SEMANTIC LINKAGE OF THE INVISIBLE GEOSPATIAL WEB

Francisco Javier López Pellicer

PhD DISSERTATION

RESEARCH ADVISORS

Dr. Rubén Béjar Hernández
Dr. Francisco Javier Zarazaga Soria

February 2011

Computer Science and Systems Engineering Department
Universidad de Zaragoza

– To Juan José, Josefina and Rafael.

They cheer me on even though sometimes it was difficult to explain what in the world I was doing.

Acknowledgments

I would also like to thank the members and ex-members of the *Advanced Information Systems Laboratory* (IAAA) of the Computer Science and System Engineering Department of the University of Zaragoza, and its spin-off *GeoSpatiumLab*, for introducing me to the Geospatial Web and its problems. I am thankful to Prof. Pedro R. Muro-Medrano and Dr. F. Javier Zarazaga-Soria. I should also include Aneta, Javier N., Javier L., Rubén, Maria José (MJ), Silvia, Raquel, Jesús, Covadonga, Íñigo, Rodolfo, David, Miguel, Miguel Ángel, José Miguel, Alberto, Christian, Rocio, Esperanza, Fernando, Walter, Alex and Oscar. I hope not forget anyone, but I have a *little* problem with proper names.

I am grateful to the members and ex-members of the *XLDB Research Team* at LaSIGE, led by Prof. Mário J. Silva and Dr. Francisco Couto, of the Department of Informatics of the University of Lisbon that I met during my research stay. The Geographic Knowledge Base developed by XLDB has been the testing ground for much of the content of this thesis. I am thankful for the opportunity to work with them, especially to Mário, Francisco, Ana, Bruno, David, Nuno, Daniel and Patricia (and my lab-mates Luis Filipe, Catia and Hugo). I hope (again) not forget anyone.

I also recall the experience of collaborating with the *National Geographic Institute of Spain* (IGN) and the *Zaragoza City Council* in their respective Spatial Data Infrastructures. This experience have opened my mind to unsuspected issues.

Finally, I would like to thank the people that have reviewed this thesis. Despite all of their help, I take full responsibility for any errors or omission herein.

Resumen ejecutivo

Antecedentes. El acceso efectivo a la información geoespacial adquiere una importancia fundamental en una economía basada en el conocimiento. Es esencial para muchas actividades tener acceso a los recursos geoespaciales a través de la Web. Desde la respuesta a un desastre a la decisión de desarrollar un nuevo negocio, el acceso a información geoespacial actualizada ofrecida a través de Web Geoespacial puede ser la diferencia entre el éxito y el fracaso. Se define la Web Geoespacial como la colección de servicios Web, datos geoespaciales y metadatos que permiten el uso de datos geoespaciales en una amplia gama de aplicaciones de dominio.

Esta tesis trata tres problemas relacionados con la Web Geoespacial. En primer lugar, los servicios estándar de la Web Geoespacial forman parte de la Web invisible. La literatura utiliza el término invisible para identificar la parte de la Web ignorada por los motores de búsqueda (invisible Web, Sherman and Price, 2001). Los usuarios comunes pueden estar al tanto del contenido de la Web invisible sólo por casualidad. A continuación, esta tesis clasifica el contenido de la Web Geoespacial accesible a través de servicios estándar como parte de la Web profunda (deep Web, Bergman, 2001). El término profunda destaca que un contenido invisible tiene suficiente valor en si mismo para ser tratado como si estuviera en la superficie de la Web, en otras palabras, merece ser indexado por un motor de búsqueda. Los contenidos de la Web Geoespacial son parte de la Web profunda por su propia naturaleza. Es decir, los contenidos de la Web Geoespacial se ocultan detrás de servicios Web. Por último, los servicios estándar y contenidos de la Web Geoespacial están socialmente desconectados del resto de la Web. Los servicios públicos de la Web Geoespacial no están concebidos para servir a su funcionalidad a los consumidores de servicios y, *al mismo tiempo,* como herramienta social para la comunicación entre los agentes relacionados con los servicios. Actualmente, la vinculación de los servicios Web con la iniciativa de Datos abiertos enlazados[1] (Linked Open Data, Bizer et al., 2009), una de las materializaciones de la Web Semántica (Berners-Lee et al., 2001), parece prometer dicha posibilidad.

Objetivo. Los problemas derivados de la invisibilidad de los servicios Web Geoespaciales podría mitigarse si es posible realizar un rastreo sistemático y enfocado de la Web en busca de dichos

[1] `http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

servicios en combinación con la publicación como Datos enlazados (Linked Data) de las descripciones y los contenidos de los servicios descubiertos.

Esta tesis analiza la viabilidad de una solución basada en el desarrollo de una araña Web (Web crawler) enfocada a los servicios Web geoespaciales, y el acceso a la información sobre los servicios descubiertos como descripciones procesables por máquinas. Además, la solución analizada deberá permitir el acceso a contenido geoespacial remoto en el mismo formato.

Este enfoque requiere el desarrollo de dos ontologías de dominio. La primera ontología describe la arquitectura abstracta de los servicios Web geoespaciales. La segunda ontología describe el contenido geoespacial.

Ámbito. El rastreo se restringe a los servicios Web compatibles con las especificaciones de Open Geospatial Consortium (OGC). OGC lidera el desarrollo de especificaciones abiertas y estandarizadas interfaz de servicios Web para acceder a la información geoespacial desde 1994. Más de 400 compañías, agencias gubernamentales y universidades son miembros de OGC, y participan en los procesos de consenso para desarrollar a disposición del público las normas de interfaz de servicios Web para el acceso a los datos geoespaciales. Muchas especificaciones OGC que definen la interfaces de servicios Web se han convertido en estándares ISO (International Organization for the Standardization).

Adicionalmente, el rastreo se limita a las especificaciones OGC de servicios Web relacionadas con la Directiva Europea INSPIRE (European Parliament and Council of European Union, 2007). La Directiva INSPIRE es el marco más grande y más reciente de tamaño continental que establece normas compartidas entre los países europeos para el acceso a la información geográfica digital. Los servicios Web de OGC han sido señalados como una de las posibles implementaciones de los servicios de INSPIRE (INSPIRE DT NS, 2008, 2009). Esta restricción garantiza que las contribuciones de esta tesis pueden beneficiar a investigadores y desarrolladores interesados en la aplicación de la Directiva INSPIRE (2010-2019).

La conceptualización de la arquitectura de servicios Web de OGC debe ser reutilizable, y debe evitar una fuerte dependencia de la terminología de OGC. La arquitectura de servicios Web de OGC está basada en la familia de normas RM-ODP (Reference Model of Open Distributed Processing, ISO / IEC 10746-2 Foundations (2009), ISO / IEC 10746-3 Arquitectura (2009), ISO 15414 Enterprise language (2006), véase Percivall, 2002). RM-ODP es un modelo de referencia que proporciona un marco para la normalización de los sistemas abiertos de procesamiento distribuido. La terminología, los conceptos y las relaciones utilizadas para la conceptualización de esta tesis se derivan principalmente de la RM-ODP. Estos conceptos se aplican a conceptualizar el conocimiento del dominio que se encuentran en la especificación abstracta de OGC para los servicios Web (Percivall, 2002) y en la recopilación de normas específicas de OGC para interfaces de servicios Web (Whiteside, 2007; de la Beaujardiere, 2006; Vretanos, 2010; Baumann, 2010; Schut, 2007; Voges and Senkler, 2007).

Cuando sea necesario, el conocimiento se formalizará mediante Lógica Descriptiva, una familia de formalismos de representación del conocimiento (Baader et al., 2003). Lógica Descriptiva está equipada con una semántica formal, basada en la lógica, que proporciona los procedimientos de inferencia para derivar conocimiento implícito del conocimiento que está explícitamente representada. El uso de Lógica Descriptiva se ve obstaculizado por la complejidad computacional de algunos de sus algoritmos de inferencia. Sin embargo, se reconoce que sistemas relativamente expresivos de tamaño real con problemas de razonamiento exponenciales en tiempo pueden ser procesados en un tiempo razonable (Horrocks, 1998). En esta tesis se formalizará el conocimiento tomando como límite la Lógica Descriptiva $SROIQ(D)$. Esta Lógica Descriptiva tiene propiedades interesantes de cómputo (see Horrocks et al., 2006). Razonadores populares, como Pellet (Parsia and Sirin, 2004) y HermiT (Motik et al., 2009b), soportan $SROIQ(D)$. Además, el lenguaje de ontologías OWL 2 (Web Ontology Language 2, OWL WG, 2009) es compatible con algunas restricciones con $SROIQ(D)$.

Solo dos tipos de accesos a contenidos Geoespaciales se consideran en esta tesis: el acceso a los metadatos geoespaciales almacenados en OGC CSW (CSW, Nebert et al., 2007) y el acceso a los datos geoespaciales almacenados en OGC WFS (WFS, Vretanos, 2010). Por otra parte, el acceso a los metadatos geoespaciales almacenados en las instancias de servicio CSW se limita a los metadatos de los registros cuyo esquema de metadatos tiene una correspondencia conocida por el modelo abstracto de Dublin Core Metadata Initiative (Powell et al., 2007). El acceso a los datos geoespaciales se limita a los casos de servicios WFS que devuelvan fenómenos, es decir, representaciones de lugares, que se puedan corresponder con una extensión del meta modelo GKB, que fue desarrollado en el proyecto GREASE (Chaves et al., 2005).

Método. La aproximación metodología de esta tesis tiene dos aspectos: uno relacionado con la ingeniería de software y otro relacionados con la ingeniería del conocimiento. La metodología en la parte relacionada con la ingeniería de software es un desarrollo clásico incremental de la solución. La metodología en la parte de relacionada con la ingeniería del conocimiento se basa en Methontology (Fernández-López et al., 1997).

Importancia social, experiencia previa, trabajo futuro. El desarrollo de una Web Geoespacial pública, abierta e interoperable es consecuencia de las iniciativas para el desarrollo de Infraestructuras de Datos Espaciales (IDEs). Por ejemplo, la iniciativa europea INSPIRE (materializado en 2007 como Directiva Europea) tiene por objeto crear una IDE europea. Una de las líneas de investigación del grupo de investigación IAAA se centra en aspectos relacionados con las las IDEs como la descripción de los datos geoespaciales y los servicios, el descubrimiento de estos recursos a través de catálogos estándar, y los aspectos conceptuales y arquitectónicos relacionados con datos geoespaciales y servicios. Algunos resultados de la investigación de la línea de investigación de IDEs, donde ha participado el autor son la exploración de nuevas alternativas para el descubrimiento de servicios geoespaciales (Lopez-Pellicer et al., 2010a), y el estudio de nuevas formas de dar acceso a los datos geoespaciales y los metadatos (Lopez-Pellicer et al., 2010b,d). Este trabajo ha consistido en la caracterización de modelos de contenido (Lopez-Pellicer et al., 2007, 2008), el desarrollo de conjuntos de datos geoespaciales (Lopez-Pellicer et al., 2006; Nogueras-Iso et al., 2007), y la colaboración en la formalización y la producción de grandes geo-ontologías (Lopez-Pellicer et al., 2009; Cardoso et al., 2009; Lopez-Pellicer et al., 2010c). Además, el autor ha participado en la elaboración de normas para el contenido geoespacial (Modelo de Nomenclátor de España, Rodríguez-Pascual et al., 2006), y servicios geoespaciales (OGC Table Joining Service, Schut et al., 2010). Esta tesis es resultado de la investigación citada. Las líneas de trabajo futuro están orientadas a mejorar las contribuciones relacionadas con en el descubrimiento de servicios Web geoespaciales, la formalización de los modelos y el acceso a los datos geoespaciales.

Executive summary

Background. The effective access to geospatial information acquires a critical importance in a knowledge-based economy. It is essential to many activities to have access to geospatial resources through the Web. From the response to a disaster, to the decision to develop a new business, the access to up-to-date geospatial information offered by the Geospatial Web could be the difference between the success and the failure. The Geospatial Web is the collection of Web services, geospatial data and metadata that supports the use of geospatial data in a range of domain applications.

This thesis addresses three problems about the Geospatial. First, the standard services of the Geospatial Web are part of the invisible Web. The literature uses the term invisible for identifying the part of the Web ignored by search engines (Sherman and Price, 2001). Ordinary users can be aware of invisible Web content only by chance. Next, this thesis classifies the content of the Geospatial Web accessible through standard services as part of the deep Web (Bergman, 2001). The term deep highlights that an invisible content is worthwhile to be surfaced, in other words, indexed by a search engine. The Geospatial Web contents are deep by their own nature. That is, the Geospatial Web contents are hidden behind the Geospatial Web services. Finally, the standard services of the Geospatial Web and its contents are socially disconnected from the rest of the Web. Public Geospatial Web services not are conceived to serve functionality to service consumers and, at the same time, to be social tool that enable the communication among agents related with the services. Today, linking Web services with the Linking Open Data[2] initiative (see Bizer et al., 2009), one of the materializations of the Semantic Web (Berners-Lee et al., 2001), seems to promise such a possibility.

Objective. The problems derived of the invisibility of Geospatial Web services might be mitigated if it is possible to perform a focused and systematic crawl of the Web for Geospatial Web services combined with the publication as Linked Data of the descriptions and the contents of the discovered Geospatial Web services.

This thesis analyses the feasibility of a solution based in the development of a crawler focused on Geospatial Web services, and the Web access to machine processable descriptions of the discovered

[2] `http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

Geospatial Web services. In addition, the analysed solution should give access to remote geospatial content as machine processable data.

This approach to the solution requires the development of two domain ontologies. The first ontology describes the abstract architecture of Geospatial Web services. The second ontology describes Geospatial content.

Scope. The focused and systematic crawl is restricted to Web services compliant with the Open Geospatial Consortium (OGC) specifications. The OGC leads the development of open and standardized Web service interface specifications for accessing geospatial information since 1994. More than 400 companies, government agencies and universities are members of OGC, and they participate in consensual processes to develop publicly available Web service interface standards for the access to geospatial data. Many OGC Web service interface specifications have become standards of the International Organization for the Standardization.

Moreover, the crawl is restricted to OGC Web service specifications related with the European INSPIRE Directive (European Parliament and Council of European Union, 2007). The INSPIRE Directive is the largest and most recent regional framework in the world that establishes shared standards between European countries for the access to digital geographic information. The OGC Web services have been indicated as one of the possible implementations of INSPIRE compliant services (INSPIRE DT NS, 2008, 2009). This restriction guarantees that the contributions of this thesis can benefit academic and industrial audiences interested in the implementation of the INSPIRE directive (2010-2019).

The conceptualization of the OGC Web services architecture should be reusable, and should avoid being tightly dependent on OGC terminology. The OGC Web service architecture is loosely based in the *Reference Model of Open Distributed Processing* (RM-ODP, ISO/IEC 10746-2 Foundations (2009), ISO/IEC 10746-3 Architecture (2009), ISO 15414 Enterprise language (2006)) family of standards (see Percivall, 2002). RM-ODP is a reference model that provides a framework for the standardization of open distributed processing systems. The terminology, concepts and relations used for the conceptualization in this thesis are mainly derived from the RM-ODP. These concepts are applied to conceptualize domain knowledge found in the OGC Abstract Specification for Web services (Percivall, 2002) and the collection of OGC Standards for specific Web service interfaces (Whiteside, 2007; de la Beaujardiere, 2006; Vretanos, 2010; Baumann, 2010; Schut, 2007; Voges and Senkler, 2007).

When required, knowledge will be formalized using Description Logics, a family of knowledge representation formalisms (Baader et al., 2003). Description Logic is equipped with formal, logic-based semantics that provide inference procedures for deriving implicitly knowledge from the knowledge that is explicitly represented. The use of Description Logics is hindered by the intractability of some of their inference algorithms. However, it is acknowledged that relatively expressive systems of realistic size with exponential-time reasoning problems can be processed in reasonable time (Horrocks, 1998). This thesis will develop systems less expressive than the $SROIQ(D)$ Description Logic. The $SROIQ(D)$ Description Logic has useful computational properties (see Horrocks et al., 2006). Popular reasoners, such as Pellet (Parsia and Sirin, 2004) and HermiT (Motik et al., 2009b), support $SROIQ(D)$. In addition, the OWL 2 Web Ontology Language (OWL 2, OWL WG, 2009) is compatible with some restrictions with $SROIQ(D)$.

Two kinds of accesses to Geospatial content are considered: the access to geospatial metadata stored in OGC Catalogue Service for the Web (CSW, Nebert et al., 2007), and the access to geospatial data stored in OGC Web Feature Services (WFS, Vretanos, 2010). Moreover, the access to geospatial metadata stored in CSW service instances is restricted to metadata records whose metadata schema has a well-known mapping to the abstract model of the Dublin Core Metadata Initiative (Powell et al., 2007). The access to geospatial data is restricted to those WFS service instances that return *geographical features*, that is, representations of places, that can be mapped to an extension of the GKB metamodel, which was developed in the GREASE project (Chaves et al., 2005).

Method. The methodological approach of this thesis has two aspects: one related with software engineering and other related with knowledge engineering. The software engineering methodology is a classic incremental development of the solution. The knowledge engineering methodology is based in the steps proposed by the Methontology framework (Fernández-López et al., 1997).

Social importance, previous experience, future work. The development of a public, open and interoperable Geospatial Web is consequence of the initiatives for the development of Spatial Data Infrastructures (SDIs). For example, the European INSPIRE initiative (materialized in 2007 as European Directive) aims to create a European SDI. One of the research lines of the IAAA research group focuses on SDI aspects related with the description of geospatial data and services, the discovery of these resources through standard catalogues, and the conceptual and architectural aspects related to geospatial data and services. Some research results of the SDI research line where the author has participated are the exploration of new alternatives for the discovery of geospatial services (Lopez-Pellicer et al., 2010a), and the study of new ways to give access to geospatial data and metadata (Lopez-Pellicer et al., 2010b,d). This work has involved the characterization of content models (Lopez-Pellicer et al., 2007, 2008), the development of geospatial datasets (Lopez-Pellicer et al., 2006; Nogueras-Iso et al., 2007), and the collaboration in the formalization and production of large geo-ontologies (Lopez-Pellicer et al., 2009, 2010c). Additionally, the author has participated in the development of standards for geospatial content (Modelo de Nomenclátor de España, Rodríguez-Pascual et al., 2006), and geospatial services (OGC Table Joining Service, Schut et al., 2010). This thesis is included in the aforementioned research line and it is the result of the cited research. Future work will improve contributions in the discovery of geospatial Web services, the formalization of models and the access to geospatial data.

Contents

List of Tables

List of Figures

Nomenclature

ADL	Alexandria Digital Library
APA	Agência Portuguesa do Ambiente
API	Application Programming Interface
AWWW	Architecture of the World Wide Web
CAT	Catalogue Service
CRS	Coordinate Reference System
CTS	Coordinate Transformation Service
DCAM	DCMI abstact model
DCMI	Dublin Core Metadata Initiative
DCP	Distributed Computing Platform
ETRS	European Terrestrial Reference System
FCCN	Fundação para a Computação Científica Nacional
GIS	Geographic Information System
GKB	Geographic Knowledge Base
GML	Geography Markup Language
GOS	Geographic Ontology Serializer
HTTP	Hypertext Transfer Protocol
IDEE	Infraestructura de Datos Espaciales de España
IGeoE	Instituto Geográfico do Exército

IGP	Instituto Geográfico Português
INSPIRE	Infrastructure for Spatial Information in Europe
ISO	International Organization for Standardization
ISO GMD	ISO Geographic Information – Metadata
ISO SRV	ISO Geographic Information – Services
ISO/TC 211	ISO Technical Committee 211 Geographic information/Geomatics
KML	Keyhole Markup Language
KVP	Keyword Value Pair encoding
MEN	Modelo Español de Nomeclátor
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OntoOWS	Ontology for OGC Web Services
ORM	Object-relational mapping
OWL	Web Ontology Language
OWS	OGC Web service
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
RIF	Rule Interchange Format
RPC	Remote Procedure Call
SDI	Spatial Data Infrastructure
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
TAG	Technical Architecture Group

UDDI	Universal Description Discovery and Integration
UNGEGN	United Nations Group of Experts on Geographical Names
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
voID	Vocabulary of Interlinked Datasets
W3C	World Wide Web Consortium
WARC	Web ARChive
WGS84	World Geodetic System 1984
WKT	Well-known text mark-up language
WSDL	Web Services Description Language

Chapter 1

Context and research issues

The Web has a geospatial dimension. Web sites *consume* geospatial information to offer points of reference on the Earth to their users. Web sites, such as Google Maps, consume geospatial data, such as the roadways of the United States, the forests in Europe, or the hydrography in South Asia, *produced* by authoritative data producers, such as the US Department of Transportation, the European Space Agency and the Environmental Information Centre of India. The skills of consumers and providers for managing geospatial information using the Web, for structuring geospatial Web-based workflows, and for communicating each other the meaning of geospatial data shape the Geospatial Web. For the purposes of this thesis, the definition of the Geospatial Web is broad:

> **The Geospatial Web is the collection of Web services, geospatial data and metadata that supports the use of geospatial data in a range of domain applications.**

Hereafter, this thesis uses the term *Geospatial Web services* to identify Web information services concerned with geospatial data and metadata. Geospatial Web services, geospatial data, and metadata about geospatial resources are collectively identified as *Geospatial Web resources.*

Geographic Information Systems (GIS) are the origin of the Geospatial Web. A GIS is an information system that captures, stores, analyses, manages, and presents data with geographic information. A defining characteristic of GIS systems is the separation and special treatment of geospatial information from other available data. The concept of GIS was introduced by in the late 1960s. One of the first applications was the work of Tobler (1970) that coined the *First Law of (digital) Geography* from its experience with an early GIS system:

> *Everything is related to everything else, but near things are more related than distant things.* Tobler (1970)

In other words, GIS systems help to make visible to analysts relations between datasets which are spatially related that otherwise may be kept hidden. Between the late 1970s and the early 1980s, public domain GIS systems (e.g. GRASS GIS) and commercial GIS systems (e.g. ESRI Arc-Info) were available. Coppock and Rhind (1991) described the 1980s as the end of the beginning of GIS and predicted that:

> *It is a reasonable expectation that routine (and often boring, if valuable) use of GIS will be nearly ubiquitous over the next 20 years.* Coppock and Rhind (1991)

At the same time, the Web was in development at the CERN[1]. Soon, the GIS community realized that the Web was a revolutionary opportunity to gain easy access to remote geospatial data, tools and processes. Longley et al. (2005), which describes the evolution of GIS between the late 1980s and the beginning of the 21st century, affirms:

> *In the context of the World Wide Web, geospatial information becomes truly more useful to more people and its potential for enabling progress and enlightenment in many domains of human activity can be more fully realized.* Longley et al. (2005)

Since 1994, the Open Geospatial Consortium[2] (OGC) leads the development of open and standardized Web service interface specifications for accessing geospatial information (see Table 1.1). More than 400 companies, government agencies and universities are members[3] of OGC, and they participate in consensual processes to develop publicly available Web service interface standards for the access to geospatial data. Many OGC Web service interface specifications have become standards of the International Organization for the Standardization (ISO). The ISO Technical Committee 211 Geographic information/Geomatics[4] (ISO/TC 211), which works on standardization in the field of digital geographic information, has a working arrangement with OGC that often results in virtually identical OGC and ISO standards (see Kresse and Fadaie, 2004). OGC and ISO/TC 211 share the objective of providing a framework for the development of domain applications using geospatial resources.

The aim of this thesis is to contribute to the effective use of Geospatial Web resources, in particular, those resources related with the OGC Web services. There are aspects related with the discovery of Geospatial Web services and the access to Geospatial content behind these services that can be improved. Concrete technical issues, such as the *invisibility* of Geospatial Web services, the *hiddenness* of Geospatial content and the *disconnection* of the information about them, are the motivations of the contributions of this thesis.

[1] A copy of the original first Web page on the CERN website is still available as a historical document on the World Wide Web Consortium site: `http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html`

[2] `http://www.opengeospatial.org/`

[3] 415 active members as of January 20, 2010: `http://www.opengeospatial.org/ogc/members`

[4] `http://www.isotc211.org/`

Table 1.1: Relevant OGC Web Service interface specifications.

Name	Acr.	Since	Current version	Objective
Web Map Service	WMS	2000	1.3.0 (de la Beaujardiere, 2006)	Portrayal
Web Feature Service	WFS	2002	2.0.0 (Vretanos, 2010)	Download features
Web Coverage Service	WCS	2003	2.0.0 (Baumann, 2010)	Download coverages
Catalogue Service for the Web	CSW	2004	2.0.2 (Nebert et al., 2007)	Discovery
Web Processing Service	WPS	2005	1.0.0 (Schut, 2007)	Remote invocation
Sensor Observation Service	SOS	2006	1.0.0 (Na and Priest, 2007)	Download observations

1.1 Motivation

The effective access to geospatial information acquires a critical importance in a knowledge-based economy. It is essential to many activities to have access to geospatial resources through the Web. From the response to a disaster, to the decision to develop a new business, the access to up-to-date geospatial information exploiting the Geospatial Web at maximum could be the difference between the success and the failure. The term *Geospatial Web* encompasses map-based sites, such as Google Maps and Yahoo Local Maps; geobrowsers, such as NASA World Wind and Google Earth. It also includes Web communities, such as GeoNames and OpenStreetMap. Map-bases sites and Web communities may use Web services with standard OGC interfaces, such as the NOAA Web Map Service that provides historical data of tsunamis and the Web Feature Service that gives access to the urban planning data of the town of Roses (Spain), that give access to geospatial data. Public datasets available for download with geospatial content, such as the TIGER database of US addresses and the PNOA collection of aerial photos of Spain, are also part of the Geospatial Web. These sites, communities, services and data located on the Web (see Table 1.2) reveal the existence of an active community of providers and consumers of data and products related with geographic locations.

Web services, such as the OGC Web service interface standards, are the enabling technology of the Geospatial Web. They have revolutionized the production and consumption of geospatial information. These technologies not only enable access to online geospatial information, but also enable the development of tools that bring together people of similar interests, browsing behaviour, or geographic location.

The Geospatial Web offers to the users different methods to get access to geospatial data. Three major ways give access to geospatial data: *online GIS*, *Spatial Data Infrastructures*, and *spatial browsing systems.*

Online GIS systems are one of the earliest components of the Geospatial Web. Online GIS systems should not be confused with desktop GIS systems. A desktop GIS system is installed and runs on a personal computer, and is accessed and controlled from the same computer (see Steiniger

Table 1.2: Examples of Geospatial Web resources.

Type	**Site**	**URL**
Map-based sites	Google Maps	`http://maps.google.com/`
	Yahoo Local Maps	`http://maps.yahoo.com/`
Geobrowsers	NASA World Wind	`http://worldwind.arc.nasa.gov/`
	Google Earth	`http://www.google.com/`
Communities	GeoNames	`http://www.geonames.org/`
	OpenStreetMap	`http://www.openstreetmap.org/`
Standard Web services	Historic Tsunamis Web Map Service (NOAA, US)	`http://map.ngdc.noaa.gov/wmsconnector/com.esri.wms.Esrimap/hazards?request=getcapabilities&Service=wms&version=1.1.1`
	Urban planning Web Feature Service (Roses, Spain)	`http://www.roses.cat/cgi-bin/mapserv.exe?map=../Data/mapserver/planejament.map&REQUEST=GetCapabilities&SERVICE=WFS`
Datasets	TIGER Database	`http://www.census.gov/geo/www/tiger/`
	PNOA Collection	Available at `http://centrodedescargas.cnig.es/CentroDescargas/`

and Bocher, 2009). The GIS community was an early adopter of the Web, and soon GIS software vendors, such as ESRI and Autodesk, started to offer Web based extensions to their desktop GIS. The enabling technologies of Web based extensions were proprietary protocols (e.g. ESRI ArcXML) that were used by Web clients to make requests to GIS systems. By the end of the 1990s, the core element of many GIS product lines were application servers (e.g. ESRI ArcGIS Server, Autodesk MapGuide) that supply mapping and GIS capabilities to other products (e.g. ESRI ArcGIS Desktop, AutoCAD Map).

GIS systems may not be necessarily interoperable each other because they are often developed for different purposes, jurisdictions or clients. The lack of interoperability was acknowledged as a costly limitation to the collection, sharing and use of digital geographic information as soon as GIS systems were widely adopted by public administrations in the late 1980s (Darman, 1990). The promotion of the geospatial interoperability is the cause of inception in the 1990s of public-led initiatives oriented to the development of the basic technological and organizational structures, policies and standards that should enable an efficient discovery, transfer and use of geospatial data using the Web (FGDC, 2004). These initiatives were named *Spatial Data Infrastructures* (SDI, see Nebert, 2004).

However, when users surf the Web looking for content with location, such as an address in London, or images about New York, they use neither online GIS applications nor SDI based applications.

Table 1.3: Examples of SDIs that endorse the use of OGC Web service standards.

Spatial Data Infrastructure	**Scope**	**Reference**
Committee on Earth Observation Satellite (CEOS)	Global	WG ISS (2008)
National System for Geospatial Intelligence (NSGI)	USA	GWG (2009)
Federal Enterprise Architecture (FEA)	USA	AIC and FGDC (2006)
Canadian Geospatial Data Infrastructure (CGDI)	Canada	GeoConnections (2007)
Infrastructure for Spatial Information in Europe (INSPIRE)	Europe	INSPIRE DT NS[a](2008)
Infraestructura de Datos Espaciales de España (IDEE)	Spain	IDEE Service Directory[b]

[a] The technical guidance of this architecture endorses the use of OGC standards. Further information can be found in `http://inspire.jrc.ec.europa.eu/`.

