road Construction Data Service (WFS based)
   f) Public Weather Data Service (WFS based)
   g) Traffic Data Service (WFS-based)
3) Portrayal services
   a) OpenLS Presentation Service

4) Processing services

   a) OpenLS Route Determination Service

   b) OpenLS Geocoder Service

   c) OpenLS Tracking Service (future)

# Annex A: References

This section contains those OGC Specifications that are referenced in this OGC Reference Model.

### 7.1 OpenGIS Abstract Specifications

[1] Topic 1 - Feature Geometry

[2] Topic 2 - Spatial Reference Systems

[3] Topic 6 - The Coverage Type

[4] Topic 7 - Earth Imagery

[5] Topic 11 - Metadata

[6] Topic 12 - The OpenGIS Service Architecture

Not referenced:

Topic 3 - Locational Geometry
Topic 4 - Stored Functions and Interpolation
Topic 5 - The OpenGIS Feature
Topic 8 - Relations Between Features
Topic 9 - Accuracy
Topic 10 - Feature Collections
Topic 14 - Semantics and Information Communities
Topic 13 - Catalog Services
Topic 15 - Image Exploitation Services
Topic 16 - Image Coordinate Transformation Services

### 7.2 OpenGIS Implementation Specifications

[7] OpenGIS® Simple Features Specification for OLE/COM

[8] OpenGIS® Simple Features Specification for CORBA

[9] OpenGIS® Simple Features Specification for SQL

[10] OpenGIS® Catalog Interface Implementation Specification Version 1.1

[11] OpenGIS® Grid Coverages Implementation Specification

[12] OpenGIS® Coordinate Transformation Services Implementation Specification

[13] OpenGIS® Web Map Service Interfaces Implementation Specification Version 1.1.1

[14] OpenGIS® Geography Markup Language (GML) Implementation Specification Version 3.0

[16] OpenGIS ® Web Coverage Service Interface Specification, Version 0.0

### 7.3 OpenGIS Recommendation Papers

Observations and Measurements

### 7.4 OpenGIS Discussion Papers

[15]

[16]

[17] OGC Web Map Service Implementation Specification Part 2: XML for Requests using HTTP POST

[18] OGC Web Terrain Server (WTS)

[19] Units of Measure and Quantity Datatypes

[20] Location Organizer Folder Specification

[21] Gazetteer Service Specification

[22] Geoparser Service Specification

[23] Geocoder Service Draft Candidate Implementation Specification

[24] Web Registry Server

[25] Basic Services Model

[26] XML for Imagery and Map Annotations

[27] High-Level Ground Coordinate Transformation Interface

### 7.5 ISO TC211 References

[28] ISO 19109 Geographic information — Rules for application schema

[29] Other ISO TC211 specs need to be listed here.

### 7.6 Other References

[30] Reference Model of Open Distributed Processing [ISO/IEC 10746].

[31] "Summary Report of Version 1.0 of the Bureau Architecture", Bureau of Land Management, 31 March 2000, (http://www.blm.gov/ba/pubs/BA_Summary_v1.pdf), BUREAU OF LAND MANAGEMENT

[32] ISO/IEC 13235-1:1998. *Open Distributed Processing – Trading Function: Specification*.

[33] OLS Reference Architecture, April 2002.

[34] ISO 19121, Imagery and Gridded Data, April 2000.