[b] `http://www.idee.es/CatalogoServicios/CatServ/directorio_servicios.html`

They use instead Google Map or OpenStreetMap to visualize in a map online content with location, or use Web applications that embed the cited applications. These Web applications exhibit *spatial browsing* functionality. Larson coined the term spatial browsing in 1996. Spatial browsing combines spatial questions with interactive display of digital maps in such a way that allow users to explore the geographical dimension of one or several information systems in new and unexpected ways. Spatial browsing may use standard Geospatial Web services, such as OGC Web Map Service and Web Feature Services, to access to the geospatial data stored in remote information systems. Having said that, the most popular spatial browsing systems found in the Web (e.g. Google Maps, OpenStreetMap) use non-standardized proprietary specifications, which is mainly due to how they were conceived. Web developers developed these systems for a massive use and reuse in the Web, taking into account constraints and dynamics that only happen in the context of the Web. The reuse has allowed the development of new spatial browsing systems on top of existing spatial browsing systems. These applications are know are geo-mashup. A mashup is a Web application that integrates information from more than one source (Jhingran, 2006). A geo-mashup[5] is one on which the map metaphor plays a key role in the integration of the information (Elson et al., 2007). For example, the street gazetteer of the Zaragoza City Council[6] (Spain) is a geo-mashup that uses Google Maps as base spatial browsing system. It overlays Google Maps with cadastre maps from a Web Map Service managed by the Directorate General for Cadastre of Spain, and points of interest from a Web Feature Service managed by the Zaragoza City Council.

Online GIS and SDI rest on open and proprietary standardized Web service interfaces. The

[5] At the beginning, nearly all the mashups were geo-mashups. In December 2008 about 54% of the mashups exhibited spatial browsing functionality. In January 2010, the percentage shows a slow decline to 45% of mashups (`http://www.programmableweb.com/apis`).

[6] `http://idezar.zaragoza.es/callejero/`

geospatial community endorses the use of the open standards specified by OGC for the implementation of Geospatial Web services in SDIs (Nebert, 2004). Table 1.3 shows the most relevant national and global SDI that endorse the use of OGC Web service standards. SDI communities work in a *hierarchical federal-like scheme* (Rajabifard and Williamson, 2001; Dasgupta, 2009; Béjar et al., 2009). That is, each federation member propagates downward the federation level consensus on the use of OGC standards for the communication between members. For example, the community built around the European INSPIRE Directive (European Parliament and Council of European Union, 2007) endorses the use of OGC standards between members of the European Union. The national SDI of Spain (IDEE), member of the INSPIRE community, assumes the use of the OGC standards and recommends their use in Spain. Finally, SDIs members of the IDEE, whose geographical scope is often an administrative division of Spain (e.g. IDEC - Catalonia, IDEZAR - Zaragoza City), follows the recommendations and adapt their online GIS systems to the use of OGC standards.

There are several ways to classify the different OGC Web service interface specifications (e.g. ISO 19119:2005, OGC Web service Common Specification (Whiteside, 2007)). The European INSPIRE Directive provides a general and convenient classification:

- *Discovery services* are the services that help users for the discovery of spatial data and services. Examples of discovery services are those who broke queries to geospatial catalogues, directories and clearinghouses. OGC has an abstract specification for discovery services that defines bindings to different distributed computing platforms. The *Catalogue Service for the Web* (CSW, Nebert et al., 2007) is the name of the binding for a distributed computing platform formed by hosts that execute remote procedure calls using the HTTP protocol.

- *Download services* provide access to representations of spatial data encoded in a processable file format. Examples of download services range from those that provide access to a part of a dataset to complex gazetteer services that accept fuzzy queries. The *Web Feature Service* (WFS, Vretanos, 2010) for discrete geographic *features*[7], the *Web Coverage Service* (WCS, Baumann, 2010) for data coverage sets of the Earth, and the *Sensor Observation Service* (SOS, Na and Priest, 2007) for real-time sensor data are some of the OGC Web service standard specifications related with download services.

- *Portrayal services* are services that render images derived from one or several geospatial datasets. Examples of portrayal services are mapping services and services that manage the symbols used for the visualization in mapping services. The specification *Web Map Service*

[7] In the domain of geographic information, the term *feature* is used instead of a term more familiar to software engineering, such as *object,* when it is used to denote the elements of a dataset. The term *feature* has been adopted by OGC (see Kottman and Reed, 2009) and ISO/TC 211 (see Kresse and Fadaie, 2004) as the term that designate the elements of a dataset and *feature type* as the term that designate their class. The term *feature* also refers to a distinguishing characteristic of a software item (e.g., performance, portability, or functionality) (IEEE 829). In sections related to software description, the term *geospatial feature* will be used instead of *feature* to refer to abstractions of the real world in order to avoid ambiguity.

(WMS, de la Beaujardiere, 2006) is the most successful specification of OGC. Four out of five products that implements OGC Web service interface specifications implement revisions of the WMS specification (OGC, 2010).

- *Invocation service*s are remote invocation services whose inputs or outputs are related with spatial data or services. OGC promotes the *Web Processing Service* (WPS, Schut, 2007) as the specification for geospatial invocation services.

The adoption of the OGC Web services implies not only a set of Web service interfaces but also the adoption of an architectural style. The architectural style for OGC Web Services (Lieberman, 2003; Whiteside, 2007) has its roots in the Service Oriented Architecture (SOA, see Erl, 2005). SOA is an architectural style that packages functionality as a suite of interoperable services. A SOA implementation can provide a way for consumers of services, such as web-based applications, to be aware of available SOA-based services.

The discovery of OGC SOA services is based on the *publish-find-bind* pattern for Web services (Gottschalk et al., 2002; Papazoglou, 2003). This pattern includes the next interactions. First, service providers describe the features of each Web service in standardized documents named *service metadata*, also known as capabilities XML when encoded in XML. Then, these descriptions are published in registries, or are made accessible through hyperlinks that encodes a HTTP *GetCapabilities* request to the service (see Whiteside, 2007). The mandatory *GetCapabilities* operation allows an OGC Web service to self-describe by returning a service metadata document. Alternatively, metadata records encoded in an appropriate metadata schema, such as the extension of the ISO/TC 211 Geographic Metadata schema for geospatial services (ISO 19119:2005), can be stored in a transactional discovery service. Both kind of metadata should describe in sufficient detail not only the technical characteristics of the services, but also its functionality and the data that is accessed by them. Next, service consumers search for service descriptions in the Web (e.g. capabilities XML) and within discovery services (e.g. ISO 19119 metadata record). Finally, when a user finds a relevant service description, it can extract all the binding information and connect to the service.

OGC envisioned that a SOA based architecture will lead to the creation of an automated world wide public interoperable e-market based in discovery services where geospatial providers and geospatial consumers trade data and services (McKee and Kottman, 1999). However, SOA based architectures have failed to fulfil this vision (see Al-Masri and Mahmoud, 2008b; Batcheller, 2008; Pedrinaci et al., 2010). *Geoportals*, that is, web sites that acts as single point of access that links to logically related Geospatial Web resources (Rose, 2004; Maguire and Longley, 2005; Béjar et al., 2009), may be though to some extent as a non-automated replacement to the expected e-market.

The lack of an e-market of services and the non-automated nature of the geoportals make difficult an estimation of the amount of OGC Web services. A partial account can be obtained by reviewing the sites that maintains lists of OGC Web services. *Microimages*[8] maintains a collection of more

[8] `http://www.microimages.com/wmscatalog/`

than 2,500 OGC Web services and more than 20,000 ArcIMS services (Microimages, 2008). According to Ager et al. (2010), a Web crawler feeds this large collection. *WMS-Finder*[9] (Alta4, 2010) and *Geopole*[10] (Kalberer, 2010) maintain lists with 3,000–4,500 WMSs whose origin is unknown. However, according to Masser (2005), only in the US there are more than 100,000 organisations engaged in SDI related GIS activities. Hence, it is reasonable to expect more Geospatial Web services listed in directories. Perhaps, service directories contain few services because Geospatial Web services, in particular OGC Web services, are *invisible* for Web users, including the small elite of spatially aware professionals.

1.2 Problem statement

There are three hypotheses about the Geospatial Web that this thesis addresses.

1. **The standard services of the Geospatial Web are part of the invisible Web.**
2. **The contents of the Geospatial Web accessible through standard services are part of the deep Web.**
3. **The standard services of the Geospatial Web and its contents are socially disconnected from the rest of the Web.**

Empirical evidences related with the above motivations support these hypotheses.

The standard services of the Geospatial Web are part of the invisible Web. The literature uses the term *invisible* for identifying the part of the Web ignored by search engines (Sherman and Price, 2001). Ordinary users can be aware of the invisible only by chance. Indexing invisible resources in the Web is a technical, economical and legal challenge. The decision should take into account the cost of the required resources and the liabilities related to the reuse of published content. Even if a search engine company accepts the economical and legal risks, its developers must face several technical challenges: disconnected resources, not indexable formats, unintelligible content, web forms, real-time data, stateful content, scripted resources, and access restrictions.

The Geospatial Web merges the geographic information with the Web for the good and for the worse. A common-sense implication is that the merged content should share the visibility problems of the Web. That is, there is an *invisible Geospatial Web.* Standard Geospatial Web services are part of the invisible Geospatial Web because search engines are experts in deal with HTML documents, some formats with embedded metadata and plain text content (see the still valid work of Sherman and Price, 2001). That means that Geospatial Web services are a kind of Web resource that fly under the radar of search engines.

[9] `http://www.wms-finder.de/`

[10] `http://www.geopole.org/`

The contents of the Geospatial Web accessible through standard services are part of the deep Web. The Web is one of the largest repositories of knowledge and, at the same time, the entry point to databases full of online valuable information. The literature often associate the term *hidden* to the online content that it is accessed only through search forms designed primarily for human consumption or returned by services in response to submitted queries (e.g. "*hidden behind search forms*" (Ntoulas et al., 2005), "*the content hidden behind HTML forms*" (Madhavan et al., 2008)). The term *deep* highlights that an invisible content is worthwhile to be *surfaced*, in other words, indexed by a search engine. The Geospatial Web contents are *deep* by their own nature. That is, the Geospatial Web contents are *hidden* behind the Geospatial Web services.

There is no reliable way to estimate the size of the *deep Geospatial Web*. Few figures can be drawn from studies about the deep Web. The quantitative survey of Bergman (2001) affirmed that six Geospatial Web sites out of 60 largest deep Web sites at that time contained an 80% of the deep Web content. In the qualitative survey of (He et al., 2007), the sites of the Geospatial Web identified by Bergman can be considered under the category *scienc*e, which includes less than 3% of the databases identified.

The standard services of the Geospatial Web and its contents are socially disconnected from the rest of the Web. Although technical limitations could be the cause of the apparently low implantation of Geospatial Web services, the motivations behind their relative failure to deliver of services on the Web could be more social that technological. Public Geospatial Web services not are conceived to serve functionality to service consumers *and at the same time* to enable the communication among service consumers. A recent work about Web services highlights that the driven force of the success of some kinds of Web services resides on the communication links between people, which is subject to the network effect (Pedrinaci and Domingue, 2010). The *network effect* (Hendler and Golbeck, 2008) is the intuition that when the number of people in a network in the Web grows, the connectivity increases. More connectivity in the Web implies that more people could link to each other's content. Rich linked content could attract more people to the network, feeding back into the network. The network effect on the Web has paved the way to the Social Web (e.g. Facebook), new business models (e.g. Wikipedia), and movements for the publication data on the Web (e.g. Linking Open Data initiative). In this sense, enabling the addition of links from or to the Geospatial Web services and its contents should increase the perception of connectivity, and could trigger a network effect that cross the boundaries of the geospatial community.

The network effect is also the driving force of the Semantic Web (Berners-Lee et al., 2001). The Semantic Web is an initiative led by the World Wide Web Consortium (W3C), with participation from a large number of researchers and industrial partners, whose goal is the development of a common framework that should allow data to be meaningfully shared and reused across application, enterprise, and community boundaries using the Web as platform. The Web was designed as a Web of documents. Humans can use the Web of documents to carry out complex tasks, such as the discovery of a web mapping service with information about an emergency disaster[11], or finding the Spanish word for *geographic feature*[12]. User agents cannot perform these activities alone because the Web content are represented as documents designed to be understood by humans (e.g. Web pages, images, movies, music), not machines. The early vision of the Semantic Web is a Web where the information can be interpreted by user agents, that is, machine processable with formal semantics, so user agents can performs tedious tasks on user's behalf returning the same results that a user directed work. This idea remained largely unrealized (see Shadbolt et al., 2006) until the skyrocketed development of the Linked Data vision of Berners-Lee (2006) and the Linking Open Data[13] initiative (see Bizer et al., 2009). The data model of the Semantic Web is the Resource Description Framework (RDF, Carroll and Klyne, 2004) that enables the representation of resources as statements in the form of subject-predicate-object expressions. The nodes of the graph are resources, named or blank, and values, also known as literals. Each named node has an associated Uniform Resource Identifier (URI, Berners-Lee et al., 2005) that uniquely identifies the node.

The Geospatial Web is perceived from part of the Semantic Web community as a provider of datasets rich in geographic descriptions that need to be extracted from their silos. This is, for instance, the approach of the publication as RDF of OpenStreetMap (Auer et al., 2009). On the other side, the Semantic Web is perceived from the Geospatial Web community as a provider of formal machinery, such as the Web Ontology Language (OWL, McGuinness and van Harmelen, 2004). The application of Semantic Web technologies includes from enabling meaningful geospatial information retrieval using geospatial ontologies (Egenhofer, 2002) to the development of profiles of Geospatial Web services with RDF and OWL support (Janowicz et al., 2010). Large geographical information providers are investigating how Linked Data and other Semantic Web technologies can assist in the diffusion of geographic data. Goodwin et al. (2009) describes how Ordnance Survey is developing datasets in RDF and publishing them using the Linked Data principles, Auer et al. (2009) explains in detail how the LinkedGeoData project provides OpenStreetMap data as Linked Data, and Vatant and Wick (2007) documents how GeoNames provides access to its place names database through a Linked Data interface .

1.3 Research questions

The central argument of this thesis is the following. The research in search engine technologies shows that the main restrictions to the visibility of Web resources are technical, economical and legal. A focused crawl of geospatial Web services is technically feasible only if specialists in the geospatial domain are part of the project. The costs of such crawl (e.g. storage) can be restricted if the crawl

[11] The terms *wms haiti disaster* and *wms tsunami indonesia* returns as first results in any search engine pages with maps services about these emergencies.

[12] The word in Spanish is *fenómeno* in the sense fact or situation that is observed to exist or happen.

[13] `http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

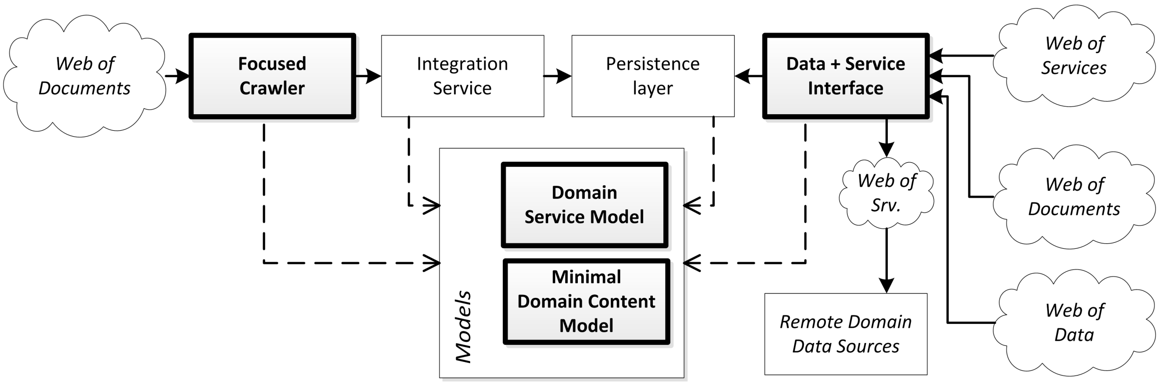


Figure 1.1: The conceptual solution.

only locates geospatial services and does not surface the geospatial content. There are few legal problems due to the interest of SDIs in disseminate geospatial information. In relation with the disconnected content, Linked Data seems to be the most promising technology to add links from or to the Geospatial Web. It is possible to publish RDF documents as Linked Data that among the RDF links to Semantic Web resources includes links to URIs that resolve to requests to Geospatial Web services, opening new possibilities of interaction. These RDF documents should convey the description of the Geospatial Web services and the Geospatial content in a uniform way. This can be done through the specification of ontologies.

It is essential to this thesis the demonstration of the feasibility of a solution based in the above argument. That is:

> ***The problems derived from the invisibility of Geospatial Web services might be mitigated if it is possible to perform a focused and systematic crawl of the Web for Geospatial Web services combined with the publication as Linked Data of the descriptions and the contents of the discovered Geospatial Web services.***

The above argument is a normative statement. Unlike a scientific hypothesis, it cannot be supported or refuted by measurable experiments. Rather, the question is how such engineering and knowledge solution could be built and applied.

Figure 1.1 presents the conceptual modules of the solution. First, this application uses a focused crawler to index domain Web service descriptions accessible in the Web. The focused crawler uses a domain service model for extracting information and making uniform the information about the different services found. Next, an integration service may use this model to enrich the description about the service before storing the description in a persistent store. On the dissemination side, a data and service interface module can publish the content of the persistent store for being browsed by human and machine users, and can act as broker for accessing the contents using the indexed services. The data and service interface will use not only the domain service content but also a minimal domain content for this task. The idea is that the access to the data can be done using the native domain interface or in a uniform and simplified way by browsing the contents of the remote repositories. That is, the data and service interface module queries for data on behalf of the user and then present the data to the user in a human or machine processable form.

1.4 Methodology

Two systematic methodologies are proposed in order to provide a context of the issues related with the invisibility of the Geospatial Web services and its consequences, and the approach proposed to mitigate this issue and its consequences. The first methodology focuses on aspects related with software engineering; meanwhile, the second methodology focuses on knowledge engineering.

The following methodology is applied in each of the research modules related with software engineering identified in the previous section. It starts with the analysis of the problem. The solution of the problem is the result of a cyclic incremental development process decomposed in problem specification, conceptualization, implementation and evaluation.

1. **Analysis**. The analysis review the existing research literature and extract an inventory of evidences related with the problem that research module addresses.
2. **Problem specification**. The specification provides a rationale of the motivations or the challenges for the research question.
3. **Conceptualization**. The conceptualization activity structures a solution identifying its key elements.
4. **Implementation**. The implementation activity develops a software platform based on the conceptualization.
5. **Evaluation**. The evaluation applies the implementation to a concrete problem and evaluates its usefulness.

The module related with knowledge engineering, that is, the development of the ontologies follows the next methodological approach. The ontologies are developed within a cyclic incremental development process. It starts with the specification of the ontology and ends with an ontology that fits the specification with cyclic iterations between. The cyclic iterations can be decomposed in three activities: conceptualization, formalization and implementation. A fifth activity, evaluation, is not described below because it is embedded in the evaluation of the system where the ontology is used.

1. **Specification**. The specification not only describes why each ontology is necessary but also specifies the limits of each ontology, the procedures required for its conceptualization and formalization, its intended users and its uses. The specification task is a pre-development activity

organized around the development of an ontology requirements specification document. This activity will follow the guidelines proposed by Suárez-Figueroa et al. (2009) for the development of such document.

2. **Conceptualization**. The conceptualization activity structures the knowledge into a conceptual model, a model that represents entities or concepts found in the universe of discourse and relationships between them. The conceptual model could capture an informal view of a domain without loss of knowledge. The conceptualization activity follows the steps proposed by the Methontology framework (Fernández-López et al., 1997). Methontology is a framework for the development of ontologies based in widely recognized software and knowledge development methodologies. It is one of the most popular methodologies for the development of ontologies (used by 13.7% of practitioners according to Cardoso, 2007). In addition, Methontology has been used for building ontologies related with Web services (Della Valle et al., 2007; Amardeilh et al., 2008; López-Cobo et al., 2008) and geospatial content (Vilches-Blázquez et al., 2010; Yaguinuma et al., 2010; Muñoz-Nieto et al., 2010).

3. **Formalization**. The formalization activity transforms conceptual models into computable models. Some domain knowledge may be lost in the transformation process if the semantic constructs used in the computable model are less expressive than those required by the conceptual model. Details about the scope of the formalization activity are given in the next section.

4. **Implementation**. The implementation activity serializes the computable model in a machine processable model using a concrete syntax. Details about the scope of the formalization activity are also given in the next section.

1.5 Scope

The characteristics of the OGC Web services and its Geospatial content vary depending on several factors, such as the protocols, the kind of service, and the type of content considered. The approaches to its conceptualization and formalization are also varied. The next list attempts to specify the limits of the present investigation by adding explicit constraints to the scope of the research.

- **OGC Web service metadata**. The focus of the crawl is OGC standardized documents named service metadata or capabilities XML when encoded in XML. Service metadata can be found in registries, or made accessible through hyperlinks that encodes a HTTP *GetCapabilities* request, or equivalent, to an OGC Web service. The crawler intentionally ignores alternative metadata record schemas about OGC Web service services, such as the ISO/TC 211 Geographic Metadata schema for geospatial services (ISO 19119:2005), and Web service description specifications, such as WSDL (Chinnici et al., 2007). The support of multiple metadata record schemas about services and service description specifications is part of the future work identified in this thesis (Chapter 6).

- **OGC Web service interface specifications related with the European INSPIRE Directive**. The conceptualization is limited to OGC Web service specifications that can be relevant in the context of the European INSPIRE Directive (European Parliament and Council of European Union, 2007). The INSPIRE directive is the largest and most recent regional framework in the world that establishes shared standards between European countries for the access to digital geographic information. Moreover, INSPIRE is a reference for ongoing worldwide SDI (e.g. United Nations SDI, (UNGIWG, 2007)). The common implementation rules of the INSPIRE Directive, adopted as Commission Decisions or Regulations, ensure that the spatial data infrastructures of the Member States are compatible and usable in a Community. The OGC Web services has been indicated in the implementation rules as one of the possible implementations of INSPIRE compliant services (INSPIRE DT NS, 2008, 2009). This restriction guarantees that the contributions can benefit academic and industrial audiences interested in the implementation of the INSPIRE Directive (2010-2019).

- **Neutral conceptualization**. The OGC Web service architecture is loosely based in the *Reference Model of Open Distributed Processing* (RM-ODP, ISO/IEC 10746-2 Foundations (2009), ISO/IEC 10746-3 Architecture (2009), ISO 15414 Enterprise language (2006)) family of standards (see Percivall, 2002). RM-ODP is a reference model, which provides a framework for the standardization of open distributed processing systems. It provides precise concepts and terminology for the specification of distributed systems. RM-ODP also defines viewpoints, subdivisions of the specification of a system focused in a concern. RM-ODP defines five complementary viewpoints: enterprise, information, computational, engineering and technology. Although the OGC Web service abstract specifications (Percivall, 2002) and the collection of interface standards (Whiteside, 2007; de la Beaujardiere, 2006; Vretanos, 2010; Baumann, 2010; Schut, 2007; Voges and Senkler, 2007) are the main source of information, the terminology, concepts and relations used for the conceptualization in this thesis are mainly derived from the RM-ODP. The rationale of this constraint is to avoid a conceptualization of OGC Web services tightly dependent on OGC terminology. This approach eases the reuse of the conceptualization for describing non-OGC Geospatial Web services.

- **Knowledge formalization**. When required, knowledge is formalized using Description Logics, a family of knowledge representation formalisms (Baader et al., 2003). Description Logic languages are a subset of first-order logic more expressive than propositional calculus. Description Logic shares with classic knowledge systems, such as *semantic network* (Quillian, 1967) and *frames* (Minsky, 1974), the notion of the representation of knowledge as *network-based structures* where nodes or unary predicates characterize *concepts* or complex relationships, and links or binary predicates describe *roles* or relations between concepts. Description Logics represents the knowledge of a domain by defining relevant concepts with representational primitives, and then, using these representational primitives specifies properties of objects and individuals occurring in the domain. Description Logic is equipped with formal, logic-based semantics that provide inference procedures for deriving implicitly knowledge from the knowledge that is explicitly represented.

- **Present-day data**. The conceptualization does not address issues related with the evolution of Web services or changes in the geospatial content between two crawls or accesses. Modelling the evolution of a Web service or its contents is part of the future work.

- **Earth data and services**. Although the INSPIRE Directive constraint may legitimate a restriction of the crawl to European OGC Web services, worldwide services are considered. A consequence of this statement is that the knowledge formalization and implementation must take into account that human readable properties are multilingual. Additionally, the geographic space is restricted to the Earth. Only geometric representations where its coordinate points can be transformed into latitude and longitude using the World Geodetic System 1984 (WGS84, NIMA, 1997) as its reference datum will be considered. In surveying, navigation, and geodesy, a datum is a set of reference points on the surface of the Earth that define a reference frame which is used to describe the location of points on the Earth.

- **Dublin Core metadata**. The access to geospatial metadata stored in CSW service instances is restricted to metadata records whose metadata schema has a well-known mapping to the abstract model of the Dublin Core Metadata Initiative (Powell et al., 2007). There are available several mappings from well-known OGC and ISO XML schemas to Dublin Core. For example, the CEN Workshop Agreement 14857:2003 describes the crosswalk of the geographic metadata ISO 19115:2003b to the Dublin Core. Although, the CSW interface specification defines a core model based in Dublin Core that all service instances must support. This core model is only useful for obtaining list of metadata records because implementers are free to map native metadata schemas to the core model.

- **Simple features**. The access to geospatial data is restricted to those WFS servers whose information model can be mapped to extensions of the Geographic Knowledge Base (GKB, Chaves et al., 2005) metamodel, which was developed by the project Geographic Reasoning for Search Engines[14] (GREASE, Silva et al., 2006). WFS servers return data encoded in Geography Markup Language (GML, Portele, 2007). However, the detail of the spatial representations in GML is beyond the needs of most Web use cases. An adequate balance between simplicity and usefulness can be reached with a simplified formalization of geospatial content based in

[14] `http://xldb.fc.ul.pt/wiki/Grease`

the gazetteer data structure (Hill et al., 1999). The GKB metamodel, which is based in the gazetteer data structure, is the starting point for the formalization of the minimal content model described in this thesis.

- **Software implementation**. The software artefacts developed in this thesis should run in Java Virtual Machines (JVM, Lindholm and Yellin, 1999). Although the JVM was primarily aimed at running compiled Java programs, now other languages can be compiled and run in JVMs. Two JVM languages are used in this thesis: Java (Gosling et al., 2005) and Groovy (Koenig et al., 2007). Groovy is a dynamic language for the JVM seamlessly integrable with Java but with features not found in Java, such as closures and native syntax for lists, maps and mark-up languages.

- **Knowledge expressivity**. Description Logic languages can be very expressive at the expense of high computational complexity. However, it is acknowledged that relatively expressive systems of realistic size with exponential-time reasoning problems can be processed in reasonable time (Horrocks, 1998). This thesis will develop systems less expressive than the $SROIQ(D)$ Description Logic. $SROIQ(D)$ has useful computational properties (see Horrocks et al., 2006). Popular Java-based reasoners, such as Pellet (Parsia and Sirin, 2004) and HermiT (Motik et al., 2009b), support $SROIQ(D)$.

- **Knowledge implementation**. A concrete syntax is needed in order to store the developed ontologies and to exchange them among tools and applications. The ontologies generated in this thesis will be available in OWL 2 Web Ontology Language (OWL 2, OWL WG, 2009). The OWL 2 language is the revision of the OWL Web Ontology Language (OWL 1, McGuinness and van Harmelen, 2004). Like OWL 1, OWL 2 has been designed to facilitate ontology sharing via the Web, with the ultimate goal of making Web content more accessible to machines. OWL 2 semantics are compatible with some restrictions with the $SROIQ(D)$ Description Logic.

1.6 Contributions

This thesis coins the terms *invisible Geospatial Web* and *deep Geospatial Web.* The term *invisible Geospatial Web* highlights that the Geospatial Web content has a priori the same problems of invisibility that the rest of the Web (see Sherman and Price, 2001). The term *deep Geospatial Web* is a specialization of the term deep Web (see Bergman, 2001) that identifies worthwhile Geospatial content invisible to ordinary search engines because it is behind a Geospatial Web service. Starting from these terms, the main contributions of this thesis are the following.

- First, this thesis provides an analysis of the causes of invisibility of the Geospatial Web, and proposes a mitigation strategy that combines a focused crawler and Linked Data access to Geospatial resources.

- Second, this thesis describes the architecture of a crawler focused on OGC Web services, which has been implemented.

- Third, this thesis describes two ontologies named OntoOWS and Geo-Net that have been formalized and implemented. The OntoOWS is an ontology for describing OGC Web services based on the OGC Web services abstract architecture and the RM-ODP model. The Geo-Net ontology is a place ontology that evolves from the GKB metamodel developed in the project GREASE.

- Fourth, this thesis describes a framework named Linked OWS that provides Linked Data access to OGC Web service descriptions that use as terminology the OntoOWS ontology, and remote Geospatial content. The access to the later is based in the Geo-Net ontology. In addition, the framework acts as a basic semantic proxy. That is, the framework can act as proxy for OGC requests, and can return machine processable descriptions of the requests. Moreover, the framework acts as a RESTful wrapper (Richardson and Ruby, 2007) for remote OGC Web services. The RESTful wrapper and the semantic proxy are shown as natural extensions of the Linked Data principles. This framework has been embedded in applications.

- Fifth, this thesis presents several application cases of application of the focused crawler and the Linked OWS framework.

1.7 Thesis structure

Due to the interdisciplinary nature of this thesis, each chapter may include a background section. Figure 1.2 presents a guide of the relations between the remaining Chapters of this thesis. They are organised as follows.

Chapter 2 identifies why OGC Web services are invisible, highlights optimal strategies for an OGC Web service focused crawler and depicts its architecture. In addition, it presents applications.

Chapter 3 introduces OntoOWS, an ontology for OGC Web services, which is based on concepts and ideas of RM-ODP. It also explains how the service metadata found by a crawler can be represented using this ontology.

Chapter 4 presents a minimal content model based in the Geo-Net ontology. This minimal content model is used by the data access interface described in Chapter 5.

Chapter 5 introduces the Linked OWS engine, a data and service access interface for OGC Web service that combines Linked Data best practices and RESTful Web Services. This interface gives access to a knowledge base containing OGC Web service descriptions, and allows browsing some remote Geospatial content using the minimal content model. The chapter ends with applications.