[35] ISO 19125-1, Geographic Information- Simple feature access- Part 1: Common Architecture

# Annex A Mapping ORM requirements to OGC services

| Topic 12 High Level | Topic 12 Sub-Requirement | OGC SPECS | ORM Requirements |
|---|---|---|---|
| Human Interaction | Catalog Viewer | Catalog | Visualization |
| Human Interaction | General Client Viewer | WMS, WFS, WTS, WCS, GML, OpenLS Presentation Service (Display Map w/ Overlays, Route Vectors, Display Route Directions) | Visualization |
| Human Interaction | Viewer - Animation | None | Visualization |
| Human Interaction | Viewer - Mosaicking | Partially with WMS/WCS/GML. | Visualization |
| Human Interaction | Viewer - Perspective | WTS | Visualization |
| Human Interaction | Imagery viewer | WMS, WCS, Coverage Portrayal Service | Visualization |
| Human Interaction | Spreadsheet viewer | None directly. However, could be supported by WFS and GML. | Visualization, reporting |
| Human Interaction | Service editor | None | Collaboration, Discovery, Business process modeling and workflow |
| Human Interaction | Chain definition editor | None | Collaboration, Discovery, Business process modeling and workflow |
| Human Interaction | Workflow enactment | None | Business process modeling and workflow |
| Human Interaction | Geographic Symbol Editor | Symbol Management Service; supported by SLD. | Map production, visualization |
| Human Interaction | Geographic Feature Editor | None directly. However, core functional requirements are met by WFS (transactions), SLD, XIMA, GML, and | Map production, visualization, analysis |

| Topic 12 High Level | Topic 12 Sub-Requirement | OGC SPECS | ORM Requirements |
|---|---|---|---|
| | | WTS. | |
| Human Interaction | Feature generalization Editor | None directly but WFS, GML, and SLD could be used in support. | Analysis, map production, fusion |
| Human Interaction | Data Structure Viewer | None directly but Catalog, Service Registry, and GML could be used in support | Analysis, visualization, fusion, and collection and assessment |
| Info. Management | Feature Access | Registries, Catalog, SFA, WFS, GML, and Filter | Data access, analysis, collaboration, collection and assessment, dissemination, |
| Info. Management | Map access | Registries, Catalog, WMS, SLD, and WTS | Publishing, discovery, dissemination |
| Info. Management | Coverage Access | Registries, Catalog, Grid Coverage, WMS, SLD, WCS, GML | Publishing, discovery, dissemination |
| Info. Management | Coverage Access – Sensor | SensorML, Sensor Collection Service, Catalog | Publishing, discovery, dissemination |
| Info. Management | Product Access | None directly but could be based on WMS, WFS, GML, SLD, and Catalog | Publishing, discovery, dissemination |
| Info. Management | Feature Type | Registries | Discovery |
| Info. Management | Catalog Service | Catalog | Discovery |
| Info. Management | Registry Service | Gazetteer, Registry Service | Discovery |
| Info. Management | Gazetteer | Gazetteer | Discovery |
| Info. Management | Order Handling | Sensor Planning Service | Customer support |
| Info. Management | Standing order | Sensor Planning Service | Customer support |

| Topic 12 High Level | Topic 12 Sub-Requirement | OGC SPECS | ORM Requirements |
|---|---|---|---|
| Task Management | Chain definition | Work in OWS is highly relevant. This includes the OGC General Services Model | Collaboration |
| Task Management | Workflow enactment | None | Collaboration |
| Task Management | Subscription Service | None | Data Access, Collaboration |
| Processing - Spatial | Coordinate Conversion | CT and related documents such as UOM and CT Data Definition | Analysis |
| Processing - Spatial | Coordinate Transformation | CT and related documents such as UOM and CT Data Definition | Analysis |
| Processing - Spatial | Coverage/Vector Conversion Service | None. Could utilize WFS, WCS, and GML | Analysis |
| Processing - Spatial | Coordinate Conversion Service | Image Coordinate Transformations. | Analysis |
| Processing - Spatial | Rectification Service | None. Use of SensorML, Imagery Handling. | Analysis |
| Processing - Spatial | Orthorectification Service | None. Use of SensorML, Imagery Handling. | Analysis |
| Processing - Spatial | Sensor Geometry Model | SensorML. | |
| Processing - Spatial | Image Geometry Model | SensorML, Observations & Measurements. | |
| Processing - Spatial | Sub-setting Service | None, but could be built using WFS, GML, and Filter | Analysis |
| Processing - Spatial | Sampling Service | None, but could be built based on WFS, GML, and Filter | Analysis |
| Processing - Spatial | Tiling Change Service | None | |
| Processing - Spatial | Dimension Measurement Service | None | Analysis |