Chapter 6 summarizes the contributions, and concludes with suggestions for future work and the central result of this thesis.

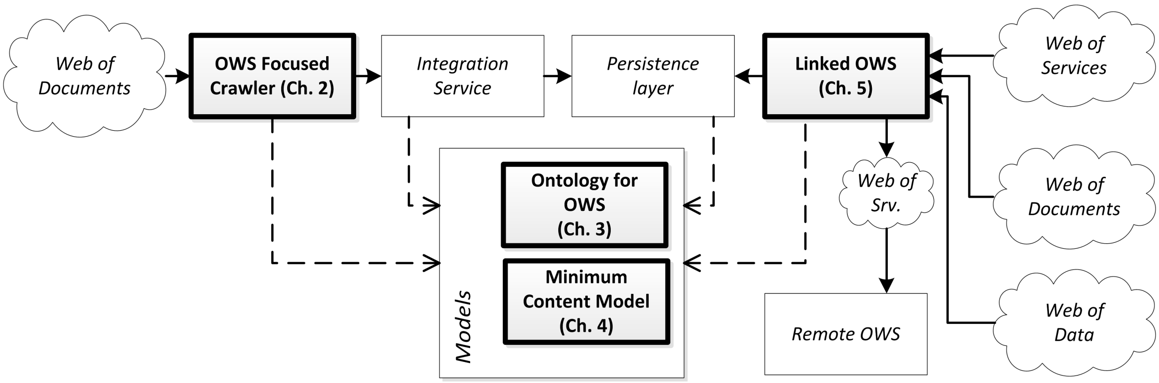


Figure 1.2: Thesis structure.

Finally, the appendixes A, B and C contain the ontologies developed in this thesis encoded in the OWL 2 Manchester Syntax (Horridge and Patel-Schneider, 2009).

Chapter 2

Crawling invisible geospatial endpoints

2.1 Introduction

This chapter presents a crawler for *OGC Web Services.* The purpose of this crawler is to index public OGC Web service instances that are ignored by standard search engines. Figure 2.1 shows that this crawler is the input module in the conceptual solution outlined in the previous chapter for the discovery of geospatial Web services, such as OGC Web services, and the access to their geospatial content.

Nobody discuss today the role of search engines to locate information on the Web. Each day, users use the search engines for finding information (e.g. weather, news) and finding specific Web sites (e.g. a booking site, a personal page). At the same time, they use search engines for obtaining Web resources such as images and music files. First, search engines have indexed pages with information, sites, images and music files using *Web crawlers.*

However, some resources are more difficult to index than others are. This chapter has its focus on the analysis of the challenges and alternatives that the designer of a crawling system must face if geospatial Web services, such as OGC Web services, are the focus of the crawl.

Searches for Web resources, such as Web services, seem to be considered as a second-class category in the research about search engines. Broder (2002) classified these searches as transactional, whose purpose is to reach a site where further interaction will happen. Rose and Levinson (2004) included these searches within a broader category named *resource searches.* The term resource searches groups those searches whose goal is to get access to an interactive resource, or collect a list of interactive resources for later use.

Rose and Levinson expressed their belief that:

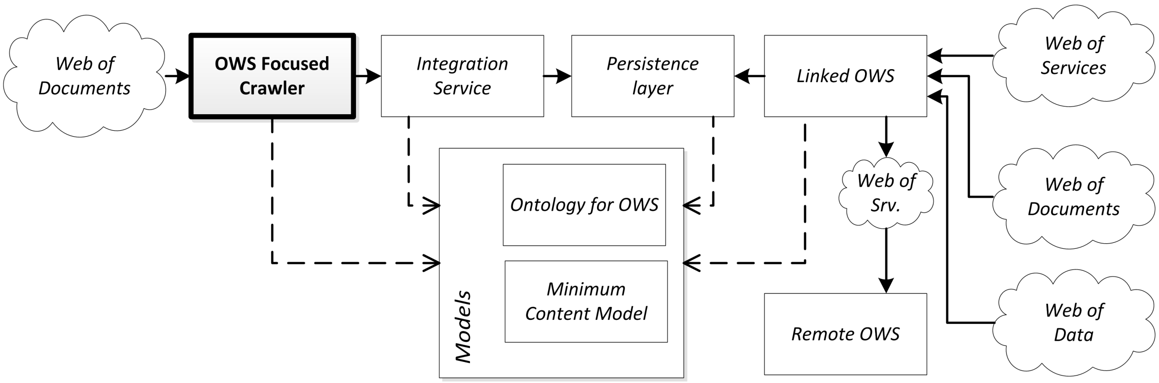


Figure 2.1: The role of the crawler for OGC Web Services in the example application outlined in the introductory chapter.

> *Resource searches are a relatively neglected category in the search engine world.*
>
> Rose and Levinson (2004)

In other words, search engine companies[1] have not yet discovered how to return the investment in the required infrastructure for crawling, indexing and searching resources. Hence, the problem of crawling geospatial Web services is part of a broader problem: how to crawl the Web to discover valuable resources under budget pressure. Search engine companies use specialized systems, known as Web crawlers, robots or spiders, for this task (see Gomes and Silva, 2008; Olston and Najork, 2010).

Search engine companies are only interested in the Web that can be monetized. This portion has two components. The monetizable Web includes part of what is named *Publicly Indexable Web* (Lawrence and Giles, 1998) or *Surface Web* (Bergman, 2001), which refers to Web resources reachable by a crawler by navigating hyperlinks. The rest is what is known as the *Hidden Web* (Raghavan and Garcia-Molina, 2001) or *Deep Web* (Bergman, 2001), which refers to public databases with valuable data available in the Web but *hidden* behind Web forms, applications and services. The rest of the Web, the content that is neither easily indexable, nor hidden and valuable, is referred in the literature as *Invisible Web* (Sherman and Price, 2001). Therefore, a search engine company divides the Web in three parts: the part where the search engine fetches indexable content for free (*Surface Web*), the part where its engineers believe that exists valuable content difficult to index (*Deep Web*), and the part that its managers consider useless for the business model of the company (*Invisible Web*).

This chapter is organized as follows. Section 2.2 presents the concepts *invisible Web* and *deep*

[1] However, the search engine world is highly dynamic. For example, Yahoo! officially transitioned its search backend to the Microsoft search platform (Bing) in the US and Canada on August 25, 2010 as part of a long-term agreement that will discontinue the efforts of Yahoo! in indexing the Web. Google and Bing are now the two main worldwide players in the search market. Regional players have risen as leaders or as strong alternative in some countries: Baidu (China), Soso(China), Sogou (China), Yandex (Rusia), Naver (South Corea) and Seznam (Cezck Republic).

Web. Next, Section 2.3 delves into the implication of these definitions in the context of the Geospatial Web. Section 2.4 discusses the notion of focused crawler for Geospatial Web services and describes the state of the art. Section 2.5 presents the challenges related with crawling this kind of resources. Section 2.6 describes the architecture of an advanced geospatial focused crawler. The implementation of a prototype based in the proposed architecture and its application to the discovery of OGC Web services are discussed in Section 2.7. Finally, the main contributions of this chapter are summarized.

2.2 The Invisible and the Deep Web

What means to be *invisible* or *deep* in the Web? The purpose of this section is to define and characterize these concepts, and then, to present the technologies that search engines use to index the *invisible* Web and to give access to the *deep* Web.

2.2.1 Definition

The literature uses the term *invisible* with the sense of ignored. Ordinary users can be aware of invisible Web content only by chance, because search engines do not index this content (Sherman and Price, 2001). The term *deep* identifies some content invisible to ordinary search engines because it is behind Web resources that play the role of interface with the content. In addition, the term *deep* calls attention to the fact that the content is worthwhile to be *surfaced*, in other words, indexed by a search engine. The concept of valuable data is broad and includes from product catalogues and real state to gene databases and satellite images. The available definitions in the literature often associate the term *deep* to the access to online databases using Web forms. "*Hidden behind search forms*" (Ntoulas et al., 2005), "*databases (...) only exposed on demand, as users fill out and submit forms* " (Barbosa and Freire, 2007), and "*the content hidden behind HTML forms*" (Madhavan et al., 2008) are good example of these definitions. These definitions are too restrictive because they do not take into consideration domain databases, such as Geospatial databases, that expose their contents through Web services. In the context of this work, the term *Deep Web* identifies the valuable content exposed on demand by submitting forms or through Web services.

Some authors have obtained approximations to the number of *deep Web* sites[2]. Bergman (2001) provided a first estimate of the size and importance of the *deep Web*: 43,000 - 96,000 *deep Web* sites and 7,500 terabytes of data[3]. That is, the size of the data of the *Deep Web* was 500 times the size of the *Surface Web* at that time. He et al. (2007) estimated that in 2004 existed 236,000 - 377,000 deep Web sites, and in average, each *Deep Web* site provides 1.5 databases, and each database supports 2.8 query interfaces. Its study also verified in a sample of 20 databases with browseable content that

[2] An estimate of size is persuasive but due the nature of the Web, it is opportune rephrasing Mark Twain: *lies, damned lies and web statistics.*

[3] These numbers should be take with care. The work of Bergman (2001) was a white paper for marketing a tool. Although its value as a research paper is clearly questionable, it opened the interest in the Deep Web.

search engines only indexed a 37% of the available content. They concluded that, at that time, most *Deep Web* sites only offer dynamic content, and when they offer an alternative link-based access to their contents, search engines cannot keep track of the changes.

The decision to index part of the *Invisible Web* or the *Deep Web* has to consider business, legal and technical issues. Few companies require a technical infrastructure comparable to the infrastructure of a search engine company. For example, Google runs multiple data centres worldwide each with several thousand machines (Barroso et al., 2003). A business decision that could increase the size of the main index of a search engine should evaluate the extent to which it uses its current and planned infrastructure, and the impact of the increment of the size of the index. Several studies estimate that the size of the main index of a search engine should increase many times to cope with the *Invisible Web* (Bergman, 2001; He et al., 2007; Madhavan et al., 2007). The decision should also take into account the liabilities related to the reuse of published content. The content owner may restrict the reuse of the published content, and it may be willing to litigate to protect its legal interests.

2.2.2 Characterization

Even if the search engine company accepts the economical and legal risks, its engineers must face several technical challenges. These challenges can be classified in the following eight categories:

- **Disconnected content**. There is no way for a standard crawler of a search engine to discover a Web resource without backlinks, in other words, a resource not linked from other Web resources.

- **Opaque content**. Search engines have a limited capability to index content. Search engines are experts in deal with HTML documents, some formats with embedded metadata and plain text content.

- **Ignored content**. Search engines do not handle some media formats, such as compression formats (e.g. ZIP, RAR) and database file formats (e.g. dBase, Microsoft Access).

- **Dynamic content**. Much of the information available on the Web cannot be browsed. Instead, it is accessed only through search forms designed primarily for human consumption or returned by services in response to submitted queries. Therefore, standard crawlers cannot index this information.

- **Real-time content**. The real-time content may be generated on the fly from databases (e.g. personalized pages) and by streaming data providers (e.g. sensor data). The real-time content is ephemeral because it has business value for a short period.

Table 2.1: Types of invisible Web content.

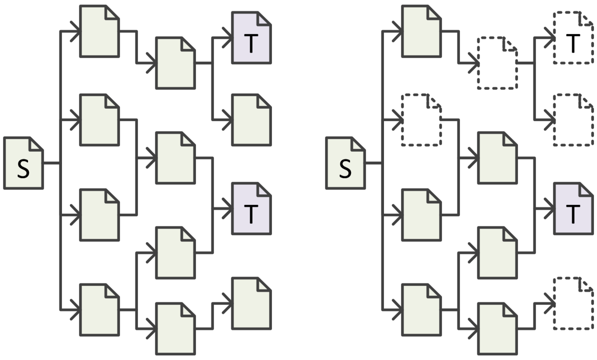
Invisible content	**Sherman and Price (2001, pg. 61)**	**Why it's invisible**
Disconnected content	Disconnected page.	No links for crawlers to find the page.
Opaque content	Page consisting primarily of images, audio, or video.	Insufficient text for the search engine to "understand" what the page is about.
Ignored content	Pages consisting primarily of PDF or Postscript, Flash, Shockwave, Executables (programs) or Compressed files (.zip, .tar, etc.)	Technically indexable, but usually ignored, primarily for business or policy reasons.
Dynamic content	Content in relational databases.	Crawlers cannot fill out required fields in interactive forms.
Real-time content	Real-time content.	Ephemeral data; huge quantities; rapidly changing information.
Contextual content	Dynamically generated content.	Customized content is irrelevant for most searchers; fear of "spider traps".
Scripted content	Sherman and Price's book is before the rise of Web 2.0 applications.	
Restricted content	This type of content is identified in Sherman and Price's book but it is not included in the list of invisible content.	

- **Contextual content**. When a user agent asks for Web content, the server returns a representation of the content that depends on the user agent's context (e.g. the languages accepted, the previous navigation sequence). Without further information, a standard crawler would ignore the alternative representations.

- **Scripted content**. Web applications have been evolving from simple HTML forms to Rich-Internet Applications (Duhl, 2003) that generate the mark-up, text and links in browser-side running script programs. *Scripted content* requires advanced techniques of inference and simulation for generating the pages that client sees in its browser (Mesbah et al., 2008).

- **Restricted content**. This category refers to server side methods that advice crawlers to not index some content or avoid the access to content. These methods are based on industrial best practices (e.g. the Robots Exclusion Standard (Koster, 2010), CAPTCHAs (von Ahn et al., 2003)), Internet standards (e.g. no-cache HTTP headers, password protection schemes) and applications (e.g. digital rights management systems).

The above list adapts and extends the classic taxonomy of invisible Web content of Sherman and Price (2001). Table 2.1 contains the equivalences between both lists.



(a) Standard crawl.

(b) Focused crawl.

Figure 2.2: A standard crawler follows each link from the seed S and download all linked documents in a distance i; the coverage of the topic T is 100% but requires the download of all the documents. A focused crawler follows only the most promising links from the seed S within a distance i and downloads a subset of the documents; the coverage of the topic T is less than 100% but this process requires much less resources than the standard crawler.

2.2.3 Indexing search forms and Web services

Parts of the large amount of information that resides in the Invisible Web can become visible if a search engine is able to download and process them. A *Web crawler* is a multi-purpose system optimized for massive downloading of Web content. A Web crawler automatically traverses the Surface web, retrieves pages and resources, and builds a mirror of the portion of the Web covered in the crawl. Web search engines use Web crawlers to build a large corpus of indexed Web pages that can be queried. The use of Web crawlers is not restricted to search engines only. Web crawlers are used in a wide number of scenarios ranging from archiving systems to analytic systems. Web archives as the Internet archive[4] and Tumba[5](Silva, 2003) use Web crawlers to collect parts of the Web for the posterity. Web analysis systems, including systems as different as spyware detectors (Moshchuk et al., 2006) and espionage systems (Zhou et al., 2005), use Web crawlers for the surveillance of the Web. A recent and well-documented review can be found in Olston and Najork (2010).

However, search forms and Web services are very sparsely distributed over the Web. For example, Chakrabarti et al. (1999) retrieved only 94 search forms about movies after crawling 100,000 pages related to movies. Li et al. (2010) found only 23 Geospatial Web services among 40,000 pages related to the geospatial domain. Thus, to efficiently discover topical resources, such as Geospatial Web services, a Web crawler should be optimized to perform a broad search, to avoid unproductive parts of the Web, and to reduce the noise in the results.

This kind of Web crawler is named *focused crawler*. The empirical findings of Cho et al. (1998) and Chakrabarti et al. (1999) established the concept of focused crawl. They found that relevant

[4] `http://www.archive.org/`

[5] Now available as the Portuguese Web Archive, `http://www.arquivo.pt/`

pages of a topic frequently link to pages in the same topic, either directly or via short chains of links. This fact implies that how a crawler chooses the next page may determine the amount of topic pages that the crawler could found. A general crawler seeded with highly relevant topical pages without an appropriate strategy will wander by irrelevant pages. However, a focused crawler that follows only promising links tends to locate a growing number of relevant pages over time. Diligenti et al. (2000) estimated that a focused crawler with a learning mechanism could retrieve between 8-12 times more documents on-topic than a standard crawler in a given period does. Chakrabarti et al. (1999) also found that two or more focused crawlers will converge on substantially on the same set of results no wonder where they started the search. The differences between a conventional crawler and a focused crawler are illustrated in Figure 2.2. The use of focused crawlers for Web forms was presented in Barbosa and Freire (2007). This approach has been also applied for finding Geospatial Web services (Li et al., 2010).

Once a search form or a Web service is indexed, it may be listed in database and service directories (e.g. Brightplanet's searchable databases directory[6], DeepPeep's form search engine[7](Barbosa et al., 2010)). In relation with the Geospatial Web, there are directories about Geospatial Web services that claims the use of focused crawlers for populating their lists (e.g. GIDB's list of web mapping services[8] (Sample et al., 2006)).

2.2.4 Accessing deep Web content

Web crawlers that deal with dynamic content are named *Deep Web crawlers* or *Hidden Web crawlers*. These crawlers first identify if a Web document is a form-based search interfaces that are designed for human consumption (e.g. Raghavan and Garcia-Molina, 2001) or a service metadata description (e.g. Al-Masri and Mahmoud, 2008a). In both cases, the crawler builds models of search forms and services for further processing. The construction of the model can be quite complex for search forms (He and Chang, 2006). The construction is less complex for services as the available service metadata often conforms to well known information models (e.g. WSDL documents in Al-Masri and Mahmoud, 2008b, and OGC Web service capabilities document in Li et al., 2010). Semantics plays a growing role in deep Web crawlers. For example, Madhavan et al. (2009) describes how the identification of typed inputs in web forms (e.g. addresses, dates, prices) contributes to a better crawl. Ontology systems may help to provide clearer semantics in search forms and services. Dong et al. (2008; 2009) reviews the state of art of metadata extraction in web crawlers.

Once the deep Web endpoint is found there are two exploitation strategies of its contents: *virtual data integration* of the contents, or *surfacing* (Raghavan and Garcia-Molina, 2001) a portion of the contents accessible from the endpoint. The endpoint can be integrated in metasearch engines (e.g.

[6] `http://aip.completeplanet.com/`

[7] `http://www.deeppeep.org/`

[8] `http://columbo.nrlssc.navy.mil/ogcwms/servlet/WMSServlet`

Yippy's general-purpose search[9], DeeperWeb's general-purpose search[10]) and vertical search engines (e.g. Google Base's product search[11], Google Scholar's articles search[12]). Integration requires virtual data integration. This technique requires that a standalone tool or a human annotator create semantic mappings between the form or the service and a unifying mediator form. The goal of this system is to provide an online semantic integration across the different sources. Its maintenance is challenging. The research in virtual data integration has as main goal the progressive automatization of quality mappings between schemas of HTML-forms (He and Chang, 2003, 2006; Das Sarma et al., 2008), services (Stevens et al., 2003; Patil et al., 2004) and data (Klien et al., 2007). General-purpose search engines prefer the surfacing approach (Madhavan et al., 2008). The research in surfacing investigates how to extract the content from interfaces with a single (Gravano et al., 2003; Barbosa and Freire, 2004; Ntoulas et al., 2005) and multiple (Raghavan and Garcia-Molina, 2001; Wu et al., 2006; Madhavan et al., 2009) parameters using different techniques (e.g. sampling, adaptive learning, correlated inputs). Further information about virtual data integration and surfacing can be found in the interesting reviews on the access to deep Web content of Ru and Horowitz (2005) and Madhavan et al. (2009).

2.3 The Invisible Geospatial Web

This section is organized as follows. First, the problems of invisibility of the Geospatial Web are identified. Next, these problems are particularized in the OGC infrastructure. This section ends with a rationale of the need of a focused crawler for OGC Web services. This rationale is founded in the service discovery problems shared with other Web service architectures.

2.3.1 Characterization

The Geospatial Web refers to the global collection of applications, services and data that supports the use of geographic information in the Web. This support should not be thought as a solution that simply puts geographic data in the Web. It is an approach to merge the geographic information with the Web in order to ease the access to geographic information. In this sense:

> **The Geospatial Web has the same problems of invisibility as the rest of the Web.**

If a developer considers to develop today a crawler for indexing better the invisible content of the Geospatial Web, the developer should answer first the next question: is this invisibility related with

[9] http://search.yippy.com/
[10] http://www.deeperweb.com/
[11] http://base.google.com
[12] http://scholar.google.com

the current technical configuration of the infrastructure of the Geospatial Web? If so, the crawler will require workarounds and heuristics that only have sense with that infrastructure.

The following list characterizes the invisible content of the Geospatial Web using the classification introduced in Section 2.2.2:

- **Disconnected content**. Slips can produce disconnected content easily. For example, it is quite common to find a Uniform Resource Locator (URL, Berners-Lee et al., 1994) pointing to a Geospatial Web service as plain text. Even if the Geospatial Web service has an incoming link, it is possible that the backlink encodes a mangled request. A request with that URL may crash the service and, therefore, the service would answer with a confusing error message. Crawlers can use workarounds that use specific knowledge of a concrete Geospatial Web services architecture. For example, Sample et al. (2006) and Li et al. (2010) craft special requests to connect to an OGC Web service.

- **Opaque content**. Geospatial XML documents, including standardized formats (e.g. OGC Web service metadata document (Whiteside, 2007), geographic dataset metadata (TC 211, 2007b)), are treated as plain text. There are few exceptions; the geospatial content of Keyhole Markup Language (KML, Wilson, 2008) documents, GeoRSS feeds (Reed, 2006), and some microformats that are indexed by some search engines, such as Google (see Google, Inc., 2010). KML is a XML grammar that encodes the representation of geographic data promoted by Google (Shankland, 2008). GeoRSS is an extension of the news syndication format RSS endorsed by OGC that adds location support to news feeds.

- **Ignored content**. Crawlers are accustomed to ignore popular geospatial formats, such as ESRI Shapefile (ESRI, 1998). However, geospatial search engines, such as GeoDiscover (Gibotti et al., 2005) and Mapdex (Bartley, 2005), proved that indexing geospatial formats is mainly a business decision.

- **Dynamic content**. The access to the geospatial content in the Web is made mainly through web clients (e.g. map clients, catalogue applications) that submit queries to standardized services (Nebert, 2004).

- **Real-time content**. This is a growing issue. Real-time data is considered as the cornerstone of the near-future geospatial world. Some authors, such as Yang et al. (2010), suggest that real-time data providers, such as Earth observation systems and sensor networks, will provide petabytes of geospatial data on a daily basis in a near future.

- **Contextual content**. The situation does not differ from the rest of the Web. For instance, the content accessible in Geoportail,[13] the official geospatial portal of Luxembourg, varies when the user agent is located inside or outside of Luxembourg.

[13] http://www.geoportail.lu/

Table 2.2: The size of the deep Geospatial Web in the work of Bergman (2001).

Rank	**Name**	**Web Size (GBs)**	
1	National Climatic Data Center (NOAA)	366,000	48.90%
2	NASA EOSDIS	219,600	29.34%
3	National Oceanographic/Geophysical Data Center (NOAA)	32,940	4.40%
11	TerraServer	4,270	0.57%
20	Alexandria Digital Library	1,220	0.16%
31	NASA Image Exchange	337	0.05%
	Total Deep geospatial Web	624,367	83.42%
	Total 60 largest Deep Web sites	748,504	100.00%

- **Scripted content**. Geospatial web applications are intensive in scripted content. For example, interactive geospatial maps clients require the use of client side scripts and script libraries, such as OpenLayers[14].
- **Restricted content**. The access to geospatial data is influenced by policies on ownership and rights. Geospatial data is often only the result of transactions under the umbrella of some intellectual property protection.

There is no reliable way to estimate the size of the invisible Geospatial Web. Few figures can be drawn from studies of the deep Web. The quantitative survey of Bergman (2001) affirmed that six Geospatial web sites out of 60 largest deep Web sites at that time contained an 80% of the deep Web content (Table 2.2). In the qualitative survey of (He et al., 2007), the sites of the Geospatial Web can be considered under the category *science,* which includes less than 3% of the databases identified. These kinds of estimations become obsolete quickly. For example, Bergman (2001) estimated that the size of NASA's Earth observing system data and information system (NASA EOSDIS) was 215 terabytes. Four years later EOSDIS stored over three petabytes (Behnke et al., 2005). In 2006 alone, EOSDIS produced over three terabytes on a daily basis (NASA, 2007).

2.3.2 The invisible OGC infrastructure

As it was said in the introduction of the thesis, the geospatial community endorses the use of OGC standards as the core of the technical infrastructure of the Geospatial Web. Usually, a Web crawler reaches an OGC-based infrastructure by its geoportal. The geoportal is a Web site that acts as single point of access that links to logically related Geospatial Web resources (Rose, 2004; Maguire and Longley, 2005; Béjar et al., 2009). The content of a geoportal is mainly based in HTML. Crawlers

[14] `http://www.openlayers.org/`

may ignore some geospatial content linked from the geoportal. Contextual content might be only available if the crawler keeps a session identification issued by the geoportal. Restricted content may be excluded from the crawl if the crawler endorses the robot exclusion policies of the site.

A geoportal may include:

- **Geospatial datasets available for download**. They are often encoded in archive file formats often excluded from the crawl as these formats are out of the scope of the search engines.

- **Web clients that allow users to perform geospatial tasks**. Web clients are increasingly complex. Their user interface often depends of scripts executed on the client side and the state of the application. It is a challenge for a crawler to extract information and new links from them.

- **Web pages playing the role of geospatial service metadata**. These pages describe which OGC Web services are used by these Web clients; each OGC Web service description may include a link that encodes a request to its respective *GetCapabilities* operation. The *GetCapabilities* operation is a mandatory operation in OGC Web services that returns a capabilities document, the description of the operations and the allowed parameters of the service (Whiteside, 2007). The capabilities XML document only provides a glimpse about the data operated by the service.

- **Web pages playing the role of geospatial data metadata**. These Web pages explain which datasets are available for download or accessed through OGC Web services; each Web page may include links to metadata documents encoded as ISO Geographic Metadata (ISO GMD, ISO/TC 211, 2003b; TC 211, 2007b) about the geospatial data. Metadata encoded in ISO GMD elsewhere should provide a semantically rich description of the datasets. Note that the service itself can be described using an extension of ISO GMD (ISO SRV, ISO/TC 211, 2005; TC 211, 2007b).

Table 2.3 summarizes the degree of technical invisibility of encodings, applications schemas, service interfaces and end-user access approaches promoted by OGC from the point of view of commercial search engines. Table 2.3 shows that part of the OGC infrastructure of the Geospatial Web is invisible for typical search engines. Especially, OGC Web services are invisible twice:

- OGC Web service descriptions encoded in XML are opaque for search engines. General-purpose crawlers treat *GetCapabilities* documents and metadata encoded in ISO GMD/SRV as plain text.

- OGC Web services are ignored when search engines look for dynamic content. The use of OGC Web service endpoints instead of parsing and processing web clients by commercial search engines has not been documented yet in the available literature.

Table 2.3: The potentially invisible resources of the OGC Web architecture.

	Functionality	**Issues**
Application Schemas		
XML Schema	Describe well-known information models; describe information models of datasets accessed by a service	Opaque; disconnected from capabilities documents in services
Encoding		
KML	Encodes a representation of geographic data for display	Visible
GeoRSS	Encodes the location in a feed	Visible
GML	Encodes geospatial data	Encoding of responses of OGC download services; opaque
ISO GMD, ISO SRV	Encodes metadata from a catalogue	Encoding of responses of OGC discovery services and standalone metadata documents; opaque
Capabilities document	Encodes service metadata	Encoding of responses to OGC operation *GetCapabilities*; opaque; often back linked from geoportals
End-user access		
Geoportal	Single point of access that links to logically related Web resources	Large parts are visible; navigation may depend of the context; some content may be ignored (e.g. compressed datasets) or restricted (e.g. `robots.txt` rules)
Web Client	Allows user to perform a task using OGC Web services	Often the content is scripted (e.g. web map clients); interaction may depend of the context; give access to dynamic content valuable in some contexts; the interface may differ from the underlying services
OGC Web service Data-oriented Interfaces		
CSW	Interface for the discovery of resources	Require links that returns a well-known response (e.g. *GetRecords*); give standardized access to dynamic content
WFS, WCS	Interfaces for the download of discrete and continuous geospatial data	Require links that returns a well-known response (e.g. *GetFeature*, *GetCoverage*); give standardized access to dynamic content
SOS	Interface for the download of sensor data	Require links that returns a well-known response (e.g. *GetObservation*); give standardized access to real-time data
OGC Web service Process-oriented Interfaces		
WMS	Interface for the portrayal of geospatial data	Require links that returns a well-known response (e.g. *GetMap*); give standardized access to digital images of datasets
WPS	Interface for the invocation of remote geospatial procedures	Awareness if exists links to operations (e.g. *Execute*); give standardized description of signatures of remote procedures

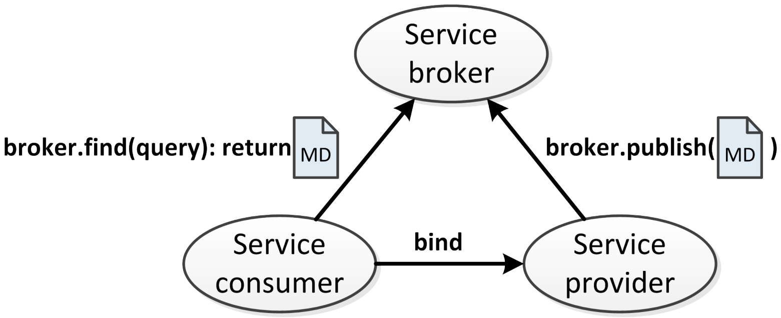


Figure 2.3: The publish/find/bind pattern. Service providers and service consumers meet each other in brokerage systems.

Virtual data integration and surfacing of dynamic geospatial content should be based in the use of OGC Web service interfaces, when available, instead of querying HTML-based forms. The detail of the standardization of OGC Web services offers a promising and sustainable alternative to access dynamic content that can avoid the challenge associated to scripted applications. For example, if the crawler looks for raw data, it should perform queries to one of the next standards: CSW for metadata records that tell where the geospatial resources are, WFS for downloading discrete geospatial data, WCS for downloading continuous geospatial data and SOS for downloading real-time geospatial data. If the crawler looks for processed data can use a WMS, that provides portrayal of geospatial data, or a WPS, that invokes remote geospatial procedures. The crawler can discover that these services are present following links that encode a request to a *GetCapabilities* operation.

2.3.3 Rationale of a focused OGC Web service crawler

The OGC reference model states that the discovery of OGC Web services should be based on the adaptation of the pattern *publish/find/bind* (see OGC, 2008). Figure 2.3 identifies three essential roles of this pattern:

- A *service provider* that publishes services to a service broker or registry, and delivers services to service consumers.
- A *service consumer* that queries the service broker to find a service provider that fits to its needs, and then, accesses to the service providers for the desired service.
- A *service broker* helps service providers and service consumers to meet each other by acting as a registry.

The service provider invokes the operation *publish* of the service broker and place (or retire) metadata about its services in a service registry. The metadata of a service typically describes its capabilities and its network bindings. The metadata has two major uses in this model: *decision support* and

Table 2.4: Vendor implementation statistics of OGC Web services (OGC, 2010).

Rank	Total[a]	Comp.[b]	Specification	Version	Profile	Year[c]
1	326	56	WMS	1.1.1	–	2001
2	217	1	WMS	1.1.0	–	2001
3	204	37	WFS	1.0.0	–	2002
4	201	0	WMS	1.0.0	–	2000
5	146	28	WMS	1.3.0	–	2006
6	122	13	WFS	1.1.0	–	2004
16	59	11	WFS	1.0.0	Transactional	2002
17	48	4	WCS	1.0.0	–	2003
18	48	13	WCS	1.0.0	Corrigendum	2005
24	36	1	CSW	2.0.1	–	2004
26	29	4	CSW	2.0.2	–	2007

[a] Total: Number of specification or interface vendor implementations (compliant or otherwise).
[b] Comp.: Number of officially OGC compliant implementations.
[c] Year: Date of the last revision of the specification.

decision-making. *Decision support* refers to scenarios where a human user has discovered a service and its metadata supports to decide if the service fits the user needs. *Decision-making* identifies automated scenarios, such as service chaining, where the metadata is used by a system to assists to its configuration and composition with other systems. The service consumer invokes the operation *find* of the service broker to discover specific service types and retrieve its metadata. Finally, the service consumer uses service metadata to *bind* to a service provider.

The use of service registries or catalogues for the discovering of Geospatial Web services in decision support and decision-making scenarios is the most popular assumption found in the literature (Singh et al., 2003; Nebert, 2004; Nogueras-Iso et al., 2005). However, the catalogue approach is based on the premise that all the Geospatial Web services are registered, and each service record contains an accurate description of the service. This is a quite strong assumption that frequently is not met in the geospatial domain (Larson et al., 2006). In addition, the development of catalogue service standards by OGC is slow with few available implementations, especially if the numbers are compared with other specifications (see Table 2.4).

Given the slow adoption and development of catalogue specifications, Mark Reichardt, President and CEO of OGC, called in 2005 for a global service registry based on SOA standards instead of OGC standards:

> *A registry of all OGC Web Services is needed for the community – a global Online Service Directory. Construction of applications or portals depends on such a 'yellow pages'. The capability does not yet exist, though the components are available. (...) A community UDDI synchronized with the UDDI Business Registries could publish a directory of all known OGC Web Service instances: Web Map Servers, Web Feature Servers, Catalogs (Z39.50), Web metadata folders, and Web Applications. It would answer the question, "What services exist globally?"* Reichardt (2005)

By 2006, service-oriented architecture registries based in the UDDI (Universal Description Discovery and Integration standard, Sabbouh et al., 2001) have failed to provide global registries of Web services, including OGC Web services. Following is a brief account of the evolution of the discovery in service-oriented architectures since the inception of the UDDI standard.

At year 2000, the promoters of SOA standards envisioned that consumers and providers of Web services would be linked with a platform for service publishing and discovery. In this vision, a rich metadata description of the service plays a decisive role. Anyone needing a Web service would go to a service broker and will select one matching with its needs. A service provider may place in the service broker a rich metadata description to get a good placement in the broker. The standard UDDI is the result of this vision for Web Services. UDDI was promoted as a possible core Web Service standard (Sabbouh et al., 2001). The UDDI v2 and UDDI v3 specifications are OASIS[15] standards (UDDI/TC, 2004). OASIS is a global consortium that drives the development, convergence and adoption of e-business and Web service standards. Important software vendors launched public search points for services named UDDI Business Registries. However, there is no public UDDI registry available since 2006[16] and the group defining UDDI closed in 2008[17].

With no available public UDDI Business registries, service providers usually advertise their services by providing access to a couple of Web pages explaining the services with links to machine processable descriptions of the services (e.g. a WSDL file). Search engines index this information, or make it available in portals specialized in Web services (e.g. seekda[18]) where service consumers can discover Web services (see Figure 2.4).

Bachlechner et al. (2006) analysed this approach to the discovery of Web Services. That work made a surprising discovery: at that time, all existing specialized portals provided probably much less coverage than a general-purpose search engine. For example, Google alone returned 25,600 pages that contain the string *wsdl*. Li et al. (2007) makes an exploratory study and ascertained 3,021 Web services by querying Google, and only found 501 services in other sources. However, as Bachlechner et al. (2006) says, the standard model of a commercial search engine is not well suited for web service discovery: search and ranking is focused on HTML content. A WSDL document is *opaque content*,

[15] Organization for the Advancement of Structured Information Standards, `http://www.oasis-open.org/`

[16] Microsoft, IBM, SAP To Discontinue UDDI Web Services Registry Effort, `http://soa.sys-con.com/node/164624`

[17] Closure of OASIS UDDI Specification TC `http://lists.oasis-open.org/archives/uddi-spec/200807/msg00000.html`

[18] `http://webservices.seekda.com/`

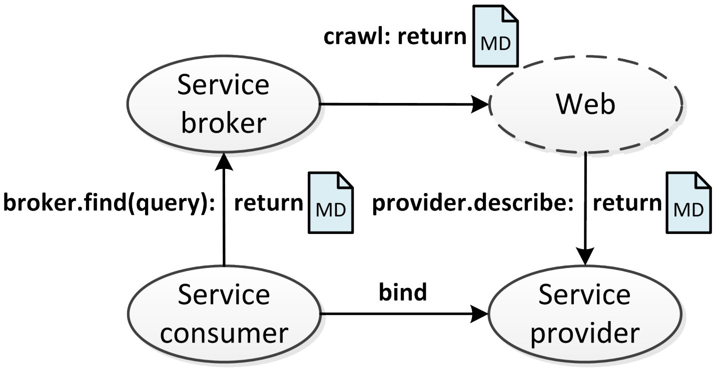


Figure 2.4: The publish/find/bind pattern revisited. Service consumers find public services in brokerage systems that crawl accessible descriptions of services.

that is, the search engine only sees the text and not its structure. The rank of the service is based on the hyperlinks pointing from and to the document and not by its qualities.

Al-Masri and Mahmoud (2008a) introduces the notion of a focused Web Service crawler and Web Service search engine. Steinmetz et al. (2008) describes the focused crawling techniques used in the European project *Service-Finder*[19]. The aim of the *Service-Finder* project is the development of a platform for service discovery based on a semantic search engine. The *Service-Finder* work in locating Web services has concentrated its efforts in the detection of WSDL service descriptions. Its approach for prioritizing the crawl of WSDL related documents is based on fixed rules that modify the priority of each new URL based on the URL contents (e.g. a URL containing the string *api* is prioritized). This crawler found more than 28.000 Web services (Steinmetz et al., 2009), and its ideas are the base for the Web service crawler used in the discovery platform *iServe* (Pedrinaci et al., 2010) of the European project *SOA4ALL*[20]. Today, Web service search engines backed by focused Web service crawlers are the only viable alternative for finding public Web services.

Table 2.5 shows the parallelisms between the initial SOA and the current OGC architecture. Both are based in the same pattern for the discovery of services. Both consider the service broker a key element for finding services. Finally, public brokers have failed as providers of public services for the SOA community, being replaced by search engines. In the OGC community, there are search engines specialized in OGC Web services coexisting with CSW based registries. Given the evolution of general purpose SOA and the parallelisms between the initial SOA and the OGC architecture, it is natural to consider search engines specialized in OGC Web services based in crawlers focused on OGC Web services as the future solution for the discovery of OGC Web services.

[19] http://www.service-finder.eu/

[20] http://www.soa4all.eu/

Table 2.5: Equivalences between the approaches of SOA and OGC to service discovery.

Element	SOA	OGC
Service description	WSDL	OGC Web service metadata
Service broker API	UDDI API	CSW
Service broker information model	UDDI	ebRIM, ISO GMD/SRV

2.4 Crawling geospatial Web services: state of the art

The *Spatial Information Search Engine* (SISE) prototype developed by Bai et al. (2003) could be considered as the first Web crawler focused on OGC Web services. This prototype introduced a heuristic that assumes that many OGC Web services are part of the *disconnected content* of the geoportals: each URL found is tested with a HTTP GET *GetCapabilities* request to check disconnected WMSs. If the response is an OGC Web service metadata document, SISE parses the document and its main elements (e.g. operation, layers, bounding boxes, etc.) are stored in a database. This system was able to find two out of a handful of testing WMSs that existed at that time.

Mapdex[21] was the first example of a viable Geospatial Web map service search engine (Bartley, 2005). Focused to the popular ESRI ArcIMS web map servers (Waxman, 2000), *Mapdex* collected soon over 25,000 ArcIMS and 100 WMS querying the Google Search API. *Mapdex* was the proof that the use of Web crawlers, even if they are the crawlers of a commercial search engine, is a viable alternative for finding Geospatial Web services. Mapdex was discontinued in 2006 when its main developer was hired by ESRI.

The growing number of on-line WMSs and the nonexistence of a distributed network of catalogues listing the available WMSs boosted the interest in the use of search engines to create list of WMS. For example, Refractions Research[22] (Reichardt, 2005; Refractions Research, 2006) conducted between 2005 and 2006 surveys querying the Google Search API for pages with an explicit *GetCapabilities* request in its URL. Its surveys discovered more than 600 WMS worldwide.

Skylab-Mobilesystems[23] (Skylab, 2005; Schutzberg, 2006) is the first of a series of WMS lists compiled by GIS software and services vendor companies that claims that they were compiled with a home-grown crawler or by querying a search engine that includes, among others, *Microimages*, *Geosoft*, *Mapmatters*, *Geopole* and *WMS-Finder*. The lack of technical details about the crawling system is usual in these lists. *Microimages*[24] (Microimages, 2008) maintains a collection of more than 2,500 OGC Web services and more than 20,000 ArcIMS services. According to Ager et al. (2010), a Web crawler feeds this large collection. *Geosoft* uses a Web crawler to build a catalogue

[21] http://www.mapdex.org/index.cfm

[22] http://www.refractions.net/

[23] http://www.skylab-mobilesystems.com/en/wms_serverlist.html

[24] http://www.microimages.com/wmscatalog/

of OGC Web services and ArcIMS and integrate them in a globe viewer (Burns, 2009). Although the catalogue is available to *Geosoft* clients, the technical details about how the Web services are discovered are scarce. *Mapmatters*[25] (Geops, 2010) is a portal that offers search and monitoring of WMS. Müller and Mandery (2009) explains that the service list has been obtained by querying Google. *Geopole*[26] (Kalberer, 2010) and *WMS-Finder*[27] (Alta4, 2010) maintain respectively lists of more than 3,000 WMS whose origin is unknown.

Sample et al. (2006) developed[28] a system for the automated discovery and binding of WMS services. The open source Web crawler *Heritrix* (Mohr et al., 2004) was used for finding WMS service metadata documents. The results are accessible through a WMS broker offered in the *Naval Research Laboratory's Geospatial Information Database*[29] (GIDB) portal that has become in one of the largest originators of WMS content. The GIDB WMS broker republish the discovered WMS services and, additionally, provides access as WMS to vendor geospatial services. These vendor geospatial services include ESRI ArcIMS, Autodesk's MapGuide, and legacy US government protocols. The vendor geospatial services collection of the GIDB is handmade. In an attempt to reduce the set of pages to crawl, the system is feed by GIS related Web pages provided by Google. GIDB uses a variant of the heuristic employed in SISE: the *GetCapabilities* request is added only if the URL does not contain a query string.

The system *Data Discovery and Retrieve Service* (DDRS) includes a module that harvests the Web for finding OGC Web services (Chen et al., 2008). The harvest service uses *Nutch*, an open source Web crawler (Cafarella and Cutting, 2004), for creating a collection of Web resources, and then, to mine the content for finding documents with promising links not only to WMS but also to WCS, WFS and SOS. If such a promising link is detected, its system creates an appropriate HTTP GET and POST *GetCapabilities* request. This work is based on a previous system focused to the discovery of WMS (Chen et al., 2007). A similar system focused in WFS is described in Jiang et al. (2008).

Wang and Wang (2009) proposes an agent-based architecture whose goal is to aggregate images from several WMS. This architecture requires a registry of existing WMS. One of the agents has the goal of crawling the Web for finding WMS services. As seed, the Web crawler uses Google search results.

Li et al. (2010) tackles the development of a real WMS-focused Web crawler. Its crawler explores the Web sorting the links with the help of a basic self-learning approach. The priority of the URL is computed using an accumulated term frequency (ATF) based model that learns from URLs that links directly to WMS service metadata documents. The ATF model probed to be more efficient than breadth-first approach in the evaluation for finding WMSs. Its crawler applies a basic deep

[25] `http://www.mapmatters.org/`

[26] `http://www.geopole.org/`

[27] `http://www.wms-finder.de/`

[28] And patented(see Sample et al., 2007).

[29] `http://columbo.nrlssc.navy.mil/ogcwms/servlet/WMSServlet?REQUEST=ServiceLinks`

Web heuristic similar to the used by Sample et al. (2006): if the next URL to processed is likely to link to a WMS service instance then the crawler also sends a valid *GetCapabilities* request to that service instance. This crawler found 1126 services, and, although its absolute performance no was greater than approaches based in query search engines (Skylab, 2005; Sample et al., 2006; Refractions Research, 2006), the crawler showed that was it was better locating all WMSs in a single host. Li et al. (2010) believes that the number of services found with its focused Web crawler underrepresents the total amount of WMS as many of them are situated in the deep Web. They also suggest that WMSs seems to be part of a small-scale linkage graph, that is, the WMSs found are leaf nodes of a small-world network of pages about geospatial Web services. However, these results are far from the number of WMS discovered by the crawlers of WMS search engines, such as *Mapmatters* (1521 WMS), *WMS-Finder* (3023 WMS) and *Geopole* (4339 WMS). Nonetheless, these search engines do not disclose details about their crawling engines.

Lopez-Pellicer et al. (2010a) applies the crawler architecture described in this chapter for the discovery of different types of geospatial services in the context of the application of the European directive INSPIRE (European Parliament and Council of European Union, 2007). The investigation is focussed in OGC Web service standards that have been indicated as possible implementations of INSPIRE compliant network services, i.e. WMS for view services, WFS and WCS for download services, WPS for invoke spatial service services and transformation services), and CSW for the discovery services. Using as seed Web pages retrieved from Google, Yahoo! and Bing, the crawler found 6,544 OGC Web services in Europe. 3,714 services found where WMS.

Table 2.6 summarizes the number of Geospatial Web services found by focused crawlers in the literature, and details the specifications targeted in each work.

2.5 Challenges: best paths, crawl ordering and coverage

Given the dynamic nature of the Geospatial Web, where new OGC Web services are constantly added, manual approaches to create and maintain a collection of geospatial services are not feasible. As was discussed in previous sections, it is important to automatically discover OGC Web services. Nevertheless, OGC Web services are very scattered over the Web, even more than other entry points to invisible content. For example, Li et al. (2010) found only 23 geospatial Web services among 40,000 pages related to the geospatial domain. Thus, a Web crawler focused on OGC Web services must perform a broad search of the Web and, at the same time, avoid potentially unproductive sites. That is, the crawler should encode strategies and heuristics about *geospatial paths, crawl ordering* and *coverage*.

1. **Heuristics for geospatial paths**. A focused crawler should start with good *seeds*, i.e. starting pages. A good seed is a page that acts as a hub to relevant pages. Many relevant pages should be accessed from a good seed within a *small* number of *hops*. How big is small?

Table 2.6: Comparison of number of Geospatial Web services found by focused crawlers.

Authors	Year	Total	WMS	WFS	Other
Bai et al.	2003	2	2	0	0
Bartley	2005	51,994	129	0	51,865[a]
Skylab	2005	904	904	0	0
Refractions Research	2006	612	612	0	0
Sample et al.	2006	1,187	1,187	0	0
Microimages	2008	22,731	2,416	144	20,171[a]
Li et al.	2010	1,126	1,126	0	0
Geops	2010	1,521	1,521	0	0
Kalberer	2010	4,339	4,339	0	0
Lopez-Pellicer et al.	2010	6,544	3,714	1,492	1,338[b]

[a] ArcIMS
[b] Other OGC Web service specifications

This value is dependent on the topic and it is not necessarily the same for all the relevant pages connected. Which are the best hops? Typically, the hops are between connected pages. Geospatial disconnected content may require the application of heuristics for accessing to them.

2. **Crawl ordering policies**. A focused crawler attempts to crawl first those pages that seems to be relevant to a topic to improve its efficiency. The rationale of this decision is to minimize the overall number of downloaded Web pages for processing and, at the same time, maximize the percentage of relevant pages. In a context of scarce resources, which is the priority strategy that minimize the number of downloads of non-geospatial resources?

3. **Coverage**. The size of the Geospatial Web is unknown, thus, when a crawler should stop downloading the Web? Is it possible to establish a coverage objective (e.g. 95%)? How a crawler could know that it has not missed a part of the Geospatial Web when it stops?

2.5.1 Heuristics for geospatial paths

A geoportal is a Web site that acts as single point of access that links to logically related Geospatial Web resources including OGC Web services (Rose, 2004; Maguire and Longley, 2005; Béjar et al., 2009). The next list presents different heuristics for the discovery of paths that can be applied when a *geoportal* is explored:

- **Navigable link that returns an OGC *GetCapabilities* response**. Geospatial service providers often publish their services by providing access to a couple of Web pages explaining the geospatial services with links that may encode a *GetCapabilities* request. A crafted request can reveal the presence of other OGC Web services in the same address.

- **Navigable link to an OGC Web service endpoint**. If the link returns a standard OGC Web service error message, the crawler should craft appropriate *GetCapabilities* requests. As above, a crafted request can reveal the presence of other OGC Web services in the same address.

- **Navigable link that returns an ISO GMD**. Publishers can also provide links to rich descriptions of the service encoded in ISO GMD. An ISO GMD is a standard XML document that encodes information for decision support and includes information about the network address of the service. The situation is similar to the previously described as the link can be mangled.

- **Plain text URLs**. The Web based publication of services is not standardized. Publishers can publish the address of the service or its metadata as plain text in a Web page. Moreover, the link can encode a mangled OGC Web service request that returns a standard OGC Web service error message. Hence, the crawler should follow any potential link in the document. If the link returns an OGC Web service metadata document should proceed as was described above. If the link return a standard OGC Web service error message should craft appropriate *GetCapabilities* requests.

- **Dynamic content**. The collection of OGC Web services accessible from a geoportal can be hidden behind a CSW endpoint. That is, the focused crawler should be able to perform a deep Web crawl of metadata repositories. The information model of CSW is well defined. Lopez-Pellicer et al. (2010b) shows that CSW can be easily crawled.

- **Scripted content**. The links to OGC Web services can be found in the source code of Web applications. The links can be hidden among the scripts of the application (e.g. web mapping clients based in the Java Script library OpenLayers[30]). In addition, the web application can be a simple search form (e.g. *Microimages*[31]) or a complex scripted application (e.g. IDEE catalogue[32], IDEC catalogue[33]) that give access to a database with links to OGC Web services.

An efficient focused crawler should explore the possible paths that the OGC architecture offers. At least, it should:

[30] `http://openlayers.org/`

[31] `http://www.microimages.com/wmscatalog/`

[32] `http://www.idee.es/search/indexLayout.jsp?PAGELANGUAGE=es`

[33] `http://delta.icc.cat/SDIExplorer/cercaCataleg.jsp?lang=es_ES`

- **Test the presence of a disconnected OGC Web services**. The crawler should be able to craft *GetCapabilities* requests for the discovery of disconnected services.
- **Process OGC and ISO metadata**. The crawler should parse geospatial metadata encoded in XML (e.g. OGC Web service metadata documents, OGC Web service exception report documents, ISO GMD documents) being aware of its information model.
- **Deep Web crawl of CSW found**. The crawler should harvest standardized geospatial discovery services as CSW that can contain geospatial metadata.
- **Aggressive HTML parsing**. The crawler should parse HTML documents for finding references to OGC Web service endpoints in the texts. An aggressive parsing may discover links hidden among the source code of the strings and in plain text.

Once a Geospatial Web resource is discovered, how many neighbours should be examined for finding additional Geospatial Web resources? Which is the maximum of hops between resources to be considered? There is no available a study about the mean distance between relevant geospatial resources. Hence, the maximum number of hops between relevant resources is an arbitrary parameter. This fact should be considered in the analysis performed about crawl ordering policies in the next section.

Little information is available related to Geospatial Web metrics. Studies about state-of-the-art geoportals are focused in the geoportal as a concepts or as part of an architecture (Bernard et al., 2005; Maguire and Longley, 2005; Yang et al., 2007; Béjar et al., 2009; de Longueville, 2010). An exception is Li et al. (2010) that proposes that OGC Web service providers seems to be part of a small-scale linkage graph, that is, the OGC Web services found are leaf nodes of a *small-world network* (Adamic, 1999) of pages about geospatial Web services where nodes are spatially clustered. A small word is a graph highly clustered, like regular lattices, with small path lengths, like random graphs. If the hypothesis of Li et al. (2010) is true, hub sites related with worldwide SDIs, such as the geoportal of Global Spatial Data Infrastructure Association[34], are the best candidates for seeding the crawler.

2.5.2 Crawl ordering policies

First, this section presents different crawl ordering policies, analyses their application in the available literature about geospatial crawlers, and then, suggests specific recommendations that should be take into consideration in the design of a crawler focused on OGC Web services.

The efficiency of a focused crawler depends on how the crawler identifies pages with higher probability of giving access to relevant content. As a rule, an appropriate crawl ordering policy increases the ratio of resources in scope among the downloaded pages (Baeza-Yates et al., 2005),

[34] http://www.gsdi.org/

and may improve the fraction of downloaded resources in scope in relation with the total of resources available in the Web (Chakrabarti et al., 1999). In other words, an appropriate crawl ordering policy increases the precision and could increase the recall of crawling sessions. The recall is related with how greedy is the policy. A crawling policy too greedy will traverse few bridge pages, finding few relevant pages. On the opposite, a policy too relaxed will behave as a non-topical crawler. This problem is known as exploration versus exploitation (Pant et al., 2002). Figure 2.5 presents hypothetical curves of the described behaviours. Nevertheless, random ordering (Boldi et al., 2005) and FIFO queues (Najork and Wiener, 2001), also known as breadth-first strategies, may perform well in precision and recall in some scenarios.

The crawl ordering policy used in focused crawlers can be categorized as follows:

- **Fish Search** (De Bra and Post, 1994) crawlers evaluate the page and, if the page is in scope, they add its links ahead in the list the links found. Children pages in a radius r from the relevant page are also crawled, and their links are added after the links from pages in scope.

- **Shark Search** (Hersovici et al., 1998) is a variant of De Bra's Fish Search. Each out link has a score equals to the score of its parent pages minus a decay factor. The queue is ordered with this value. When the page is downloaded, it is computed a similarity metric with the topic. If the similarity is above a threshold the page is considered in scope, and then, its score is updated to the computed metric. This strategy forces the crawler to traverse first the most relevant bridge pages, that is, off-topic or irrelevant pages.

- **Best First** (Cho et al., 1998) crawlers assign to each link a priority estimate that only uses information related to the link seen and remembered by the crawler. Each link of the resource is explored, traversing first the most promising links. Different strategies can be used, and can be as complex as desired. For example, the link selection process may be guided by computing a lexical similarity metric between the topic's keywords and the terms of the source page for each link (Menczer et al., 2004).

- **Learning mechanism** (Chakrabarti et al., 1999). The crawler learns from the content of the resources discovered, and uses this knowledge to classify and order the links. Aggarwal et al. (2001) proposes a holistic approach that combines self-learning arbitrary predicates on page content, URL strings, and link structure statistics. There are different approaches for learning: *Bayes Classifiers* (Chakrabarti et al., 1999), *Context Graphs* Diligenti et al. (2000), *Decision Trees* (Li et al., 2005), *Neural Networks* (Pant and Srinivasan, 2005), *Support Vector Machines* (Pant and Srinivasan, 2005) and *Hidden Markov Models* (Batsakis et al., 2009). As Diligenti et al. (2000) points, learning approaches must face with the fact that some bridge pages leads to relevant content. Barbosa and Freire (2007) proposes to learn not only is the page is in the scope but also the distance to previous relevant content.

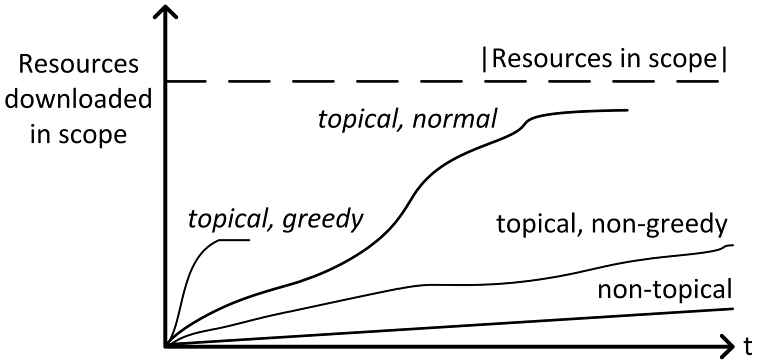


Figure 2.5: Greedy policy versus non-greedy policy.

- **Semantic relevance** (Ehrig and Maedche, 2003). A document can be in the topic although lexically it does not match with topic's documents. The crawler computes the score of the links using a knowledge base or ontology instead the topic's documents. Entities relevant to the knowledge base are extracted from the page, and then, the score is computed using several measures that may take into account the relations in the knowledge base of the extracted entities. Semantic relevance can use learning mechanism. Xu and Zuo (2007) proposes the use of knowledge representation to describe the crawling session and mining topical rules that can give clues of relevance of future links.

The literature containing the description of crawl ordering policies in the geospatial domain is scarce. The work of Ahlers and Boll (2009) describes an algorithm that implements the idea of decay factor introduced by the Shark search policy. The decay factor is arbitrary as it was discussed before. Li et al. (2010) uses a basic self-learning mechanism that uses the URL of the pages combined with a weighted list of terms. The goal of this work is traversing first links whose URL looks like a request to a geospatial service found in pages whose content is about geospatial services. Fritz et al. (2010) has developed a breath-first crawler for mining Web information with geographical content in emergency management scenarios. Other crawler systems developed for the geospatial domain, such as GeoDiscover (Gibotti et al., 2005) and GEDMWA (Pultar et al., 2008), do not disclose details about the crawl ordering policy. The default policy of the open-source crawlers used in Sample et al. (2006) (*Heritrix*) and Chen et al. (2008) (*Nutch*) is FIFO. It is not documented if these works have implemented their own crawl ordering policies.

An efficient focused crawler should allow to plugging any of the priority assignment systems described above, or a combination of them. From the description of the paths of the OGC architecture and the experience learn in the state-of-art OGC-focused crawlers, an efficient crawl ordering system should consider the support:

- **Non-opacity of URLs**. OGC Web service query strings are easy to identify (see Whiteside, 2007), and some software vendors present patterns in the URI of their OGC Web services (e.g.

ESRI ArcIMS WMS connector (ESRI, 2003)). The priority estimation should consider first the structure of the URL of the link (e.g. the tokens, the keyword-value pairs in the query fragment) and the type of link.

- **Hybrid priority policies**. Simple crawl ordering policies, such as FIFO and fish search, should be avoided. The priority estimation should include information derived from the URL, from the content (e.g. an hyperlink, plain text, link found in an ISO GMD), and other factors, such as the priority estimated to the page modified with some decay value. That is, the crawl ordering policy should combine a shark search policy with a policy based in best first, learning mechanism or semantic relevance.

- **Dynamic priorities**. The priority of a Web page and its links should not be considered as a static value. If a link leads to an OGC Web service, it is possible that a sibling link could link to an OGC Web service too. This situation can appear in medium and large geoportals, service directories and geospatial development forums.

In addition, a focused crawler should use multi-level FIFO queues, each with different priority, instead of a single ordered queue. Multi-level queues reduce the reordering effect of dynamic priorities as a change on the priority of a link implies only to remove its entry from a queue, and then, to add the entry to a queue with lower or upper priority.

2.5.3 Coverage

The Web has an intractable large size in both the surface and the invisible side. It is impossible to download the entire Web. A focused crawl has the same problem. The WMS-focuser crawler of Li et al. (2010) found only 23 WMS after downloading 40,000 pages. However, Lopez-Pellicer et al. (2010a) claim that there are more than 6,000 services only in Europe. Should a crawler download ten million[35] of pages to be confident enough that it has find “all” the public OGC Web services?