| Topic 12 High Level | Topic 12 Sub-Requirement | OGC SPECS | ORM Requirements |
|---|---|---|---|
| Processing - Spatial | Feature Manipulation | None | Analysis |
| Processing - Spatial | Feature Matching | None | Analysis |
| Processing - Spatial | Feature Generalization | None | Analysis |
| Processing - Spatial | Route Determination | OpenLS Route Determination Service | Analysis |
| Processing - Spatial | Positioning Service | OpenLS Gateway Service | Analysis |
| Processing - Spatial | Proximity Service | OpenLS Directory Service | Analysis |
| Processing - Thematic | Geoparameter calculation | None | Analysis |
| Processing - Thematic | Thematic Classification | None currently but could be built based on capabilities in WFS and WCS. | Analysis |
| Processing - Thematic | Feature Generalization | None | Analysis |
| Processing - Thematic | Sub-setting service | WFS, Filter, OpenLS Directory Service | Analysis |
| Processing - Thematic | Spatial Counting | None currently but could be based on WFS, Filter, and GML | Analysis |
| Processing - Thematic | Change Detection | None. | Analysis |
| Processing - Thematic | Geographic Information Extraction | None | Analysis |
| Processing - Thematic | Image Processing | Imagery Handling, SensorML, Sensor Collection Service, Sensor Planning Service | Analysis |
| Processing - Thematic | Reduced Resolution Generation | Imagery Handling | Analysis |

| Topic 12 High Level | Topic 12 Sub-Requirement | OGC SPECS | ORM Requirements |
|---|---|---|---|
| Processing - Thematic | Image manipulation | Imagery Handling | Analysis |
| Processing - Thematic | Image understanding | None | Analysis |
| Processing - Thematic | Image Synthesis | None | Analysis |
| Processing - Thematic | Multi-band Image Manipulation | Imagery Handling | Analysis |
| Processing - Thematic | Object Detection Service | None | Analysis |
| Processing - Thematic | Geoparsing Service | Geoparser | Analysis |
| Processing - Thematic | Geocoding Service | OpenLS Geocoding/Reverse Geocoding | Analysis |
| Processing – Temporal | Temporal Reference System Transformation | None, although UOM and GML/temporal could provide the basis. | Analysis |
| Processing – Temporal | Sub-setting | Imagery Handling | Analysis |
| Processing – Temporal | Sampling | Imagery Handling | Analysis |
| Processing – Temporal | Temporal Proximity | None | Analysis |
| Processing - Metadata | Statistical Calculation | None | Analysis |
| Processing - Metadata | Geographic Annotation | XIMA for annotations. Registries. | Visualization, publishing |
| Communication | Encoding Service | None. Use OGC encodings: GML 2.1, XLS, SensorML, XIMA, Observations & Measurements, LOF | Messaging and communications, Dissemination |
| Communication | Transfer Service | None. Use OGC encodings: GML 2.1, XLS, SensorML, XIMA, Observations & Measurements, LOF | Messaging and communications, Dissemination |

| Topic 12 High Level | Topic 12 Sub-Requirement | OGC SPECS | ORM Requirements |
|---|---|---|---|
| Communication | Geographic Compression | None | Messaging and communications, Dissemination |
| Communication | Geographic Format Conversion | None. Use OGC encodings: GML 2.1, XLS, SensorML, XIMA, Observations & Measurements, LOF | Messaging and communications, System Integration |
| Communication | Messaging Service | Web Notification Service (new under OWS 1.2) | Messaging and communications |
| Communication | Remote file and executable management | Cascading features of WMS and WFS. | Messaging and communications |