This section analyses different methods for the estimations of the size of a topical Web resource: *sampling*, *pairwise overlap* and *capture-recapture*. Then, the problem of dependency between crawls is introduced. Sampling and pairwise overlap do not considers dependency. In other words, crawl sessions that start from different seeds are considered independent. This assumption implies that if d_i is the set of Web resources in scope in a non-focused crawl C_i, $1 \leq i \leq n$, then:

$$Pr\left(d \in d_i | d \in d_j\right) = Pr\left(d \in d_i\right), \, i \neq j \tag{2.1}$$

That is, the conditional probability of finding a relevant resource in the crawl C_i given finding the same relevant resource in the crawl C_j is the same as the unconditional probability of finding

[35] $40000/23 * 6000 = 10434782$

that resource in the crawl C_i. In its seminal work about focused Web crawlers, Chakrabarti et al. (1999) discovered empirically that in there is topic convergence on focused walks. This has been verified in several studies (Chakrabarti et al., 2002; Menczer, 2004). Topic convergence means that no wonder where a focused crawl starts the resources collected from these crawls overlap. In other words, in a focused crawl often:

$$Pr\left(d \in d_i | d \in d_j\right) \geq Pr\left(d \in d_i\right), \, i \neq j \tag{2.2}$$

Some capture-recapture methods incorporate the influence of positive dependency in their estimations. Finally, this section summarizes the ideas behind the capture-recapture methods, and when these methods can be applied can be applied in a focused crawler.

As it was said above, the literature identifies three methods that could estimate the size of a topical Web resource:

- **Sampling**. He et al. (2006) proposes to divide the crawling space in scopes and by sampling estimate the relevant topics per scope. Then, the average of local estimates can then predict the total performance. This technique has been applied to estimate the number of deep Web databases in He et al. (2007).

- **Pairwise overlap**. The idea is quite simple: the proportion of the overlap between the two crawls that are in one of the crawls should be similar to the proportion of the total relevant population that has been crawled in the second crawl. This technique has been employed to measure the size of the Web (Lawrence and Giles, 1998; Gulli and Signorini, 2005) and the number of deep Web databases (Bergman, 2001).

- **Capture-recapture**. The capture-recapture sampling and models have been widely used to estimate parameters of populations where the samples are dependent (Chao, 2001). This technique has been applied to estimate the size of the content of deep Web databases (Lu and Li, 2010).

Sampling and pairwise overlap assumes independence between crawls. However, dependence between crawls is a phenomenon that can happen even if the crawl is not focused. There are two main causes:

- **Local dependence** of relevant resources, that is, being crawled in a crawl has a direct causal effect in being crawled or not again. For example, two Web resources belonging to the same topic may be closer than any other pair of Web resources. The local dependence is positive or negative. For example, if two independent crawlers with the same configuration reach the same page from different seeds it is highly probable that the subgraph accessible from that page will appear in both crawls. This is a positive local dependence. If the server of the page forbids access to crawlers after the first visit, it is highly probable that the subgraph will appear only in one of the crawls. This results in a negative local dependence.

- **Heterogeneity** among relevant resources; it happens in situations where the crawler work with several topics with different prevalence in the Web. This is a statistical paradox. The different population of the topics causes a bias in the estimators.

In addition, some studies suggest that some topics in the Web, such as universities (Thelwall, 2002) and geographical Web services (Li et al., 2010), near things are more linked that far things. Hence, two Web resources belonging to the same topic and published or about the same geographical area, may be closer than other web resources in the same topic.

Several estimation models with dependence have been proposed in the literature. Chao (2001) classifies these models into three categories: ecological models, the sample-coverage approach, and log-linear models. Ecological models are based on empirical investigations on animal populations. The sample-coverage approach measures the overlap information to quantify population and identify dependences. Log-linear models model the problem as an incomplete 2^t contingency table of t samples for which are unobserved rows, fits a log-linear model and then projects the model on the unobserved rows.

Log-linear models fits better with the nature of the crawler results. An example with data from three crawl sessions can be illustrative. The contingency table has seven observed values: z_{100}, z_{010}, z_{001}, z_{110}, z_{011}, z_{101}, and z_{111}. The value of the row z_{100} is the number of relevant resources only found in the crawl C_1, the value of the row z_{110} is the number of relevant resources found in the overlap between crawl C_1 and C_2 but not in C_3, and so on. The unobserved value z_{000} is the number of relevant resources not found by the crawls C_1, C_2 and C_3. The log-linear approach models with a logarithm the expected values for each row. The most general log-linear model for three crawls is:

$$\begin{aligned} log\, E\left(z_{ijk}\right) =& u + u_1 I\left(i=1\right) + u_2 I\left(j=1\right) + u_3 I\left(k=1\right) + \\ & u_{12} I\left(i=1, j=1\right) + u_{13} I\left(i=1, k=1\right) + u_{23} I\left(j=1, k=1\right) + \\ & u_{123} I\left(i=1, j=1, k=1\right) \end{aligned} \tag{2.3}$$

The function $I(A)$ represents the occurrence of the event A. The parameters u_{ij} indicate interaction between two crawls, meanwhile the parameter u_{ijk} indicates an interaction between the three crawls. Once the log-linear model is resolved, the unobserved value z_{000} is estimated as:

$$E\left(z_{000}\right) = e^u \tag{2.4}$$

Libraries, such as CARE (Chao et al., 2001) and R-Capture (Baillargeon and Rivest, 2009), provide support to a variety of capture-recapture techniques, including log-linear models.

Topical and geospatial dependences might occur between crawl sessions of a crawler focused on OGC Web services. The population size estimators and the information about dependence can be applied to stop a crawl session given the following assumptions:

- **Population closed**. There is a set of previous recent crawler sessions with the same configuration as the current crawl session that only differs from the current session in the seeds. That is, neither the target of the crawl varies, nor a sudden increase or decrease of the total of targeted OGC Web services is expected during the crawling sessions.

- **Contingency table**. The contingency table is part of the crawler data model, and it is populated with the results of the considered crawler sessions. The contingency crawl is updated with the results of the current crawl. The contingency table includes the predicted values of resources not found of previous crawls.

- **Non-incremental crawl**. The crawl session does not use data from the crawlers sessions considered for the contingency table for detecting duplicated or visited resources.

The crawler should halt the crawler session when detects with the capture-recapture methods one of the following situations:

- **No significant new results**. The estimate standard error of the estimated number of OGC Web services becomes under a specified threshold, and then, the ratio of the estimate number of OGC Web services not discovered with respect the estimate total of OGC Web services is below a specified threshold. Continuing the crawl session could not lead to the discovery of a significant number of new relevant resources.

- **Revisited OGC Web services**. The log-linear model that models the contingency table shows that exists a high degree of dependence between this crawl and previous crawls. Continuing this crawl session might imply to revisit the same sites that were visited in the previous crawls sessions.

2.6 Architecture of an advanced geospatial crawler

An advanced geospatial crawler is a focused crawler optimized for the discovery of geospatial content. An overview of its architecture should reveal that it does not differ essentially from other crawlers (see Mohr et al., 2004; Gomes and Silva, 2008; Olston and Najork, 2010), but offers extension points to add new functionality. The key characteristic of the advanced geospatial crawler is that the challenges identified in the previous section are addressed in these extension points. Next, these issues are presented in detail.

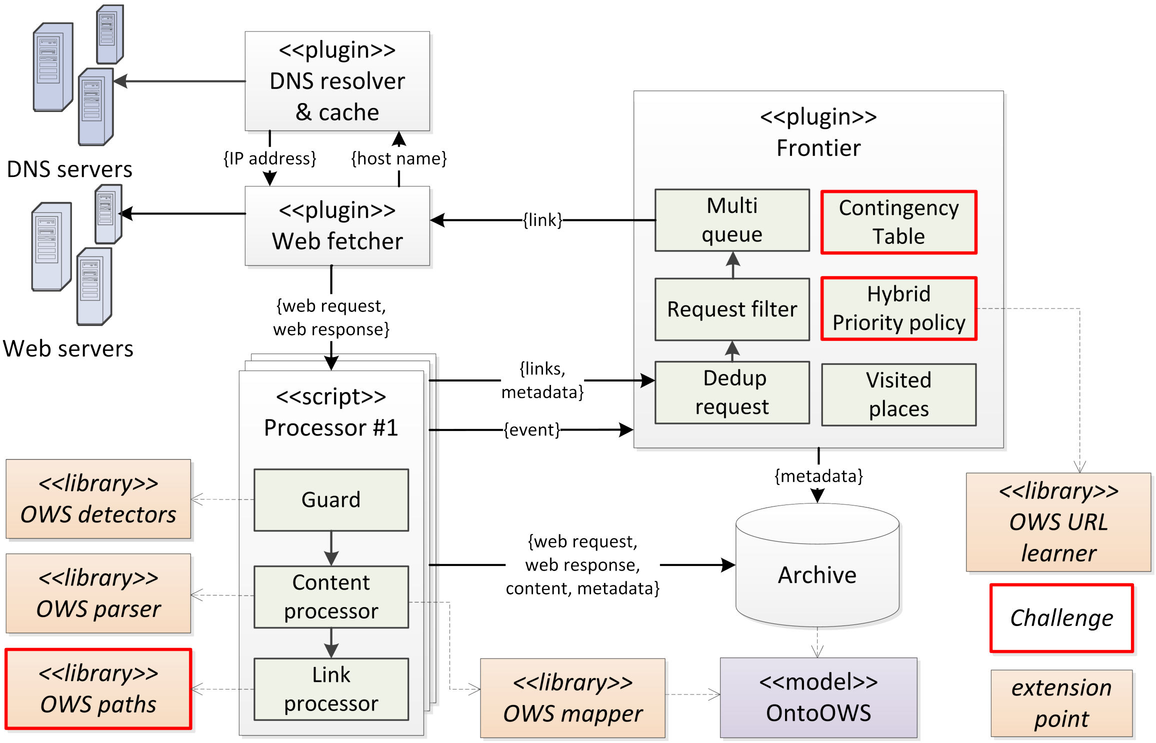


Figure 2.6: Architecture of an advanced crawler of OGC Web services.

2.6.1 Architecture overview

Figure 2.6 shows the high-level architecture of an advanced geospatial crawler with points of extensions focused on OGC Web services. Details on the extension points of the architecture and the support for the discovery and process of Geospatial Web services, exemplified in the support of OGC Web services, are given in Section 2.6.2 and Section 2.6.3 respectively. Hence, this Section provides a description of the architecture as generic as possible.

The Web crawler system is composed of one or multiple crawling processes running on different machines. Each crawling process consists of a control thread and a managed set of worker threads. The control thread finalizes when all the worker threads are stopped and the *frontier* data structure is empty (or halted, see below). The frontier data structure is a pluggable structure that contains a collection of links. Each link contains the target URL, an optional payload, and additional information, such as the HTTP method.

If there are stopped worker threads available in a thread pool, the control thread obtains a link from the *frontier* data structure, and then, the control thread starts one worker thread from the thread pool to crawl the link. First, the worker thread invokes the appropriate *Web fetcher*. The operation of each fetcher is specific of the protocol and the kind of request. Without loss of generality, let us analyse a typical fetch using a *HTTP fetcher*.

By default, the fetcher calls first the *DNS resolver* to resolve the hostname to an IP address. Resolved IP addresses can be cached in to reduce DNS traffic. In some scenarios, the use of a cached IP might lead to undesirable responses. Hence, the link metadata might hint about the use of cached IPs. Additionally, the resolver tells to the fetcher if the host has been visited. If this request is the first visit to the host in this crawling session, and the crawler is configured to respect robot inclusion/exclusion rules, the request is returned to the head of the frontier's queue and then replaced by a request to a robot inclusion/exclusion document[36].

Once the hostname is resolved, the *HTTP fetcher* attempts to download the Web resource. The link instructs the HTTP fetcher which HTTP verb should be used in the request (e.g. a GET method, the more complex POST method). The link should provide the whole body of requests if required, such as a POST request to a Geospatial Web services. The content of the response depends on headers added to the request: information about the content type requested, the charset accepted, the encoding accepted, the language preference, the user agent that performs the request, cookies and other additional headers. The values of the request headers default to values found in the crawling session configuration. These values can be replaced by values provided by the request. The configuration also defines other features, such as the maximum retrievable size, communication timeouts, and the support of mangled HTTP responses and clumsy HTTP servers. The support of mangled HTTP responses and clumsy HTTP servers is a relevant feature because several implementations of Geospatial Web services on HTTP use in-house servers. Additionally, the fetcher should download the body of responses conditioned on certain values of the response headers. For example, the fetcher might be instructed to not continue downloading the response if the header tells that the response is an image. Finally, the Web fetcher should support transparently temporary and permanent redirections and basic authorization schemes.

The response to the request may be a success or a failure, and may contain a payload. In either case, the request and the response details are processed in sequence by a list of registered scripted *processors*. Each processor contains a *guard*, a *content processor* and a *link processor*. The guard checks if the script can process the input. Typically, the guard may use a library for identity the type of content. If the guard returns true, the input is passed to the content processor. Otherwise, the worker thread continues with the next registered script. The *content processor* implements or use a parser defined in a library for parsing the payload and extract links contained therein. The set of links discovered by the content processor may be processed by a *link processor* that is a hook for adding specialized strategies for the generation of next links. Web request, web responses, metadata, content extracted, and so on, may be stored or not in an archive. The responsible of this decision is the script. Links discovered and failed URLs that should be retried, are passed to the *frontier*.

[36] The only well known location of a robot inclusion/exclusion document is `/robots.txt` in the root of the Web domain or subdomain. Each `/robots.txt` is a plain text file contains robot exclusion rules defined by the Robots Exclusion Protocol (REP, Koster, 2010). A non standard practice is including in the `/robots.txt` links to multiple Sitemaps documents (Google, Inc. et al., 2008), a XML file that defines robot inclusion roles. Main search engines's crawlers honour REP and Sitemaps.

Additionally, the frontier can receive events from the scripts, such as the finding of an OGC Web service metadata document. Finally, if the content processor returns true of this is the last script, the thread worker releases resources and then stops. If not, the worker thread continues with the next registered script.

Scripts can modify the behaviour of the crawler at run-time. The best example is a script that updates the inclusion/exclusion rules. If such script detects that the response is a `robots.txt` file compliant with the Robots Exclusion Protocol (Koster, 2010), then the script could extract the `Disallow:` rules, and then, it could reconfigure the *frontier* for excluding matching requests from being crawled.

In the *frontier*, the candidate link first passes through a duplicate link eliminator and a link filter. The duplicate link eliminator, which maintains the collection of visited places with their respective in links and out links, passes on only first visited URL or URL that should be retried. First, the link filter normalises the URL, assigns a unique identifier to the request based on the URL and other features and updates accumulate distance metrics (e.g., number of hops). Then applies several filters for determining if the request is in the scope defined for this session crawl. Then, the hybrid priority policy in use (see Section 2.5.2) is applied to score the link. This score is then mapped to a position for the link in the frontier data structure. Adding a link to the frontier may generate metadata that could be stored in the archive if required. Link filters, links eliminators and priority policies are pluggable components.

If the use of a contingency table by the frontier to predict the coverage is enabled (see Section 2.5.3 for the implications and requirements), the values of the contingency table are updated when the frontier receives an event signalling the discovery of a relevant resource. The contingency table maintains a minimum amount of information of relevant resources found in previous crawls that suffices to identify that if the resource was previously found. The frontier enters in a halting mode if there is a reasonable expectative of no new results, or there is a chance to start to revising previous crawls. In halting mode, the frontier only might supply to the fetcher links that are tagged as sure relevant resources (e.g. a link whose origin is a hyperlink in a HTML page that encoded an explicit *GetCapabilities* request) until the frontier runs out of them.

2.6.2 Extension points

The proposed architecture for an advanced crawler is similar to other extensible crawler architectures, such as Nutch (Cafarella and Cutting, 2004) and Heritrix (Mohr et al., 2004), based on ideas implemented in the seminal Mercator (Heydon and Najork, 1999). The architecture defines a set of core interfaces and extension points that allow extending by plugins the functionality of the crawler. Plugin architectures allow facing the evolving nature of the media formats in the Web, and the idiosyncratic requirements of user communities. For instance, Nutch and Heritrix plugins allow personalizing network protocols, indexing schemas and parsing strategies.

The extension points are useful to add additional functionality to a crawling session. For example, a Web crawler administrator requires adding politeness policies to avoid barrages of request to the same Web server that may be considered by the Web server administrator as a denial-of-service attack. The Web crawler administrator may use the extension point of the frontier data structure to implement this new functionality. The frontier data structure contains an ordered list or queue of request to crawl. There are several data structures based on the idea of *queue* that can be used by the crawler's frontier to keep a list of requests pending: *breadth-first transversal queue*, *priority queue*, *queue with timestamps* and *multiple queues*.

A strategy based only in a unique *breadth-first transversal queue* or FIFO queue is discouraged. A FIFO strategy can generate a barrage of requests to the same server (see Olston and Najork, 2010). A *priority queue* is the next choice if the crawl is focused on OGC Web services. However, there is a chance that a priority queue can also generate a barrage of requests. A better solution is a *queue with timestamps* (Heydon and Najork, 1999), where each request has attached a future time t that forces a temporal order in dequeue operations. That is, dequeue operations only return requests whose time t had been reached. The goal of such a queue is to prevent the above barrage of requests. Timestamps allow establishing policies biased to query more often fast servers. For instance, the request to slow servers can be spaced out if the estimation of time t for a server takes into account the average time that took to download resources from that server (Heydon and Najork, 1999). Then, a possible solution is a priority queue that forces additionally a temporal order in the dequeue operations. A simple implementation is the use of a data structure composed by *multiple queues* (see Section 2.5.2). This structure may discretize a combination of priority and time into levels, each assigned to a FIFO queue (Diligenti et al., 2000) among other possibilities. A frontier with multiple queues should returns links form the non-empty queue with the highest level. The multiple queues strategy may include extra code to avoid *queue starvation*, that is, queues that are never selected because there is always a non-empty queue with higher priority. The code guarantees the selection of a minimum amount of links from queues with low priority even if there are full queues with high priority.

2.6.3 Geospatial extension points

The extension points may add domain related functionality. In the context of OGC Web services, the following extension points can be used for enabling the focused behaviour of the crawler and implementing some of the heuristics identified in Section 2.5:

- **Content parser plugins** are the core of extensible crawler architectures. These plugins, given the output of a request composed of raw content and metadata, first evaluate if can process the output, and then text, links and metadata (author, title, etc.). Extensible crawlers include parsers for well-known media formats such as HTML, PDF, Word, RTF, etc. Indexing the Geospatial Web requires the use of modules able to deal with descriptions of geospatial

Web sources, such as OGC Web service metadata documents and ISO GMD records, and to identify Web pages containing explicit or implicit links to these descriptions. Developing such modules require lot of testing and debugging. Code targeted to parsing concrete hosts can be embedded in the processing scripts to avoid cluttering crawler libraries. The plugins required for OGC Web service specifications can be implemented from scratch, using domain libraries, such as the based in the OGC standard GeoAPI (Reynolds, 2005), or by generating source code with XML processing libraries and tools. Tamayo et al. (2010) provides an evaluation of the different choices for dealing with XML documents specified by OGC XML Schemas.

- **Content type detection**. Content type detection libraries help to detect if a script can process a response. It is often possible to detect or confirm the type of a response payload by looking for special patterns of bytes, also knows as *magic numbers*, at the beginning of the payload. Magic numbers are documented in the Unix platform and its derivatives, and in forensic tools (see Sammes and Jenkinson, 2007). For OGC Web service metadata documents and ISO GMD documents, the standard magic numbers are not enough. The detection of content type requires the precise identification of the standard. This task can be accomplished by extending existing magic number lists with patterns that include the character sequences found in the root nodes of OGC Web standards (e.g. the sequence "`<WMT_MS_Capabilities`" identifies WMS instances), or developing lightweight parsers focused on the identification of the standard.

- **Content mapping**. The example that illustrates the conceptual solution proposed in this thesis (see Figure 2.1) shows that the information about an OGC Web service discovered in a crawling session will be republished later as RDF using an ontology for OGC Web services, which is described in the next chapter. The translation of the downloaded OGC Web service metadata to a RDF model can be done during the crawling session delegating the mapping to a pluggable component. This extension allows archiving the raw payload and the mapped rich metadata side by side in the archive. The details of implementation of the mapping are out of the scope of the thesis. However, given a conceptual mapping between a XML schema and the ontology, the implementation of the transformation does not differ from existing solutions that include the RDFizers[37] of the Simile project at MIT for the general domain, and the Geospatial Semantic Interoperability experiment at OGC (Lieberman, 2006) for the geospatial domain.

- **Domain next URLs**. The different strategies described in Section 2.5.1 for testing the presence of a disconnected OGC Web service or for the deep Web crawling of a CSW are implemented as a library that can be used by the link processor. Given a candidate link, the link processor may return additional *next links* using this library. For example, from

[37] `http://simile.mit.edu/wiki/RDFizers`

the link `http://www.example.com/someserver`, it is possible to derive different *GetCapabilities* requests, such as the WMS request `http://www.example.com/someserver?service=WMS&request=GetCapabilities` and the WFS request `http://www.example.com/someserver?service=WFS&request=GetCapabilities`. If the response being processed is a CSW service metadata document, the link processor may return a link that encodes a request for retrieving metadata records lists from the CSW. Finally, if the response is a record metadata list from a CSW, the links processor may generate next links for retrieving each of the metadata records identified in the list, and a next links that query for the continuation of the record list. The derivation of *GetCapabilities* request is documented in Section 2.4. The harvest of CSW services is well described in the available technical documentation about SDIs (e.g. INSPIRE DT NS, 2009).

- **Domain URL learner**. Aggarwal et al. (2001) introduces the use of URL learners. It's application to WMS-focused crawlers to prioritize candidate links was documented by Li et al. (2010). A URL learner focused on OGC Web services starts with a basic knowledge base composed of query parameters defined in OGC specifications (e.g. "service", "request", Whiteside (2007)) plus well-known naming patterns of service implementers (e.g. ESRI ArcIMS WMS connector (ESRI, 2003)). The parameters used to score URLs may be updated at run-time by analysing the structure of the URLs visited, taking into consideration if the URL points or not to an OGC Web service. URL learners are the first step towards a holistic approach in the design of a crawling order policy.

2.7 Application

The architecture described in the previous Section is the blueprint of a multithreaded geospatial crawler prototype. This Section presents two examples of application that equal or improve the results of the crawlers described in the literature (see Section 2.4). This section is organized as follows. First, the prototype and its technical details are presented. Next, Section 2.7.2 describes how performs this prototype in the discovery of OGC Web services in Europe. This work has been published in Lopez-Pellicer et al. (2010a). Finally, Section 2.7.3 shows that the geospatial crawler can help to measure how the commercial search engines cover the Geospatial Web, and then, select the search engine most appropriate for the discovery of OGC Web services. This work will appear in Lopez-Pellicer et al. (2011).

2.7.1 Prototype

The prototype has been developed from scratch in Java (Gosling et al., 2005) and Groovy (Koenig et al., 2007) in the IAAA Laboratory. Groovy is a scripting language for the Java Virtual Machine seamlessly integrable with Java but with additional features such as closures and native syntax for lists, maps and mark-up languages. The scripted processors that implement the logic of the session crawl and the configuration of the session crawl are implemented as Groovy scripts. The content type detection and processing of documents not related with OGC Web services is delegated to open source libraries, such as Apache Tika[38] and NekoHTML[39].

The points of extension that deal with OGC Web service content described in Section 2.6.3 are implemented in Java and Groovy. The OGC Web service content parser is implemented in Groovy. The parser uses a XML pull parser able to deal with arbitrary large OGC Web service documents with a small memory footprint. The OGC Web service content type detection is implemented in Java as an extension of the Apache Tika library. The content mapping to RDF models is implemented in Groovy and uses the ontology for OGC Web services described in the next chapter. The domain next URL implements heuristics to derive when required *GetCapabilities* for the discovery of disconnected OGC Web services, and *GetRecords* and *GetRecordById* requests for harvesting CSW. Finally, the domain URL learner implements a URL learner based in ideas presented in Li et al. (2010).

The prototype uses the WARC (Web ARChive) file format for the persistence of the content downloaded, and the SPARAQL database TDB[40] to archive RDF models generated during the crawl. TDB is s non-transactional file-based RDF storage optimized for large-scale models. TDB is a component of Jena, a framework for building in Semantic Web applications. The Jena framework provides a RDF API for reading and writing RDF in several format, and OWL API, in-memory and persistent storage of RDF models and a SPARQL query engine. TDB provides for large-scale storage and query of RDF dataset. Jena and TDB are open source projects initially developed in the HP Labs. The standard ISO 28500:2009 WARC file format specifies a method for combining multiple digital resources into an aggregate archival file together with related information. The WARC format is a revision of the Internet Archive ARC file format that has traditionally been used to store Web crawls. The WARC format standardizes and generalizes the older format, and accommodates the use of URIs to relate entries in a WARC file with metadata stored elsewhere. Additionally, besides the primary content currently recorded, the WARC format accommodates related secondary content, such as assigned metadata and transformations to RDF models.

2.7.2 Discovery of services

The experiment uses the geospatial focused crawler prototype for the discovery of OGC Web services in Europe. The crawler was configured for using as seed search results from commercial search engines, and then crawling the Web within three hops of distance from the search engine results. The crawl is restricted to the discovery of WMS, WCS, WFS and CSW servers. The commercial search engines queried are Bing (formerly Microsoft Live Search), Google and Yahoo!. The queries

[38] `http://tika.apache.org/`

[39] `http://nekohtml.sourceforge.net/`

[40] `http://openjena.org/TDB/`

were made through their respective free search APIs, Bing API [41], Google AJAX search API [42], and Yahoo! Search Web Services [43]. In addition, the crawler tests if the services were active in the moment of the crawl. The term *active* in this experiment means that the OGC Web service metadata ascertains which layers, coverages, feature types and metadata schemas are available in the service.

The logic of the experiment has been implemented in the geospatial crawler prototype as process scripts. The identification of search engine results, the parsing of the search results, the request of the next search responses, and the geolocalization of OGC Web services have required the development of additional code. The developed code was made available to the process scripts of the experiment as extension points for content detectors, content parsers, and domain next URLs.

The experiment was done during April and May of 2010 and found 6,544 services in Europe. The data for all identified OGC Web services is aggregated as a categorical data format, that is, the frequencies of the same pattern are grouped (Table 2.7). Each row aggregates the number of ascertained services present or absent from a source. The columns *Google*, *Yahoo!* and *Bing* refer to the OGC Web services present in the seeds. The column *other* identifies the Web services found by the focused crawler applying several heuristics. $Z_{s_1 \dots s_n}$ denotes the total of Web services in each row. For example, Z_{0001} is the number of OGC Web services in Europe not found in search engines, Z_{1010} is the number of OGC Web services only found in the search engines Google and Bing, and Z_{1110} is the number of OGC Web services only found simultaneously in all search engines. Figure 2.7 describes how a service is assigned to the row that it pertains.

This data is used as input for estimate the scale of OGC Web services in Europe applying the log-linear capture-recapture methods presented in Section 2.5.3. The results are presented in Table 2.8. As baseline the Table 2.8 includes estimation of size using the Petersen estimator, which is often found in the literature about the size of Web resources (Lawrence and Giles, 1998; Bergman, 2001). The Petersen estimate can underestimate or overestimate the results when the sources are dependent, that is, when a service is found in a source, this event affects the probability of being found in other source. Table 2.8 contains an estimate of population ($\hat{N}$), measures of goodness of fit for log-linear methods (GF), standard errors (SE), and a 95% confidence interval (CI 95%).

The experiment evaluates two log linear models: the first estimates the population range under the hypothesis of independence of each source, and the later estimates the population range under the hypothesis of local dependence between search engines. The dependent log-linear model, which considers Google, Yahoo! and Bing locally dependent each other, fits the observed data well and yields an estimate of 6,717 services. The independent log-linear mode, which considers independent the sources of services, has a low estimated standard error but it behaves worse than the dependent model if the goodness of fit is considered. The measure of goodness of fit of the dependent model is

[41] http://www.bing.com/toolbox/developers/

[42] http://code.google.com/apis/ajaxsearch/

[43] http://developer.yahoo.com/search/

Table 2.7: OGC Web service data for each search engine in Europe.

Google	Yahoo!	Bing	Other	Data	
			X	$Z_{0001} =$	2,918
		X		$Z_{0010} =$	144
		X	X	$Z_{0011} =$	323
	X			$Z_{0100} =$	33
	X		X	$Z_{0101} =$	140
	X	X		$Z_{0110} =$	3
	X	X	X	$Z_{0111} =$	344
X				$Z_{1000} =$	6
X			X	$Z_{1001} =$	56
X		X		$Z_{1010} =$	10
X		X	X	$Z_{1011} =$	16
X	X			$Z_{1100} =$	2
X	X		X	$Z_{1101} =$	1,135
X	X	X		$Z_{1110} =$	5
X	X	X	X	$Z_{1111} =$	1,409
				Total =	6,544

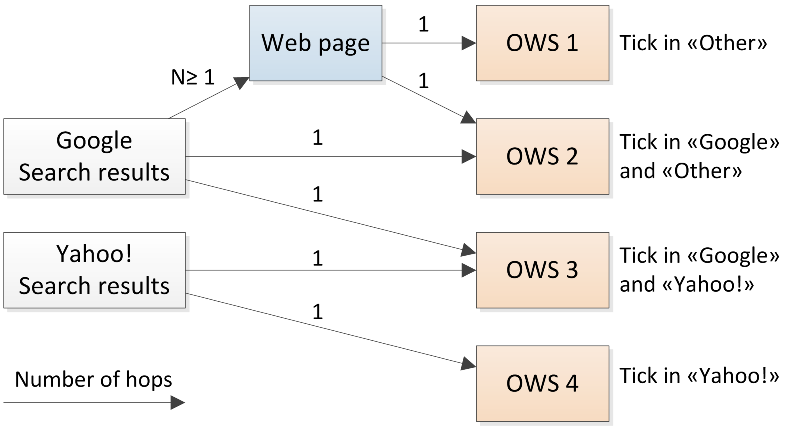


Figure 2.7: Example of list assignment procedure.

Table 2.8: Estimation of the size of public OGC Web services in Europe.

Approach	**Model**	$\hat{N}$	**GF**	**SE**	**CI 95%**
Pair of samples	Petersen (best SE)	3,177	-	5	3,170 – 3,189
	Petersen (worst SE)	4,131	-	44	4,049 – 4,222
Log-Linear	Independent search engines	6,589	7,412.45	8	6,584 – 6,617
	Dependent search engines	6,717	614.83	19	6,684 – 6,757

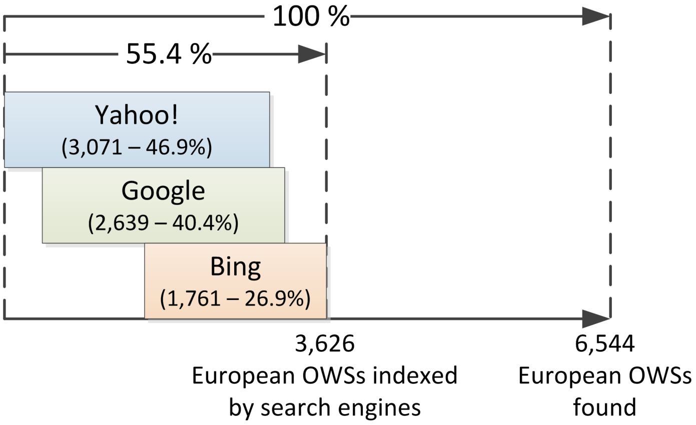


Figure 2.8: Coverage of OGC Web services by search engines in Europe.

614.83, meanwhile the goodness of fit of the independent model is 7,412.45. The log-linear model with dependence is the most adequate to estimate the population. It fits better the set of observations. That is, dependence between the search engines can be detected in the set of observations.

Yahoo!, Google and Bing, combined index a 55.4% of the public OGC Web services found. Their respective coverage is 49.6%, 40.4$ and 26.9% (see Figure 2.8). The percentage of services found simultaneously in the three search engines is 21.6%, which is far from the combined coverage. Considering pairs of search engines, Google and Yahoo! overlap an 80.7% of their results, Google and Bing a 53.5%, and Yahoo! and Bing a 57.3%. These results suggest that if a Web service cannot be found in a general-purpose search engine, it is likely that the same Web service cannot be found in other general-purpose search service. The three main search engines covers only a half of the estimate amount of OGC Web services in Europe. That is, search engines consider a half of the OGC Web services found by the crawler as invisible Web resources. This result supports the statement about the invisibility of the OGC infrastructure discussed in Section 2.3.2

Table 2.9 shows only active services found in Europe. In other words, Table 2.9 shows only OGC Web services that ascertain that they give access to geospatial data and metadata. However, the

Table 2.9: OGC Web services that ascertain their content found in Europe.

	State	**WMS**	**WFS**	**WCS**	**CSW**
EU Member State	Austria	9	0	1	0
	Belgium	58	16	0	1
	Bulgaria	0	0	0	0
	Cyprus	0	0	0	0
	Czech Republic	211	10	1	0
	Denmark	32	9	1	0
	Estonia	3	0	0	0
	Finland	85	2	1	1
	France	137	59	6	4
	Germany	910	51	4	7
	Greece	4	0	0	0
	Hungary	0	0	0	0
	Ireland	11	7	0	1
	Italy	281	184	41	4
	Latvia	8	1	0	0
	Lithuania	0	0	0	1
	Luxembourg	3	2	0	0
	Malta	0	0	0	0
	Netherlands	81	19	13	5
	Poland	24	4	0	0
	Portugal	35	16	3	0
	Romania	3	0	2	0
	Slovakia	5	0	1	2
	Slovenia	2	0	0	0
	Spain	971	303	8	7
	Sweden	26	3	0	0
	United Kingdom	99	8	82	0
EU Candidate	Croatia	3	0	0	0
	Macedonia	0	0	0	0
	Turkey	0	0	0	0
EFTA Country	Iceland	0	0	0	0
	Liechtenstein	1	0	0	0
	Norway	148	20	2	0
	Switzerland	53	18	8	2
Geolocalized in Europe	502	3	0	0	0
Total	3,705	735	174	35	17

geospatial portals of some countries, for example Hungary[44], could not be parsed with the current logic implemented in the processing scripts. Table 2.9 also shows that the implementation of services varies from countries, such as Germany and Spain, with more than 900 public services, to other apparently without public OGC Web services, such as Bulgaria. The reasons behind this apparent divide may lie on political factors (the policies on geographic data publication), geographic factors (the extent of the country), economical factors (the level of economic development), and technological factors (the implementation of geographic information systems in the public administration).

The results might present location bias. Search engines could rank results taking into account from where the query has been made, and answer with indexes available in the nearest datacentre. As the crawler was located in Spain, it is possible that there are more odds that query answers contain an OGC Web service in Spain rather than in other countries. The location bias could be avoided by deploying the crawler in hosts distributed in several countries. The localization of services is based on the country top-level domain, and if the address is an IP or the top-level domain is generic, the IP is georeferenced. The accuracy of georeferenced IP varies. For example, there are 505 services whose IP only can be georeferenced with a continental accuracy. The accuracy of the localization could be increased if the description of the OGC Web service provider found in the capabilities document is taken into account to assign the country.

The experiment shows the application of the geospatial crawler for OGC Web services restricted to a small number of OGC specifications in Europe. The results are similar or better to other geospatial crawlers identified in Section 2.4. The experiment also shows that it is possible to provide data for regional analysis and estimate the number of public OGC Web services in an area.

2.7.3 Selection of search engines

The huge success of commercial search engines, and the growing relevance of Geospatial information in search engines, especially since the successful release in 2005 of Google Maps, has triggered a paradigm shift in the Geospatial community. This community has started to move gradually from the SDI paradigm to approaches based on the best practices of the Web 2.0 (Turner, 2006; Goodchild, 2007). The development and use of Geospatial resources outside SDIs also questions the role of catalogue applications and catalogue services as main discovery tools in SDIs. There literature presented in Section 2.4 shows the ability of search engines as a replacement to traditional catalogue systems for the discovery of geographic Web services. However, these findings are questionable as seem to be based only on anecdotal evidences. In other words, the literature discloses the amount of services found but not discloses the performance of search engines in the task. In addition, the available literature has a strong bias towards Google.

The geospatial crawler prototype has been configured for collecting data to measure the performance of search engines for the discovery of OGC Web services by means of three main commercial

[44] e.g. http://geo-portal.hu/geo/

search engines, Bing, Google and Yahoo!. The performance is analysed through their free search APIs. With the required caution when generalizing the results obtained from the search APIs, the collected data could help to address the questions not answered yet by the available literature including, like what is the performance of the search engines, which is the best-suited search engine and which is the best discovery strategy.

Traditionally, the evaluation of the precision of search engine results is manual (e.g. Brophy and Bawden, 2005; Lewandowski, 2010). As the work of Shang and Li (2002) points, the manual evaluation is accurate but also subjective and time-consuming. Manual evaluation might provide promising findings, but they become out-dated when the technology of commercial search engines changes. Literature offers several examples of automatic evaluation of search engine results, such as Chowdhury and Soboroff (2002), Shang and Li (2002) and Can et al. (2004).

The study evaluates two discovery strategies:

- **Basic strategy**. This strategy queries for tokens that may appear in the query fragment of a *GetCapabilities* request to an OGC Web service plus additional terms related to the task (e.g. "tsunami getcapabilities"). The rationale behind this search strategy is that search engines can index service descriptions because geoportals often publish documents with hyperlinks to them (see Maguire and Longley, 2005).

- **Expert strategy**. This strategy refines the basic strategy by restricting the query to documents where the tokens must appear in the URL of the document (e.g. "tsunami inurl:getcapabilities"). This strategy is well documented in the literature (Bartley, 2005; Refractions Research, 2006; Sample et al., 2006).

There are three search goals evaluable in the above scenario:

- **List candidate services**. The user expects a list of documents about OGC Web services. The documents can be service descriptions or documents with links to service descriptions. For instance, an environmentalist can do an exploratory search of pollution maps and pages related.

- **Interact with services**. The user needs to interact with relevant OGC Web services found in the search response for achieving some objective. For example, an emergency manager might search for a WMS service offering real-time images of a flood for the immediate use of the service.

- **Discover services**. The user collects a ranked set of candidate Web services for an unspecified purpose. These Web services can be found in the search response, or navigating the links of the pages found in the search response. For instance, an analyst might collect OGC Web services from a search engine for building a thematic collection of services about urban planning.

The workflow required by the study has been implemented in the geospatial crawler prototype as process scripts. The identification of search engine results, the parsing of the search results, the request of the next search responses, and the aggregation of data have required the development of additional code. The developed code was made available to the process scripts of the experiment as extension points for content detectors, content parsers and domain next URLs.

The data was collected from automated queries made to Bing, Google and Yahoo! between 29 and 30 July 2010. The study was performed weeks before Yahoo! officially transitioned its search backend to Microsoft search platform in the US and Canada on August 25, 2010. As the evaluation is automated, the study could be repeated to compare the effect of the search agreement in the performance of Yahoo!.

The search engines were queried with two query sets of 1000 queries. The first query set was named *basic*, and it represents the basic strategy described above. The second query set was named *expert*, and it represents the expert strategy described above. Only the first 50 results for each query were collected.

Each query of the basic query set contains two terms: *getcapabilities* that represents the intent of a user to obtain OGC Web services, and a term from a domain vocabulary that represents a topic constraint. The term *getcapabilities* is a mandatory value that appears in the URL of a HTTP GET *GetCapabilites* request.

The queries of the expert query set contain the same 1000 queries but the term *getcapabilites* is prefixed. When the target of the queries is the search API of Google or Yahoo!, the term *getcapabilities* is prefixed by the operator *inurl*. When the operator *inurl* is included in a query, Bing and Yahoo! restrict the results to documents containing the term getcapabilities in its URL. When the target is Bing, the operator *inanchor* is used as prefix. Bing does not offer an *inurl* operator currently. When the operator *inanchor* is included in a query, Bing restricts the results to documents that the text of the anchor link from other page that points to them contains the term *getcapabilities*. *Inanchor* is the Bing operator with closer semantics to Google and Yahoo! *inurl* because geoportals often use the URL of a service description as the text of the anchor link that point to the service.

The domain vocabulary is derived from the OGC service descriptions discovered in a previous experiment described in Section 2.7.2. Each XML document in the collection was parsed for extracting its title and subjects. The XML tags that contain this information are not the same across the OGC Web service standards family. Hence, the first step was the identification of the OGC Web service interface specification. Then, the standard tags that identify title and subjects were located. The text content was converted into a bag of words. The tokenization rules applied were similar to those that search engines use: transformation to lowercase, spaces and punctuation as word separators, and exclusion of words with encoding errors. Digits, one and two letter words, and words commonly ignored by search engines are also excluded but those with semantic relevance in the geographic domain. From the resulting vocabulary of 6553 words, a random selection of 1000 terms was used for building the query sets.

Each search result is evaluated for determining its relevance in different search goals. The evaluation retrieves the search result, and optionally linked documents. Then, the evaluation analyses technical characteristics of the retrieved resources that a human judgement could have considered as necessary. Finally, computes a relevance score. All these procedures are implemented as part of the process scripts used by the geospatial crawler prototype. The relevance of a result and its relevance score is computed differently for each search goal:

- **List candidate services**. A search result is relevant or true positive with relevance score 1 when it is a plausible candidate for helping the user to achieve some unspecified goal related with geographic Web services. That is, the search result is an OGC Web service metadata document or a Web site that include out links to several OGC Web service metadata documents. Otherwise, it is considered a miss of false positive. Its precision is computed as usual (see Lewandowski, 2007).

- **Interact with services**. A search result is a true positive with relevance score 1 if it is an OGC Web service metadata document. Otherwise, it is considered false positive. Its precision is computed as usual.

- **Discover services**. A professional search should measure the utility of a search engine taking into account the number of relevant resources reachable after a search. This number includes the documents found by navigating from the result pages. The relevance of a resource should be proportional to the number of out links to new relevant resources. Then, a search result should be a true positive if it is an OGC Web service metadata document or a Web site with links to new OGC Web service metadata documents. If all the OGC Web service metadata documents linked can be also found in results with higher rank, the search result is considered a false positive. The relevance score is proportional to the number of new OGC Web service metadata documents found.

The results include an analysis of the overall results, the unique results across all queries, the cross-coverage of search engines, and the precision. Absolute precision and pseudo-precision values cannot be computed because the number of relevant documents in relation to each query is unknown, and the search engines limit the size of the search response. The estimates of precision are obtained using the micro averaging and macro averaging methods (see Sebastiani, 2002). Micro averaging precision is estimated summing over all results. Macro averaging precision requires first estimate the precision of each query, and only accounts queries with non-empty results. Following is a summary of the results:

- **Overall results.** Table 2.10 contains a summary of the overall results of the two strategies. As expected, overall results shows that it is feasible find OGC Web services or pages linking to those services in the evaluated search engines. There is sufficient statistical evidence to suggest that the overall proportions of relevant pages, resources and noise are not equal in the analysed search engines for both query sets (basic: χ^2_{obs}= 2066.90, $\wp$-value$\approx$ 0, and expert: χ^2_{obs}= 333.86, $\wp$-value $\approx$0). The percentage of OGC Web services found using an expert strategy ranges from 95.9% in Google (*inurl* operator) to 78.7% in Bing (*inanchor* operator). If the goal of our search is to interact with services, an expert query provides more direct relevant hits than a basic query. Nevertheless, if the services that can be found following the hyperlinks of the returned pages are considered, the best approach is a basic query. A little of Web mining makes the services discovered soar. For example, the mean average of discovered services per query in Yahoo! is 27.4 using simple queries compared with 5.3 obtained with expert queries. No wonder which factor used, Yahoo! outperforms Bing and Google in the number of results and relevant results returned in both strategies.

- **Unique results**. Table 2.11 shows the overview of the unique results across all queries. The statistical evidence suggests that the proportion of unique relevant pages, resources and noise continue to be different in each search engine (basic: χ^2_{obs}= 676.44, $\wp$-value$\approx$ 0, and expert: χ^2_{obs}= 26.46, $\wp$-value$\approx$ 0). The table includes the ratio between overall and unique results for the same category in each search engine (o/u). The analysis of the o/u ratio concludes that the same result might be found from 1.8 to 5.2 times across the responses of the same query set. The value of the o/u ratio for the "expert" query set is always greater than the "basic" query, notably for Google and Bing. As queries were made in sequence, it is possible that Google and Bing are prone to cache answers to queries with operators for their reuse. This search engine optimization is known as *query locality* (Baeza-Yates et al., 2007). A search engine could use a cache memory to speedup query computation by exploiting frequently terms or recently used answers. That is, the search engine can return search results from a local cache rather than from its main index. Caching relies in the supposition that users only consider the first page of results, and that there exists sufficient repetition in a stream of requests (or answers) during a time window. Query locality alone does not explain why the presence of *inurl* and *inanchor* doubles respectively the o/u ratio in Google and Bing and not in Yahoo!. This fact suggests that caching policies of Bing and Google are more aggressive when a query includes operators. Hence, a search strategy focused to the discovery of new OGC Web services with operators with successive queries is discouraged in Google and Bing. Yahoo! presents again the best absolute results. It returns for the basic strategy an amount similar to the sum of Bing and Google results, and outperforms without discussion when the query strategy is the expert one.

- **Precision**. Figure 2.9 and Figure 2.10 present micro and macro average precisions for each combination of query set and search goal for each search engine. Google is the most precise

Table 2.10: Number of results and discovered resources in 1000 queries. This table only considers the top-50 results. The operator employed in the expert query is between parentheses in the corresponding column.

Basic query	**Bing**		**Google**		**Yahoo!**	
Pages w. links to srv.	13952	70.7%	2383	51.5%	11986	51.5%
Service metadata	631	3.2%	418	9.0%	2551	11.0%
Noise	5154	26.1%	1827	39.5%	8722	37.5%
Total results	19737	100.0%	4628	100.0%	23259	100.0%
Discovered services	23688		5679		27408	
Expert query	**Bing (inanchor)**		**Google (inurl)**		**Yahoo! (inurl)**	
Pages w. links to srv.	211	14.2%	97	3.5%	614	10.3%
Service metadata	1173	78.7%	2695	95.9%	5162	86.9%
Noise	107	7.2%	19	0.7%	162	2.7%
Total results	1491	100.0%	2811	100.0%	5938	100.0%
Discovered services	1173		2699		5359	

Table 2.11: Number of unique results and discovered resources in 1000 queries. This table only considers the top-50 results. The operator employed in the expert query is between parentheses in the corresponding column

Basic query	**Bing**		**o/u**	**Google**		**o/u**	**Yahoo!**		**o/u**
Pages w. links to srv.	4214	70.1%	3.3	1372	53.0%	1.7	5226	55.3%	2.3
Service metadata	221	3.7%	2.9	214	8.3%	1.9	1347	14.2%	1.9
Noise	1576	26.2%	3.3	1002	38.7%	1.8	2882	30.5%	3
Total results	6011	100.0%	3.3	2588	100.0%	1.8	9455	100.0%	2.5
Discovered services	3272			2055			5036		
Expert query	**Bing (inanchor)**		**o/u**	**Google (inurl)**		**o/u**	**Yahoo! (inurl)**		**o/u**
Pages w. links to srv.	26	9.1%	8.1	15	2.7%	6.5	191	8.7%	3.2
Service metadata	257	89.6%	4.6	537	96.9%	5	1987	90.4%	2.6
Noise	4	1.3%	26.7	2	0.4%	9.5	21	0.9%	7.7
Total results	287	100.0%	5.2	554	100.0%	5.1	2199	100.0%	2.7
Discovered services	257			538			2015		

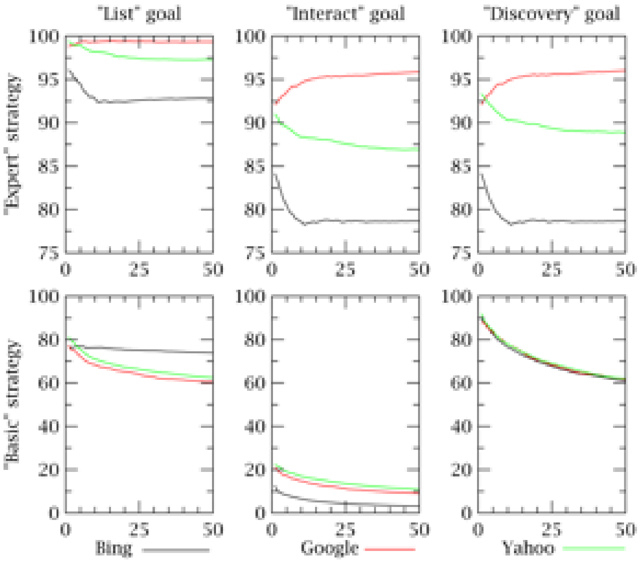


Figure 2.9: Micro average precision for each combination of query set and goal. First and second columns are standard precisions; third column is the pseudo-precision.

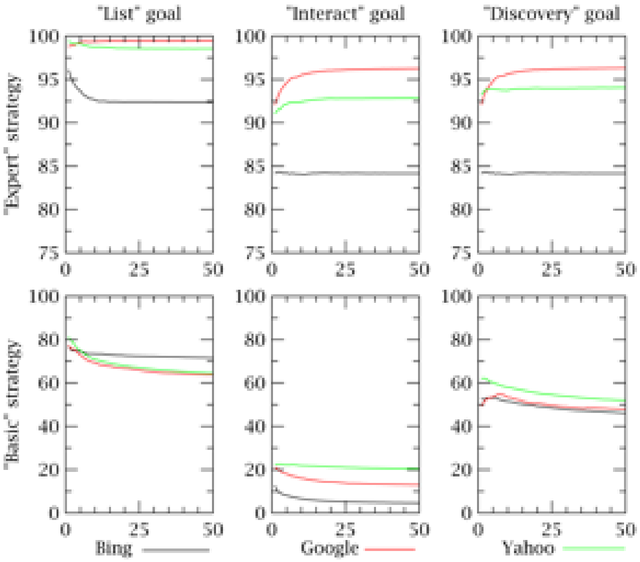


Figure 2.10: Macro average precision for each combination of query set and goal. First and second columns are standard precisions; third column is the pseudo-precision.

Table 2.12: Cross-coverage of relevant unique results for different search goals. This table only considers the top-50 results.

Found in			**Basic query**			**Expert query**		
Bing	**Google**	**Yahoo!**	**List**	**Inter.**	**Disc.**	**List**	**Inter.**	**Disc.**
		X	4825	1202	2240	1890	1709	1736
	X		878	143	354	357	350	351
	X	X	213	52	239	150	143	143
X			2739	121	597	130	107	106
X		X	1201	81	1213	108	106	107
X	X		161	7	118	15	15	15
X	X	X	334	12	1344	30	29	29
Total			10351	1618	6105	2680	2459	2487
Indexed only in one			8442	1466	3191	2377	2166	2193
Indexed only by two			1575	140	1570	273	264	265
Indexed by all			334	12	1.344	30	29	29

service for *interact* searches using operators. For instance, the micro average precision of top-10 results is 94.6% compared with 88.4% in Yahoo! and 78.6% in Bing. The precision values computed for the discovery goal are quite similar to the interactive goal for expert queries. As the query is focused on XML document, there is little chance of finding a page with hyperlinks to an OGC Web service. As a rule of thumb, Google is provides the most precise results and Bing is usually the worst performer. In basic searches, the precision of the responses drops in all the search engines. For example, the precision of Bing falls to a value of 6.6%. Yahoo! behaves slightly well than Google for this kind of queries. If the goals *list* and *interact* are compared, the goal *list* is more precise than the goal *interact* as expected. The shape of the pseudo-precision curve is identical for the three search engines and seems that the goal *discovery* is more precise than the goal *list*. Search engines seem to return first documents more linked to XML descriptions, and, as the precision only takes into account new services found, it falls when a search engine starts to return mainly pages.

- **Cross-coverage**. The cross-coverage of unique relevant results gives an approximate idea of the opportunity costs related to the exclusive use of a single search engine as provider of results. Table 2.12 shows the cross-coverage of unique relevant results per search engine. Using a basic strategy, 81.6% of relevant results are found only in one search engine. If the interactive goal is considered, that is, only OGC Web service metadata documents are relevant, the value rises

Table 2.13: Performance of search engines for each combination of search strategy and search goal. "B", "G", and "Y" refer to Bing, Google and Yahoo! respectively.

Strategy	**Factor**	**List**	**Interact**	**Discover**
Basic	- Outcome	Y > B > G	Y >> B = G	Y >> B > G
	- Precision	B > Y = G	Y > G > B	Y > G = B
Expert	- Outcome	Y >> G > B	Y >> G > B	Y >> G > B
	- Precision	G = Y > B	G > Y > B	G > Y > B

up to 90.6%. However, if the discovery goal is considered, which take into account accessible resources from indexed pages, the value drops to 52.3%. It is a fair assumption that search engines' bots can index these resources. Therefore, 52.3% should be an upper bound of OGC Web service metadata documents indexed by only one search engine. For expert queries, the percentage of relevant results found only in one search engine is around 88% no wonder which evaluation factor is applied. Yahoo! seems to be the search engine that index more OGC Web services or pages that links to OGC Web services not indexed by other search engines. Bing and Yahoo! share a significant amount of relevant results (mainly Web pages) found with basic queries. Nevertheless, the results suggest that the exclusive use of only one search engine has an elevated opportunity cost as a half of relevant results is returned exclusively by one search engine.

Table 2.13 presents a summary of the findings. Search engines are sorted by their performance for each discovery strategy and search goal.

The use of the geospatial crawler prototype helps to demonstrate that using quite simple queries at least 60% of the returned results could be relevant. Also reveals that search engines index more than 6.000 OGC Web services. This value exceeds the number of services listed in sites specialized in OGC Web services, such as *Geopole* and *Mapmatters* (see Section 2.4). Yahoo! seems to have the best performance in the discovery task, followed by Google and Bing. In addition, the study shows that search engine optimizations may harm the use of search engines as an alternative to traditional catalogues. Finally, the study has debunked two assumptions often found in the literature: Google as the best search engine for finding OGC Web services, and the use of operators for finding OGC Web services.

It is fair to question if the data from Web user interfaces and APIs provide the same results. McCown and Nelson (2007) examined this issue and concluded:

> *Researchers may need to use caution when generalizing their results obtained from the API to those results that the common user sees using the Web user interface.*
>
> McCown and Nelson (2007)

In fact, Web user interfaces and APIs have two different market targets. The target of Web user interfaces is users that use as user agent a Web browser for querying the search engine; the target of APIs is user groups that use the same user agent, often a Web application developed for supporting the user group, that queries the search engine on behalf of each user, even autonomously.

2.8 Summary of the Chapter

This chapter has presented and characterized the problems of invisibility of the Geospatial Web, exemplified in the problems of invisibility of OGC-based infrastructures. The invisibility is a problem that can be found in other SOA architectures, such as W3C Web Services. A consequence of the invisibility is that Geospatial contents become part of the so-called deep Web.

The current trend in SOA architectures is to consider the use of specialized Web search engines backed by Web services crawlers to make Web services more discoverable. Given the evolution of general-purpose SOA architectures and its parallelisms to the OGC architecture, it is natural to consider search engines for OGC Web services based in crawlers focused on OGC Web services as the future solution for the discovery of OGC Web services.

A review of the literature shows that the use of crawlers been explored, but the results are far from being the basis to an alternative to traditional discovery procedures based of search engines. The challenges that a crawler focused on OGC Web services faces are related with the development of strategies to discover invisible geospatial content, to reduce the crawling space and assure that all the relevant OGC Web services in a part of the Web have been crawled. The chapter presents the architecture of an advanced and extensible geospatial crawler that acts as blueprint for a prototype. This prototype implements the heuristics for facing the challenges identified, and has a good performance when it is applied for the discovery of OGC Web services, and the analysis of the amount of OGC Web services indexed by search engines.

It is possible to develop a system that tackles the problem of the invisibility of Geospatial Web by a focused and systematic crawl of the Web for Geospatial Web services. In addition, it presents empirical results that show that an advanced geospatial crawler, with the appropriate heuristics and strategies, can crawl the Geospatial Web better than any previous crawler can. Therefore, invisible resources can be discovered. If descriptions of these resources are published in a format indexable by search engines, they will become visible.

Chapter 3

Ontology for OGC Web Services

3.1 Introduction

Standard Web services, such as W3C Web services (Booth et al., 2004) and OGC Web services (Percivall, 2002), give access to the functionality and the content of a system to the rest of the world using a standard distributed computing platform (DCP) often based on the HTTP protocol and messages encoded in XML. A Web service is an application component that represents a self-contained unit of functionality of a system that is able to self-describe and to interact with its environment in a standardized way. The available standards for Web services specify only the *syntactic interoperability* of the interaction with the environment leaving implementers free to choice how the semantics and the implementation details of each interaction are mapped to a Web service. The resulting Web service is independent of the underlying implementation of the interaction but bears its semantics.

This chapter presents the *ontology for OGC Web Services* (OntoOWS). The purpose of this ontology is to describe an OGC Web service using semantic constructs. Figure 3.1 shows the different modules of the example application outlined in the introductory chapter that may use OntoOWS as one of their reference models.

OntoOWS has its roots in the research in Semantic Web services (McIlraith et al., 2001). The vision of Semantic Web services is that Semantic Web technologies combined with Web services could solve integration problems, such as data interoperability, formats heterogeneity and interface matching, that harm not only the development of a global e-market of Web services but also the interconnection point-to-point of systems using standard Web services. In other words, the research on Semantic Web services addresses the *semantic interoperability* of the interaction with the environment.

Semantic Web service researchers have devoted efforts to enrich syntactic specifications of Web services with semantic annotations in order to facilitate the discovery, the composition and the

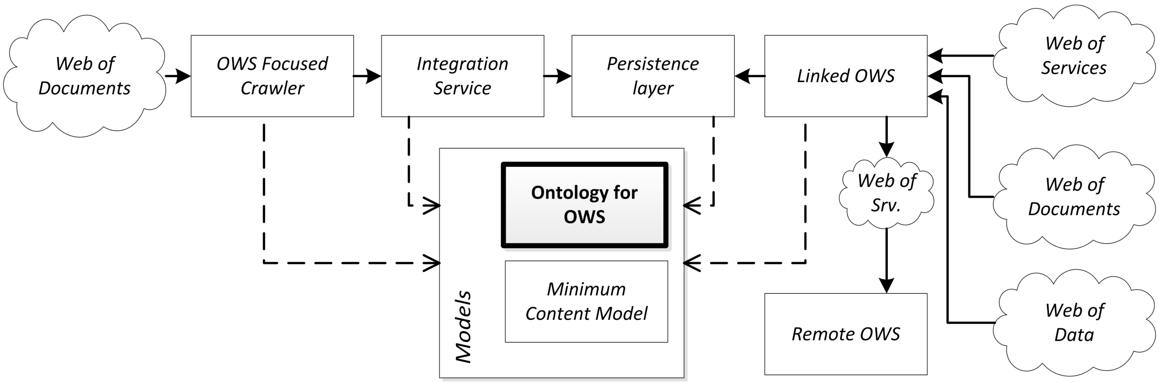


Figure 3.1: The role of the ontology for OGC Web Services (OntoOWS) in the example application outlined in the introductory chapter.

execution of Web services. The seminal work of McIlraith et al. (2001) established clearly the contents of such annotations:

> *The data and metadata associated with a service together with specifications of its properties and capabilities, the interface for its execution, and the prerequisites and consequences of its use.* McIlraith et al. (2001)

W3C Web services use WSDL (Web Services Description Language, Chinnici et al., 2007), an XML language, for documenting everything relative to the syntactic description of a service endpoint: supported operations, message exchange patterns and the definition of the data structure flowing between the clients and the service. If the data structure is based in XML, the definitions of the data structures are made with XML Schema. Other W3C standards provide a syntactic description of the policies in use when accessing a service (WS Policy, Yendluri et al., 2007) and the aggregation of services (WS Choreography, Kavantzas et al., 2005). Several frameworks and annotation languages has been proposed for enhancing these syntactic descriptions with semantics, and for supporting automated discovery of W3C Web services: METEOR-S (Patil et al., 2004), OWL-S (Martin et al., 2004), WSMO/WSML (Roman et al., 2005), WSMO-Lite (Vitvar et al., 2007), POSM (formerly MSM, introduced together with WMSO-Lite), among others. Many of them specify the knowledge representation language used for representing the semantic models. Description logic (Baader et al., 2003) is being used in METEOR-S and OWL-S. The WSMO family of languages uses RDFS (Hayes, 2004), description logic programs (Grosof et al., 2003), description logic, first-order logic (Smullyan, 1995) and logic programming (Lloyd, 1987). The W3C submissions of OWL-S and WSMO do not culminated in their standardisation. Later, W3C recommended SAWSDL (Lausen and Farrell, 2007) that only specifies how to reference arbitrary ontological models from parts of a WSDL document.

OGC Web services use the *service metadata* document (Whiteside, 2007), also know as *capabilities XML* when encoded in XML, for the syntactic description of a geospatial Web service including its policies about fees and rights, and even very specific service orchestrations (e.g. see WMS Cascading in de la Beaujardiere, 2006). Recent OGC Web service specifications can use a WSDL document at the same time as a service metadata document (e.g. WFS 2.0, Vretanos, 2010; WMTS 1.0, Masó et al., 2010). The experience in the semantic description of W3C Web services with semantic annotations has been applied to OGC Web services. For example, OWL-S has been applied to the composition of OGC Web service chains (Yue et al., 2007) and the semantic annotation of geospatial Web services within a semantically enabled catalogue with a CSW interface (Zhao et al., 2009). WSMO/WSML has been utilized for the automatic generation of descriptions of OGC Web services (Klien et al., 2007). OGC has shown interest in use of semantic technologies to the discovery of resources and its translation from a domain to another (Lieberman, 2006) and in the use of semantic annotations based in the SAWSDL approach in OGC standards (Maué, 2009).

Several prototypes (e.g. QUASAR for W3C Web services, see Belhajjame et al., 2008, SWING for OGC Web services, see Roman and Klien, 2007) have shown the advantages of semantic annotations for the discovery of Web services. However, Semantic Web service research has failed to notice the high complexity of the technology, which requires highly skilled experts in Web services and in knowledge engineering. The most evident consequence is that after 10 years of research in Semantic Web services, the industry does not have developed yet a reference killer application or a solution largely deployed based in Semantic Web services. Consequently, there is no significant body of public Semantic Web Services. The largest repositories of annotated Web Services are probably *OPOSSum* (Kuster and Konig-Ries, 2008), a database that integrates collections of Semantic Web Services, and *iServe* (Pedrinaci et al., 2010), a publishing platform for Semantic Web Services. The *OPOSSum* database contains 1525 annotated W3C Web services[1]. *iServe* provides access to 1992 descriptions of Web services, including W3C Web services[2]. To the best of the author knowledge, there is not publicly accessible a database or site with a collection of annotated OGC Web services.

The approach of the ontology proposed in this chapter for describing OGC Web services is similar to the approach described in Pedrinaci et al. (2010) for Web services. The OntoOWS ontology provides a framework that helps to translate in a consistent way OGC Web service metadata encoded in XML into assertions about the service instances, allowing further enrichment. OGC specifications provide for each type of service a rich description of its purpose and the information contents that the user can expect. These descriptions include not only information about the interfaces, the data types and the platform bindings but also expected behaviour, policies, and technological choices.

The ontology for OGC Web Services play a key role in four scenarios in the context of this thesis:

- To persist as machine processable data found about an OGC Web service by a focused crawler (Chapter 2).
- To provide a server with the information required to setup semantic endpoints (Chapter 5)

[1] Services at `http://hnsp.inf-bb.uni-jena.de/opossum/` as December 2010

[2] Services at `http://iserve.kmi.open.ac.uk/browser.html` as December 2010

- To provide a user a rich machine processable description of an OGC Web service (Chapter 5).

This chapter is organized as follows. First, Section 3.2 introduces the methodological approach for the development of the ontology for OGC Web services. Next, Section 3.3 presents the purpose, the scope and the requirements of such ontology. Then, Section 3.4 describes the different aspects represented by the ontology and its conceptualization, formalization and implementation. Section 3.5 provides a glimpse of the use of the ontology to translate the service metadata of an OGC Web service instance into a semantic description. Finally, the main contributions of this chapter are summarized.

3.2 Methodology

In this section, the methodological approach for the specification, the conceptualization, the formalization and the implementation of the ontology is presented. The objective of this section is to describe the activities required for the development of the OntoOWS ontology. This methodology has been also applied to the development of the Geo-Net ontology described in Chapter 4.

3.2.1 Methodological approach

Figure 3.2 provides a description of the methodological approach applied to the development of the OntoOWS ontology. The development can be decomposed in four activities:

- **Specification activity**. The specification activity has as output a rationale of why the ontology is needed and which are its limits. The specification not only describes the scope of the ontology but also the procedures required for the conceptualization and the formalization, the users and the uses of the ontology.

- **Conceptualization activity**. The conceptualization activity structures the domain knowledge into a conceptual model that represents entities or concepts in the domain, and relationships between them. A conceptual model can capture an informal view of the domain without loss of knowledge.

- **Formalization activity**. The formalization activity transforms the conceptual model into a computable model. Some domain knowledge may be lost in the transformation process if the semantic constructs used in the computable model are less expressive than those required by the conceptual model.

- **Implementation activity**. The implementation activity serializes the computable model in a machine processable model using an ontology language or a rule language enabling the use of the ontology.

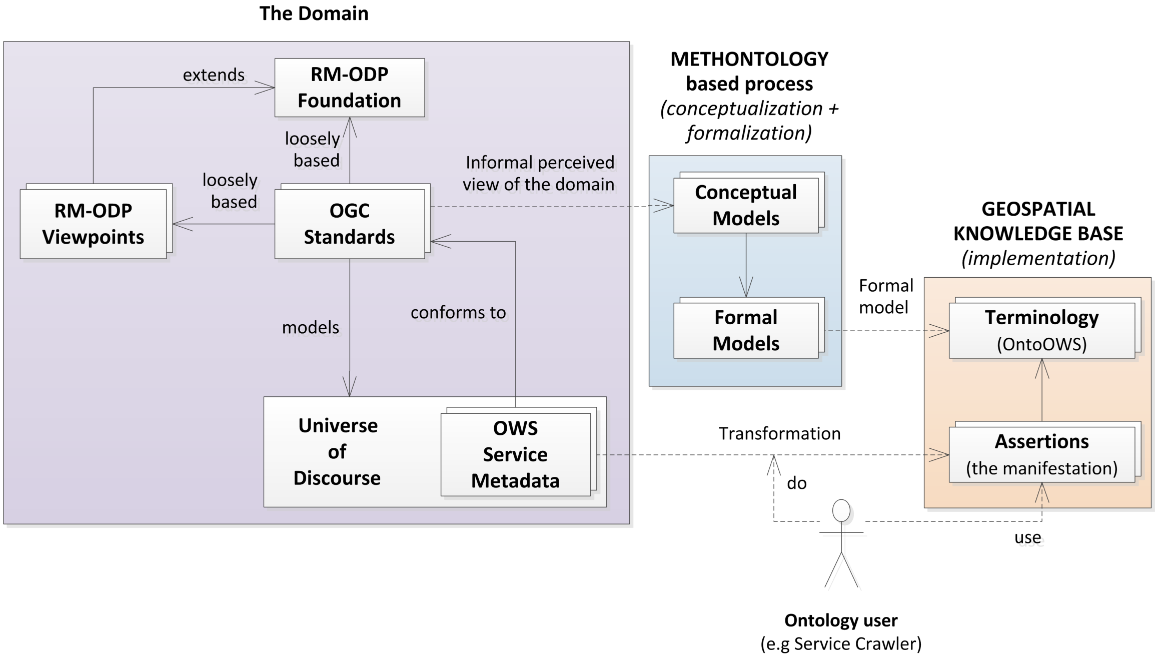


Figure 3.2: Decomposition of the development of OntoOWS in activities.

The *specification activity* is organized around the development of an ontology requirements specification. The *conceptualization* and *formalization activities* follows the ideas of the Methontology framework (Fernández-López et al., 1997) but adapted to the use of recent Semantic Web languages (e.g. OWL 2, OWL WG, 2009), and a well defined *iterative development model* (Larman and Basili, 2003) for ontologies. The *implementation activity* is part of the iterative cycle model.

Methontology is a framework for the development of ontologies based in software development and knowledge development methodologies. Methontology is one of the most popular methodologies for the development of ontologies (used by 13.7% of practitioners according to Cardoso, 2007). For example, Methontology was the methodology for developing and maintaining semantic descriptions of Web services in the project SEEMP which studied the implementation of an e-marketplace to coordinate and integrate public and private employment services around the EU member states (Della Valle et al., 2007). Methontology was also used in the project TAO for defining a gold-standard ontology of the Amazon Web Services[3] with the aim of evaluate the methodology proposed for the transitioning of legacy Web services into Semantic Web Services (Amardeilh et al., 2008). In the project DIP, Methontology was applied for the standardization and integration of financial Web services using WSMO (López-Cobo et al., 2008).

[3] http://aws.amazon.com/

3.2.2 Specification

The *ontology requirements specification* is one of the pre-development activities performed when building ontologies. Gruninger and Fox (1995) states that development of ontologies requires a previous agreement on the purpose and ultimate use. This agreement is materialized in a motivating *scenario*, the set of intended *solutions* to the problems presented in the scenario, and a set of informal *competency questions*, that is, questions that the resulting ontology must be able to answers. For Uschold (1996), the *purpose* is an essential part of the specification. It is worthless to build the ontology without a clear idea of why the ontology is wanted, what it will be used for, and possible mechanisms for its use. Next, the developer should decide the level of *formality* required by the ontology. Finally, the developer should define the *scope*, that is, the set of concepts and terms covering the full range of information that the ontology must characterise. Uschold (1996) proposes two mechanisms for determining the scope: informal *competency questions,* and *brainstorming* when that questions were not available.

Fernández-López et al. (1997) suggests that the specification should state why the ontology is built, what are its intended uses, and who are its end-users. In this framework, the *environment*, that is, the identification of the applications that it supports, is a pre-development activity separated from the specification. Staab et al. (2001) includes the analysis of the environment as part of the specification. The specification should clearly specify the ontology's *goal*, its *domain* and *scope*, the *applications* it supports, its knowledge *sources* (e.g. domain experts, organization charts, business plans, dictionaries, index lists, or database schemas), and its potential *users* and usage *scenarios.* Additionally, the specification should include an overview of possible *competency questions* to the system that clarify the domain and scope of the ontology, and potentially reusable ontologies. Suárez-Figueroa et al. (2009) resumes prior art and proposes a concrete and defined set of tasks for the specification. The application of these tasks must conclude with a specification that defines for an ontology the *purpose*, the *scope*, the *implementation language*, the intended *end-users*, the intended *uses* or scenarios, *non-functional requirements*, *competency questions* (as functional requirements) and a *pre-glossary* of terms derived form the competency questions.

The work plan for the definition of the ontology requirements of the OGC Web services ontology included the following list of tasks, which adapt recommendations from Suárez-Figueroa et al. (2009):

1. **Identify purpose, scope and level of formality**. The objective of this task is to provide statements that motivate the need of the ontology, establish the information covered and the level of detail of the description, and limit the expressive power of axioms and rules.

2. **Identify potential users and usage scenarios**. The objective of this task is the identification of intended uses and end-users in the context of this work, or in a near future, where this ontology could be used.

3. **Identify non-functional requirements**. The objective of this task is the definition of the

aspects not related to the ontology content that the ontology should satisfy. The non-functional requirements can be rewritten in the form "*<ontology> shall be <requirement>*".

4. **Identify functional requirements**. The objective of this task is the specification of content specific requirements. The functional requirements can be rewritten in the form "*<ontology> shall answer <competency question>*".

3.2.3 Iterative conceptualization, formalization and implementation

Given an ontology specification, the Methontology framework is the base for the conceptualization, the formalization and the implementation of an informal view of the universe of discourse of the ontology. The conceptualization, formalization and implementation activities do all the next tasks within an iterative cycle model (see Figure 3.3). These tasks are derived from conceptualization, formalization and implementation tasks identified in the Methontology framework.

1. **Glossary of terms**. A first version of the glossary of terms is built. If the development of the ontology is incremental, the glossary contains only terms related to the requirements addressed in this cycle. The terms are defined in as much detail possible. The output of this task is a tabular glossary of terms relevant in the scope of the ontology. The glossary provides a description of each term, its synonyms, its acronyms and an initial classification as *class*, *datatype, object property*, *data property*, *named instance*, and *constant value*. The types of the initial classification maps to entities of the OWL family of knowledge representation languages. This step is the task "*build glossary of terms*" in Methontology. Table 3.1 contains the definition and the correspondence with Methontology concepts.

2. **Preliminary design**. A preliminary design of the ontology is materialized as a *Concept Dictionary.* This preliminary design involves:

   (a) **The development of class hierarchies**. Methontology proposes the use of four taxonomic relations: *Subclass-Of*, *Disjoint-Decomposition*, *Exhaustive-Decomposition*, and *Partition.* A class C_1 is a *subclass-of* another class C if C_1 is more specific than C. A set of classes $C_1 \ldots C_n$ is a *disjoint-decomposition* of a class C if $C_1 \ldots C_n$ are pairwise disjoint, i.e. no individual can be instance of more than one class, and the union of $C_1 \ldots C_n$ is subclass of C. A set of classes $C_1 \ldots C_n$ is an *exhaustive-decomposition* of a class C, if the union of $C_1 \ldots C_n$ is equivalent to the class C. Finally, a set of classes $C_1 \ldots C_n$ is *partition* of a class C, if it is an exhaustive decomposition of C and $C_1 \ldots C_n$ are pairwise disjoint. Table 3.2 shows the equivalences between these taxonomic relations and the class axioms in description logic. The taxonomy can be represented using a UML class diagram. The UML profile for OWL defined in the *Ontology Definition Metamodel* (2009) may serve as reference for the notation. In a UML class diagram, classes represent concepts, and

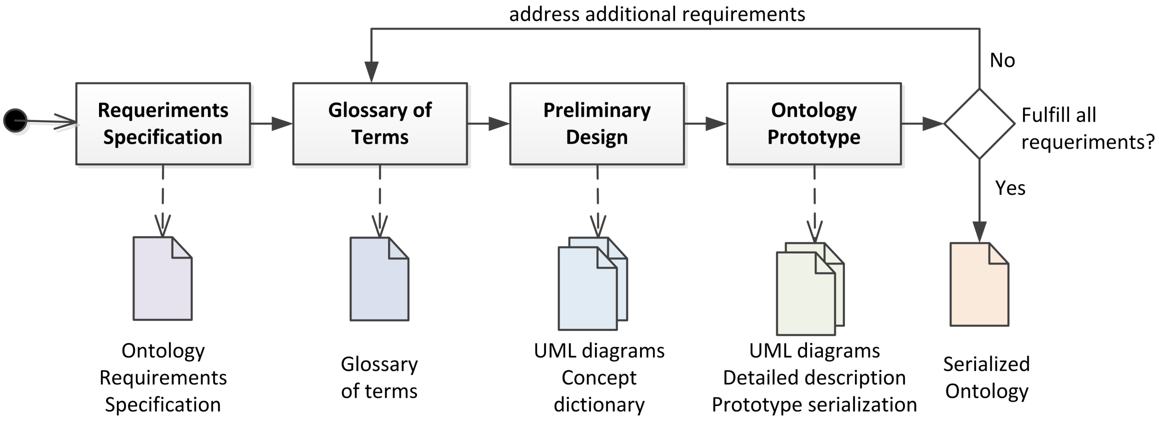

address additional requirements
No
Requeriments Specification
Glossary of Terms
Preliminary Design
Ontology Prototype
Fulfill all requeriments?
Yes
Ontology Requirements Specification
Glossary of terms
UML diagrams Concept dictionary
UML diagrams Detailed description Prototype serialization
Serialized Ontology

Figure 3.3: Iterative and incremental cycle model for the development of ontologies.

Table 3.1: Equivalence between OWL 2 Entities and Methontology concepts.

OWL 2 Entity	Methontology	Represents
class	*concept*	A collection of individuals
datatype	*value type*	A collection of data values
object property	*binary relation*	A binary relation between two concepts
data property	*instance attributes*	A binary relation between a concept and a value type
named instance	*instance*	An identifiable instance
literal (as *constant value*)	*constants*	A constant value

generalization relationships represent taxonomical relations. The specification of a relation as a disjoint-decomposition, exhaustive-decomposition or partition is modelled by adding a constraint to the generalization relationship (see Figure 3.4). This step is the task "*build concept taxonomies*" in Methontology.

(b) **The identification of relevant object properties**. This identification can be performed using a UML class diagram (see Figure 3.5). The goal of these diagrams is to ascertain relevant relationships between classes. These diagrams help developers to figure out domain, ranges and restrictions of object properties. The taxonomy UML diagrams can be reused in this task. At this stage of development, these diagrams only reflect the existence and some properties of the relations. The relations are modelled as association between concepts and the relations and the properties as comments to the relations. This step is named "*build ad hoc binary relation diagrams*" in Methontology.

(c) **The generation of a concept dictionary**. The concept dictionary specifies which are the classes, properties and instances in the domain of the ontology. The description of

Table 3.2: Equivalence between Description Logics axioms and Methontology taxonomic relations.

Methontology	**Class axioms**
Subclass-Of C	$C_1 \sqsubseteq C$
Disjoint-Decomposition of C	$C_1 \sqcup \ldots \sqcup C_n \sqsubseteq C$ $C_i \sqsubseteq \neg C_j,\ 1 \leq i < j \leq n$
Exhaustive-Decomposition of C	$C_1 \sqcup \ldots \sqcup C_n \equiv C$
Partition of C	$C_1 \sqcup \ldots \sqcup C_n \equiv C$ $C_i \sqsubseteq \neg C_j,\ 1 \leq i < j \leq n$

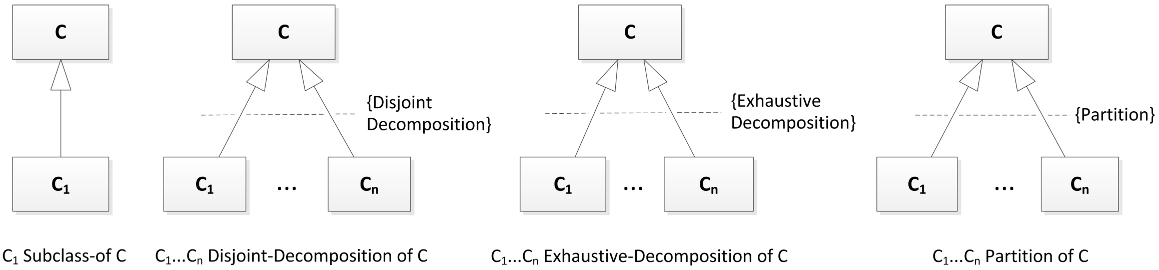


Figure 3.4: Representation of taxonomical relations in a UML class diagram.

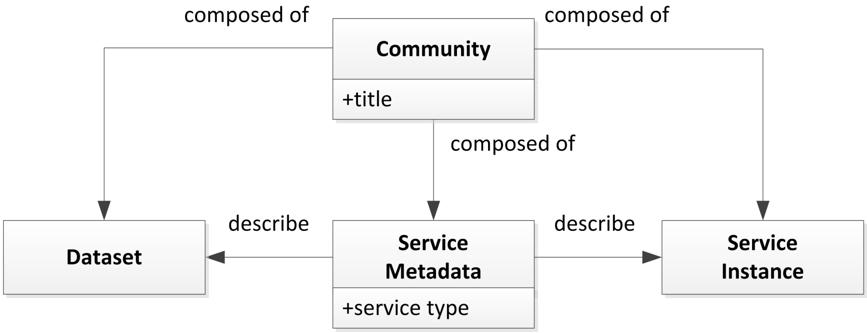


Figure 3.5: Representation of object properties in a UML class diagram.

Table 3.3: Example of tabular Concept Dictionary.

Class	***Data properties***	***Object properties***
Service metadata	service type	describes
Community	title	composed of

the concept dictionary may take the form of a tabular representation (Table 3.3). This step corresponds with "*build concept dictionary*" in Methontology.

   (d) **Identify all the possible problems the ontology draft**. In this phase, the concept dictionary is validated against the requirements of the ontology. The main goal is the identification of any kind of uncertainty in the requirements and alternative ways to represent the domain knowledge. If the validation reveals problems in the specification of the ontology, the specification should be revised and then the process of conceptualization should start again. If the concept dictionary fails to validate against the requirements, a new preliminary design of the ontology should be created.

3. **Ontology prototype**. A first prototype of the ontology is created from the *Concept Dictionary.* The prototype represents an approximation of the characteristics of the final ontology. The development of this first prototype requires.

   (a) **The definition of the properties in detail**. For each object property identified in the concept dictionary, it is specified at least the range, the domain, the cardinality, the inverse relation and some mathematical properties (e.g. functional, inverse functional, reflexive, irreflexive, symmetric, asymmetric, transitive). For each data property identified in the concept dictionary, it is specified its range, its data type, its cardinality and mathematical properties (e.g. functional, key). Relevant constant values used as values in data properties shall be identified. The description of properties may be represented in a tabular style. The details can be also represented graphically in UML diagrams. This step amalgamates the tasks "*describe ad hoc binary relation*", "*describe instance attributes*" and "*describe constants*" of Methontology.

   (b) **The definition of instances**. Relevant instances that appear in the concept dictionary should be defined. The definition of instances may take the form of a tablet. This step is the task "describe instances" in Methontology.

   (c) **The definition of formal axioms**. The goal of this task is the specification of the axioms that can be derived from the taxonomy, the concept dictionary and the definition of properties. Methontology proposes the use of first order logic to describe the axioms. In our approach, the method and level of formalization is specified in the requirements specification. Each axiom should be annotated with a name and a description. This step correspond with the task "describe formal axioms" in Methontology

   (d) **The definition of production rules**. The definition of rules follows the prescriptions of the task "define rules" of Methontology. Each rule is expressed using the template "*if <and conditions> then <only one consequent>*". The description of each rule includes at least rule name, a description and the rule expression. The method of formalization is specified in the ontology requirements specification.

(e) **Creation of the first prototype**. The serialization method is specified in the requirements specification. For example, if the use of the ontology is in the field of the Semantic Web, the prototype should consist of an OWL 2 ontology document, and a rule document in a RIF dialect (Paschke et al., 2009) or in the dialect of a rule engine. Both documents contain the axioms, assertions and rules identified in the previous steps.

4. **Evaluation**. The first prototype is evaluated against the requirements of the ontology. The goal of the evaluation is the identification of strengths (e.g. a set of functional requirements has been fulfilled), weaknesses (e.g. the ontology does not have the granularity expected), and risk (e.g. the ontology fails to represent a set of concepts). The analysis may result in the redefinition of actual requirements, or in the definition of additional requirements for the second prototype.

5. **Next incremental cycle**. A second prototype is evolved. If the development is incremental, the first step is the selection of the additional requirements to be addressed in this prototype. Then, the glossary of terms is updated to capture the details of the new and modified requirements. The iterative process continues up to the evaluation of the ontology considers that the ontology fulfils the requirements of its specification.

3.3 Requirements

The specification of the OntoOWS ontology is partially based in the structure of the requirements template proposed by Suárez-Figueroa et al. (2009). This specification defines:

- **Purpose**. The purpose of building the ontology is to provide a unified knowledge model of the different OGC Web service specifications that can be used by systems that deal with OGC Web service descriptions encoded in XML.

- **Scope**. The universe of discourse is the OGC Web service instances described by metadata documents encoded in XML as describes Whiteside (2007). Henceforth, Whiteside (2007) will be identified as the OGC Web services Common Specification. The ontology is focused on services that implement the standard baseline of standards about Web services defined by OGC. The OGC Web service architecture is loosely based in the *Reference Model of Open Distributed Processing* (RM-ODP) family of standards (ISO/IEC 10746-2 Foundations (2009), ISO/IEC 10746-3 Architecture (2009), ISO 15414 Enterprise language (2006)). RM-ODP concepts can be used when required to describe better an OGC Web service instance. The ontology shall be decomposed in modules that correspond to five system viewpoints, called *enterprise*, *information*, *computational*, *engineering* and *technology* (see Table 3.4) defined in the document ISO/IEC 10746-3 Architecture. Each module has its own level of granularity.

Table 3.4: RM-ODP Viewpoints.

Viewpoint	Definition
Enterprise	A viewpoint on a system and its environment that focuses on the purpose, scope and policies for that system.
Information	A viewpoint on a system and its environment that focuses on the semantics of information and information processing.
Computational	A viewpoint on a system and its environment that enables distribution through functional decomposition of the system into objects that interact at interfaces.
Engineering	A viewpoint on a system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system.
Technology	A viewpoint on a system and its environment that focuses on the choice of technology in that system.

- **Level of formality**. Portions of the ontology should be amenable to implementation using production rule systems, such as DLJena (Meditskos and Bassiliades, 2008) or OWLIM (Kiryakov et al., 2005). The maximum level of complexity should be $\mathcal{SROIQ}(\mathcal{D})$, which is supported by popular reasoners such as Pellet (Parsia and Sirin, 2004) and HermiT (Motik et al., 2009b).

- **Intended uses**. The scenarios in the context of this thesis are:

  – To persist in machine processable form the information found about an OGC Web service instance in a service metadata document found elsewhere (Chapter 2).

  – To provide a user with machine processable description of the behaviour and the information accessible of an OGC Web service instance (Chapter 5).

  – To provide a server with the information required for setting up semantic endpoints (Chapter 5).

  Out of the scope of this thesis, and part of the future work described in Chapter 6, are the use of the ontology as:

  – A model for describing future OGC Web service specifications.

  – Part of a system for the generation of OGC Web service metadata documents from a semantically rich description of an OGC Web service instance.

  – A low level model that can be used in combination with a high level model that describes a SDI such as the work of Béjar et al. (2009).

- **Intended end-users**. The users in the context of this thesis are the focused crawler described in Chapter 2, the semantic proxy described in Chapter 5, and the developer of both systems.

Out of the scope of this thesis, and part of the future work described in Chapter 6, are publishers responsible for the quality of the description of an OGC Web service, and researchers interested in OGC Web services.

- **Non-functional requirements**. This specification identifies the following requirements:
  - **Compliant with OGC standards**. The ontology shall be based on the documents and information models that form the OGC standards baseline and the OGC Web services Common Specification.
  - **Compliant with RM-ODP standards**. The ontology shall be based on the RM-ODP family of standards where the OGC documents do not provide information. RM-ODP documents rules over OGC documents in relation with RM-ODP terms.
  - **Flexible**. The ontology should be able to model future OGC specifications based in the principles of the OGC Web services Common Specification.
  - **Modular**. The ontology should be able to be decomposed in self-contained *ontology modules* (Doran et al., 2007) that correspond to the *viewpoints* of RM-ODP: *enterprise*, *information*, *computational*, *engineering* and *technology*. An *ontology module* is a reusable component of a larger or more complex ontology, which is self-contained in its relations and concept centred. For example, an enterprise module is centred in enterprise concepts; an information module is centred in information concepts, and so on.
  - **Multilingual**. The ontology should be able to represents human readable text in different languages.
  - **Reuse**. The ontology shall reuse other ontologies for representing descriptions or properties about location or people found in an OGC Web service metadata document.
  - **Source**: The ontology shall use as source the OGC standards baseline and the RM-ODP standards. The work of Béjar (Béjar et al., 2009; Béjar, 2009) on architectural styles for Spatial Data Infrastructures based in RM-ODP is used as source of terms and concepts to reduce the gap between OGC and RM-ODP in the *enterprise* module. The standard for Geographic Metadata ISO 19115 (2003b) is used as source of terms and concepts related with the organization of spatial datasets.
  - **Use**: A knowledge base that uses this ontology shall be able to use OGC Web service metadata documents as source of assertions about OGC Web services.
- **Functional requirements**. This specification identifies the next requirements in relation with the description of a system that implements an OGC Web service:
  - **Objects**: The ontology shall answer which are the objects involved in the system. That is, the ontology shall describe the service instances, the datasets and the agents that can interact in the system.

- **Communities**: The ontology shall answer which describe the community of objects that are part of the system, and the relations among them. That is, the ontology shall model as a group the service instances and the datasets that form part of the system and describe the rules and constraints that can be derived from the service metadata.
- **Interactions**: The ontology shall answer which interactions are offered by the system, and which roles can play the objects involved or referenced in an interaction.
- **Policies**: The ontology shall answer which are the general constrains or policies that applies to the system that can be derived from the service metadata.
- **Information types**: The ontology shall be able to answer which types of information are exchanged by the system
- **Operations**: The ontology shall be able to answer which operations are supported by the system, and their signature.
- **Platform bindings**: The ontology shall be able to answer which distributed computing platform are used by the system, which are the logical address of the operations in each supported DCP, and which are the supported interaction styles, i.e. how the requests are send.
- **Implementable standards**: The ontology shall be able to answer which standards and information models implements a system that implements an OGC Web service.

3.4 Ontology

This section contains a description of the main elements of the OntoOWS ontology. This description is focused in the presentation of the elements that can be found in the *Concept Dictionary* of the ontology. The structure of this section is as follows. First, a brief introduction to OGC Web service metadata documents is given. Next, the general structure and the assumptions in the formalization of OntoOWS are presented. The rest of the section describes OntoOWS thoroughly.

3.4.1 Introduction to OGC Web service metadata documents

Any knowledge base that uses the OntoOWS ontology shall be populated from information extracted from OGC Web service metadata documents. This section introduces these documents. Many of the concepts introduced here are used elsewhere when the concepts and roles of the ontology are presented.

An OGC Web *service metadata* document shall be the normal response to a client from performing the *GetCapabilities* operation to an OGC Web service, and shall contain metadata appropriate to the specific server for the specific OGC Web service (see Whiteside, 2007). A service metadata document describes a current configuration formed to provide access to, or work with, spatial data

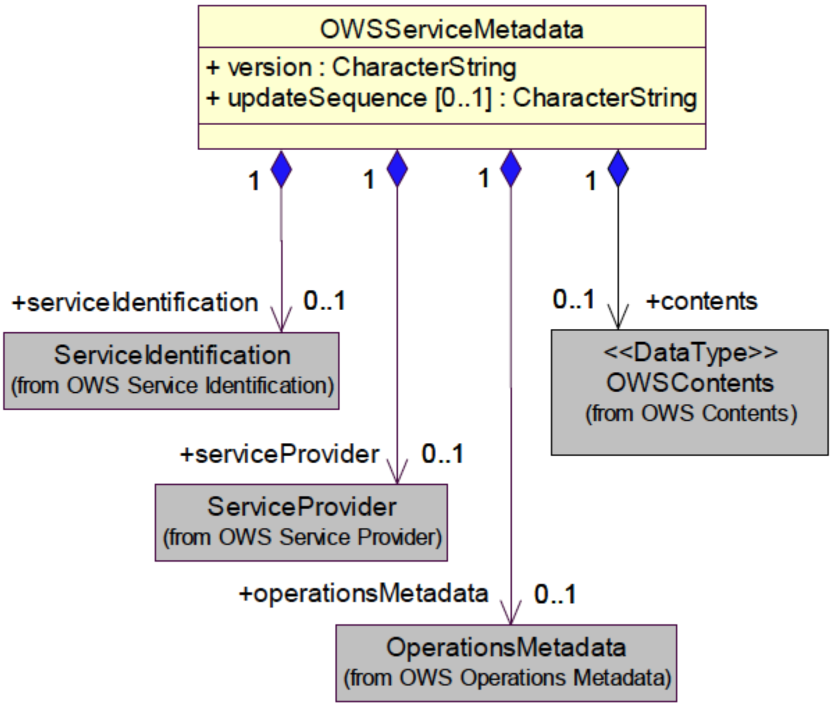


Figure 3.6: OGC Web service metadata returned by the *GetCapabilities* operation (pg. 24, Whiteside, 2007).

and metadata. In the context of OGC Web services, the following entities can be found (see figure 3.6):

- A spatial service that provides access to, or work with, spatial data or metadata through an interface (`ServiceIdentification`)
- The operations provided by this service instance (`OperationsMetadata`).
- The contents served by this spatial service (`OWSContents`).
- An organization responsible of the service instance (`ServiceProvider`).
- The service metadata that describes the above entities (`OWSServiceMetadata`).

The `ServiceIdentification` section of the *service metadata* document should contain basic metadata about the service instance that originated the document. The metadata can include the *title*, the *abstract* and a set of *keywords*. The *title* is the name of the *service instance* usually used for display to a human. The *abstract* is a brief narrative description of the *service instance* that can be available for display to a human alongside the *title*. The *keywords* are commonly used or formalised words or phrases used to describe the *service instance.* The *keywords* allow including descriptions in multiple languages and using several keywords authorities. The `ServiceIdentification` section does not provide a unique identifier of the service. The information about the *service instance* is available in the OGC Web service specifications since WMS 1.0.0 (Doyle, 2000) with few changes.

The `OperationsMetadata` section provides metadata about the operations specified by a service and implemented by the server, including the URLs and bindings for operation requests. The conceptual content of this section is the same for all OGC Web services. That is, the identification of the available operations the description of the bindings to DCP, and the URLs for each operation request. Individual services may add elements and/or change the optionality of optional elements.

The `OWSContents` section of an OGC capabilities document normally contains metadata about the information acceded from the *spatial dataset* served by the service. This information is highly dependent of each OGC Web service standard. Usually, the contents section points to the metadata of the top-level datasets. If the content is data (e.g. features) or a representation of data (e.g. maps), the contents metadata is arranged in a hierarchy. The properties described in the service metadata are consequence of the specific role that the dataset plays in the service instance and may vary from service instance of service instance. The OGC Web services common specification (Whiteside, 2007) defines a set of standard metadata parameters that should be used whenever applicable by OGC specifications. These parameters include *title*, an *abstract*, *keywords*, *language* used for contents, links to external *metadata*, an *identifier*, a *point of contact*, available *output formats*, available *coordinate reference systems* (CRS), and *bounding boxes.* The *identifier* is an optional property whose value is an unambiguous identifier of the dataset within the *service instance.* The *point of contact* may reveal a responsible of the dataset different from the responsible of the service instance. The *output format* is a reference to a format in which output data from this service instance may be encoded. The CRS informs in which CRS data from this dataset may be returned by the service instance. The *bounding boxes* describe the maximum extent of accessible geospatial data given a coordinate reference system. The spatial description provided by a bounding box is partial, because if a dataset has several bounding boxes should not be automatically inferred that the extent is the union of all the bounding boxes nor all the bounding boxes are equivalent.

The `ServiceProvider` section of a service metadata document contains metadata about the *organization* operating this server. The OGC Web services Common Specification expects that the service provider is an organization. The metadata should provide its *name.* The metadata also can provide information for contacting the service provider through a *point of contact.* The *point of contact* may be a person, a department, a postal box, a phone, etc. The description of the point of contact is based in `CI_ResponsibleParty` and subsidiary classes of ISO 19115 (ISO/TC 211, 2003b). This data structure is quite similar to a *business electronic card.* It can contain name, position and address information, phone numbers, fax numbers and email addresses. The contact information was introduced in WMS 1.1.0 (de la Beaujardiere, 2001).

The most common way to obtain an `OWSServiceMetadata` is a *GetCapabilities* request encoded in a HTTP GET request. The same operation can be *encoded* in many alternative ways, each appropriate to one or more specific *platforms*. The most common *encoding* for a HTTP GET request is named *Keyword Value Pair* (KVP) in OGC terminology. The next URL is an example of a *GetCapabilities* request encoded in KVP in a HTTP GET request with values appropriate for a WMS 1.3.0 service instance:

```
http://www.idee.es/wms/WMS-Cantabria/WMS-Cantabria?REQUEST=
GetCapabilities&SERVICE=WMS&ACCEPTVERSION=1.3.0
```

In the HTTP platform, the address of the above *GetCapabilities* request is the URL `http://www.idee.es/wms/WMS-Cantabria/WMS-Cantabria`. The parameter `service` identifies which *service type* is requested and the parameter `acceptversion` which *service type versions* accept the client. The parameter `acceptversion` is optional and defaults to the latest supported version. The service must return the service metadata document that matches with the service and version requested. The values of *service type* and *service type version* are present in the retrieved OGC Web service metadata document. Each service metadata document may include an identifier of the version of the configuration named *update sequence*. The *update sequence* must change whenever any change is made in the document. By definition, the meaning of *update sequence* values is opaque to clients. Service instances are not required to support configuration versioning. The attributes *service type*, *service type version*, *update sequence*, *URL* are part of the attributes of an OGC Web service metadata document. If the *update sequence* is not available, it is possible to create a synthetic update sequence by computing a *message digest* of the capabilities document.

3.4.2 General structure and assumptions

The OntoOWS ontology provides a framework that helps to translate in a consistent way OGC Web service metadata encoded into assertions about the service instances, allowing further enrichment. OGC specifications provide a rich description for each type of service of its purpose and the information contents that the user can expect. These descriptions include not only information about the interfaces, the data types and the platform bindings but also expected behaviour, policies, and technological choices. Figure 3.7 provides a high level overview of the translation. The following modules compose the structure of OntoOWS:

- **Core objects**, that is, the service instance, the datasets, the metadata, and the users.
- **Purpose, scope and policies**. This section represents the enterprise view of the system, i.e. the behaviour and its constraints.
- **Information types**. This section represents the information view of the system, i.e. the information exchanged in the interactions.

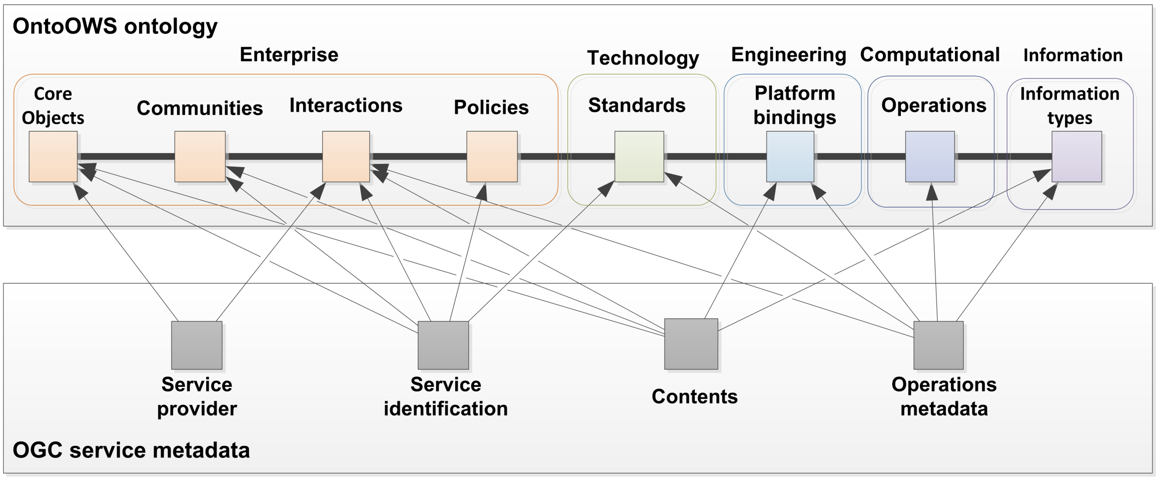


Figure 3.7: Illustration of the transformation from OGC Web service metadata into OntoOWS.

- **Operation**. This section represents the computational view of the system, i.e. the operations related with the behaviour.
- **Distributed Platforms bindings**. This section represents the engineering view of the system, i.e. how the system uses distributed platforms for distributed interactions.
- **Implementable standards and information models**. This section represents the technology view of the system, i.e. its technological choices.

Each section contains a Description Logic formalization of the presented terms. Table 3.5 contains the constructors and axioms used in the formalization. All the definitions are available encoded in OWL in Appendix A. The OntoOWS ontology reuses concepts from several ontologies:

- *gn:GeographicEntity* defines the concept geographical entity in Geo-Net (Chapter 4).
- *tm:TemporalEntity* defines the concept temporal entity in Time Ontology (Hobbs and Pan, 2006).
- *vcard:VCard* defines the concept business card in VCard (Halpin et al., 2010).
- *rdf:PlainLiteral* defines a data type equivalent to the plain literal of RDF (Bao et al., 2009).
- Basic XML Schema data types (Biron and Malhotra, 2004), such as *xs:string*, *xs:hexBinary* and *xs:anyURI*.

The prefixes *gn:*, *tm:*, *vcard:*, *rdf:*, and *xs:* identifies concepts imported from Geo-Net, Time Ontology, VCard, RDF, and XML Schema respectively. Reused concepts are pairwise disjoint each other and with respect to OntoOWS defined concepts. The following primitive siblings introduce

Table 3.5: Terminological constructors, axioms and syntactic sugar used in the formalization.

(a) Terminological constructors.

Concept constructor		**Role constructor**	
Name	***DL Syntax***	***Name***	***DL Syntax***
Concept C	C	Role R	R
Universal concept	$\top$	Inverse of role R	R^{-1}
Bottom concept	$\bot$	Role chain R and S	$R \circ S$
Intersection of C_1, C_2	$C_1 \sqcap C_2$		
Union of C_1, C_2	$C_1 \sqcup C_2$		
Complement of C	$\neg C$		
One of O_1, O_2	$\{O_1,\, O_2\}$		
Role R some values from C	$\exists R.C$		
Role R all values from C	$\forall R.C$		
Role R has values O_1, O_2	$\exists R.\,\{O_1,\, O_2\}$		
Role R has min cardinality n	$\geq n\, R$		
Role R has max cardinality n	$\leq n\, R$		
Role R has exactly cardinality n	$= n\, R$		
Role R has min cardinality n from C	$\geq n\, R.C$		
Role R has max cardinality n from C	$\leq n\, R.C$		
Role R has exactly cardinality n from C	$= n\, R.C$		

(b) Terminological axioms.

Concept axioms		**Role axioms**	
Name	***DL Syntax***	***Name***	***DL Syntax***
Concept inclusion: C_1 implies C	$C_1 \sqsubseteq C$	Role inclusion: R_1 implies R	$R_1 \sqsubseteq R$
Concept equality: C_1 equivalent to C	$C_1 \equiv C$	Role equality: R_1 equivalent to R	$R_1 \equiv R$
		Role R is functional	$Func\,(R)$
		Role R is transitive	$Trans\,(R)$

(c) Syntactic sugar.

Name	***Syntax***	***DL Equivalence***
Disjoint pairwise	$disjoint\,(C_1, C_2, \ldots, C_n)$	$C_i \sqsubseteq \neg C_j,\, 1 \leq i < j \leq n$

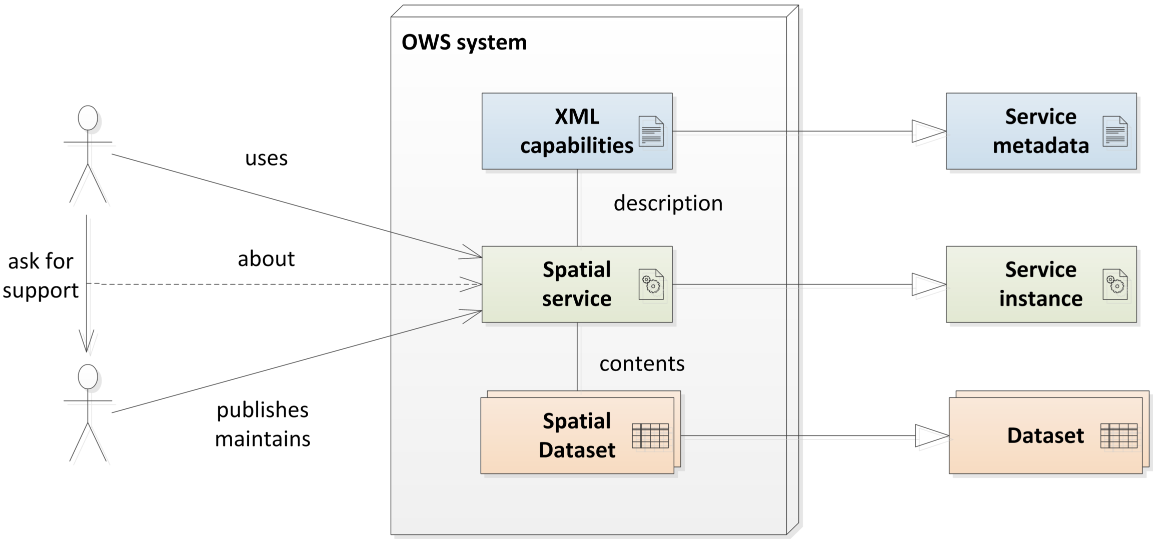


Figure 3.8: Essential players and interactions in a system that implements an OGC Web service.

in OntoOWS are pairwise disjoint: *Encoding*, *Endpoint*, *ExtendedLiteral*, *ImplementableStandard*, *InformationAggregate*, *InformationModel*, *InformationType*, *Interaction*, *Object*, *Payload*, *Platform*, and *PolicyObject*.

3.4.3 Core objects

The OGC Abstract Specification (OGC AS, Percivall, 2002) defines a framework for *providers* to set up *services* described with *metadata* that enables *users* to access and process *geospatial data.* Providers, services, users, metadata, geospatial data are the essential players involved in an OGC system (Figure 3.8). The user uses a service instance to access or process a spatial dataset. Before this access, the user may request a self-description of the service in form of an OGC Web service metadata document. The service provider is the responsible for the accessibility of the service and the appropriateness of the metadata document. The service metadata document can contain information that the user can use to ask the service provider for support.

The ontology shall answer which are the *objects* involved in the system. That is, the ontology shall describe the service instances, the service providers, the datasets and other objects that can interact in the system.

An *object* is a model of a distinguishable entity characterized by its behaviour and by its state. The conceptualization of the objects that plays a role in the system should ignore the factual role in the system. From this point of view the user of a service instance may no differ as concept from the provider of service. Users, service providers and services are characterized by manifest some behaviour when they interact. They are *agents*. When the agent that interacts is a *service instance*,

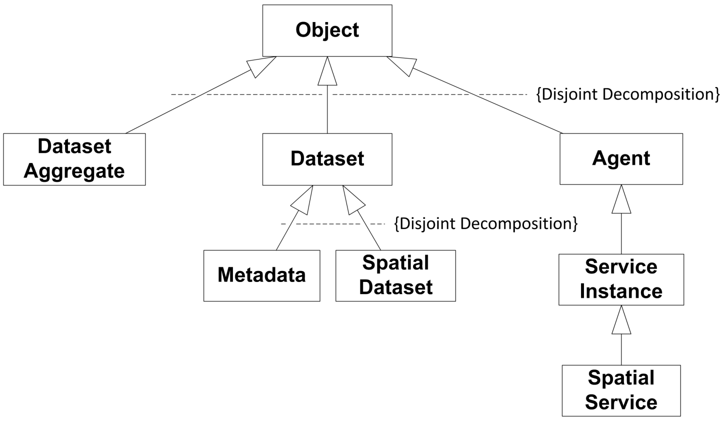


Figure 3.9: Core objects: class hierarchy.

that is, a software system with interfaces that has been designed for performing a set of task mainly related with spatial data and metadata, it can be represented by the concept *spatial service* (Béjar et al., 2009).

The definition of service provider in the OGC Web services specifications is often too restrictive. Usually, the term service provider is related with the term organization. The term organization implies a stable entity formed by people with a certain purpose, and guided by a set of, typically formal, rules. Persons and small teams or groups can be responsible of the availability of a service instance. Hence, the conceptual model uses the concept agent instead for identifying service providers.

The *datasets* are collections of data that represents a state. Datasets can be manipulated as collection or *aggregations of datasets*. The concept aggregation of datasets is used often in the geospatial domain (ISO/TC 211, 2003b). A dataset that contains a structured description of a resource, for example, the service metadata, is a special kind of dataset named *metadata*. If the dataset contains a collection of data related to geographic locations, the concept *spatial dataset* (Béjar et al., 2009) can denote them. Figure 3.9 shows the taxonomy of these concepts.

The next concepts have been identified in the conceptualization and shall be formalized: *Object*, *Dataset*, *Metadata*, *SpatialDataset*, *DatasetAggregate*, *Agent*, *ServiceInstance* and *SpatialService*.

$$Dataset \sqsubseteq Object \tag{3.1}$$

The atomic concept object represents an entity characterized by its behaviour and/or by its state. A dataset is any identifiable collection of data. A dataset may be a smaller grouping of data, though limited by some constraint such as spatial extent or feature type, is located physically within a larger dataset. A dataset may be as small as required (e.g. a single feature or feature attribute contained within a larger dataset).

$$Metadata \equiv Dataset \sqcap \exists describes.\top \sqcap \exists isSpecifiedBy.\top \tag{3.2}$$

A metadata is defined as a dataset that provides a structured description of a resource according to a standardised concept using a well-defined metadata scheme.

$$SpatialDataset \sqsubseteq Dataset \tag{3.3}$$
$$disjoint\,(Metadata, SpatialDataset) \tag{3.4}$$

A spatial dataset is a collection of data related to geographic locations. Spatial dataset and metadata are disjoint concepts. Metadata can contain geographic locations about the location of a resource, but this information is not the same as the geographic data contained in that resource.

$$\begin{aligned} DatasetAggregate \equiv\ & Object \sqcap (\exists composedOf.\top \sqcap \forall composedOf.Dataset \sqcup \exists subset.\top \sqcap \\ & \forall subset.DatasetAggregated \sqcup \exists superset.\top \sqcap \forall superset.DatasetAggregated) \end{aligned} \tag{3.5}$$
$$subset \equiv superset^{-1} \tag{3.6}$$
$$Trans\,(subset) \tag{3.7}$$
$$Trans\,(superset) \tag{3.8}$$

A dataset aggregate is a collection of datasets. The subset or the superset of a dataset aggregated is also a dataset aggregated. Dataset and dataset aggregated are not disjoint concepts. That is, a larger dataset can be considered as dataset aggregated if it can be decomposed in smaller datasets. The superset role is the inverse of the subset role. Both roles are transitive.

$$Agent \sqsubseteq Object \tag{3.9}$$
$$disjoint\,(Agent, Dataset \sqcup DatasetAggregated) \tag{3.10}$$

An agent is a person or thing that takes an active role or produces a specified effect. Agent is disjoint from the concepts dataset and dataset aggregate.

$$ServiceInstance \sqsubseteq Agent \tag{3.11}$$

$$SpatialService \sqsubseteq ServiceInstance \tag{3.12}$$

A service instance is any particular instance of a service. A spatial service is a service instance that deals with spatial data and metadata about spatial resources. The differentiation between service instances and spatial services is useful. The service instance can be used for classifying troublesome services. For example, the set of service instance can include broken OGC Web service instances and OGC Web service instances that do not give access or not process spatial data or metadata.

3.4.4 Purpose, scope and policies

Generic communities and contracts. The ontology shall describe the community of objects that are part of the system and the relations among them from the available evidence. That is, the ontology shall model as a group the service instance and the datasets that form part of a system, and describe the rules and constraints that can be derived from an OGC Web service metadata document.

A *community* is a group of *objects* formed to meet an objective and able to interact with the context or environment where the objects operate. For example, an map service instance can be analysed as a community formed by the service instance, data, metadata and configuration files whose objective is to allow clients to retrieve parts of identified geographic data as image maps. This community interacts mainly with clients that request maps from the service. Table 3.6 contains examples of generic objectives of different OGC services.

The involvement of the environment in an interaction implies that the interaction is observable, and thus, that it is possible to describe the interaction and its objects. The objective is expressed in a *contract*, which expresses how an objective can be met by defining interactions and roles required, assignments of objects to the roles, policies governing the collective behaviour and indications of the duration or periods of validity. The *interactions* are the actions required to meet the objective. For example, the main interaction in a Web map service is to get a data subset as an image map. A *role* is defined in terms of constraints to the behaviour of the object playing the role in an interaction. Examples of roles in OGC web services are the roles defined as essential in the *publish/find/bind pattern* (OGC, 2008): service, service consumer and service directory. The assignment of objects to the roles means that each object that belongs to the community has at least a role in at least an interaction owned by the community. Policies are rules related to a particular purpose that can govern several objects. Interactions, roles and policies are analysed later. These core concepts are represented in Figure 3.10.

The next concepts have been identified in the conceptualization and shall be formalized: *Community*, *Contract, Interaction, Role*, *InterfaceRole*, *PolicyObject* and *TemporalRegion. Interaction*,

Table 3.6: Objectives of OGC Web services.

Service	**Objective**
WMS	To allow clients to access part of identified layers (see Doyle, 2000; de la Beaujardiere, 2001, 2006). A layer is a map encoded a binary image format that portrays geographic data.
WCS	To allow clients to access part of identified quadrilateral grid coverage (see Evans, 2003; Whiteside and Evans, 2006; Baumann, 2010). A quadrilateral grid coverage is geographic data encoded in a binary image format.
WFS	To allow clients to access a set of features with the desired values (see Vretanos, 2002, 2005b, 2010). A feature is geographic data encoded in GML or equivalent format.
CSW	To allow clients to access a set of metadata records describing geospatial resources, which may be a data set, service, and any other information (see Nebert et al., 2007).

Role, *InterfaceRole* (subclass of *Role*) and *PolicyObject* will be defined later. The concept *TemporalEntity* is imported from the Time Ontology that represents time.

$$Community \equiv \exists composedOf.\top \sqcap \forall composedOf.Object \sqcap \exists isSpecifiedBy.Contract \tag{3.13}$$

A community is a collection of object able to interact formed to meet an objective specified by a contract.

$$Contract \equiv = 1\, specifies \sqcap \forall specifies.Community \sqcap \exists coverage.tm : TemporalEntity \sqcap$$
$$\exists owns.Interaction \sqcap \exists owns.InterfaceRole \sqcap \geq 2\, owns.Role \sqcap$$
$$\forall owns.\,(Interaction \sqcup Role \sqcup PolicyObject) \tag{3.14}$$
$$owns \equiv isOwnedBy^{-1} \tag{3.15}$$
$$Func\,(isOwnedBy) \tag{3.16}$$

A contract is an agreement governing part of the collective behaviour of a set of objects. A contract specifies obligations, permissions and prohibitions for a community of objects. The specification of a contract may include the behaviours or interactions (min one *interaction*). A contract

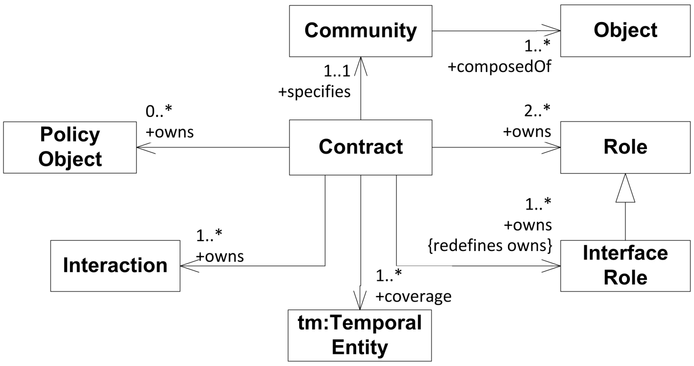


Figure 3.10: Generic communities and contracts.

also includes the roles that objects involved in the contract may assume (min two *roles*, one of them played by objects outside of the community) and the roles that interface with the environment (min one *interface roles*). Policies or constraints, such as quality of service constraints, indications of behaviour that invalidates the contract, and liveness and safety conditions (*policy objects*), and indications of duration or periods of validity (min one *temporal entity*), are also part of the contract. The contract is the owner of such specifications. The role *owns* has as inverse the role *is owned by*, which is functional. That is, an owned item has only one owner.

OGC Web service communities and contracts. The `ServiceIdentification` section of the service metadata document describes the system whose objective is the access to spatial data, and some of the constraints that applies to this interaction (see Figure 3.11). In this sense, the service identification identifies the properties and constraints of the community of objects involved in the access to spatial data as a whole. Hence, the description of the community can include the *title*, the *abstract* and a set of *keywords* found in the service metadata document. The *title* is the name of the *service instance* usually used for display to a human. The *abstract* is a brief narrative description of the *service instance* that can be available for display to a human alongside the *title*. The keywords may be related with concepts. In order to enable this possibility the keywords should be modelled as concepts, that is, as extended literals, rather than plain literals[4].

In our conceptual model, each *OWS (OGC Web service) metadata* describes a community ruled by a contract that follows an OGC specification. The capabilities document is modelled as an instance of the concept *OWS metadata* and the contract as an instance of the concept *OWS contract.* The concept *OWS metadata* refines the concept *service metadata,* which in turn refines the concept *metadata.* An *OWS metadata* provides information about a spatial service, its objectives (e.g. to

[4] This approach is the same as the approach for extending labels in the SKOS-XL vocabulary (Miles and Bechhofer, 2009).

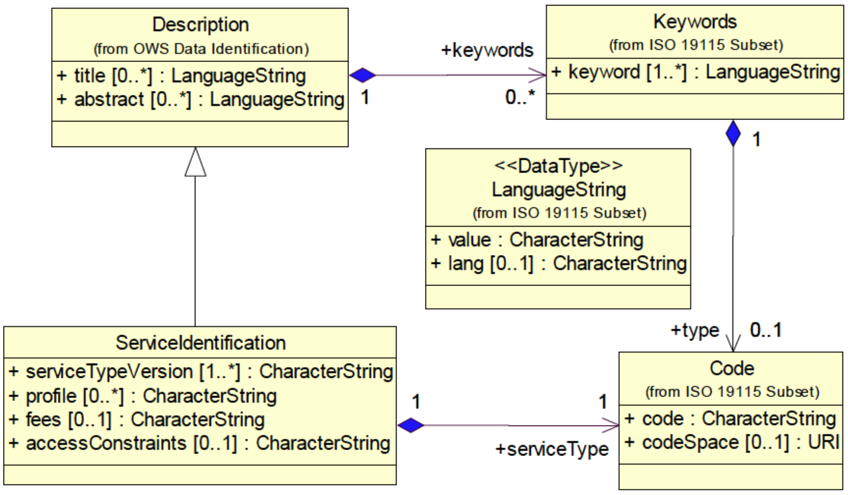

Description
(from OWS Data Identification)
+ title [0..*] : LanguageString
+ abstract [0..*] : LanguageString
+keywords
1
0..*
Keywords
(from ISO 19115 Subset)
+ keyword [1..*] : LanguageString
1
<<DataType>>
LanguageString
(from ISO 19115 Subset)
+ value : CharacterString
+ lang [0..1] : CharacterString
+type
0..1
ServiceIdentification
+ serviceTypeVersion [1..*] : CharacterString
+ profile [0..*] : CharacterString
+ fees [0..1] : CharacterString
+ accessConstraints [0..1] : CharacterString
1
1
+serviceType
Code
(from ISO 19115 Subset)
+ code : CharacterString
+ codeSpace [0..1] : URI

Figure 3.11: `ServiceIdentification` section UML class diagram (pg. 26, Whiteside, 2007).

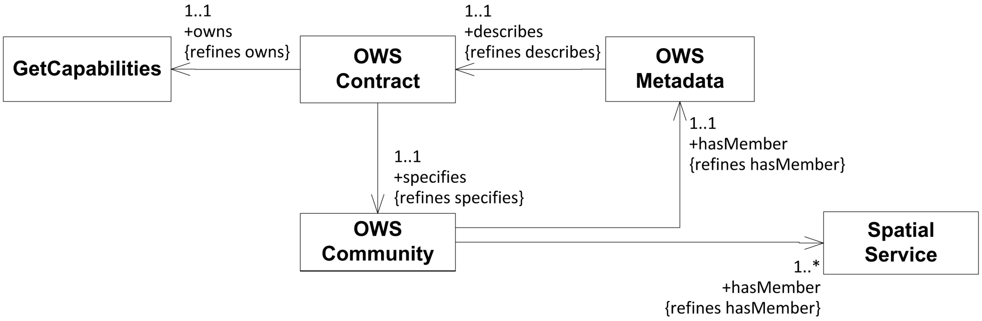


Figure 3.12: The relations of the OGC Community.

provide access to datasets) and how they can be met. The community defined is an instance of the concept *OWS community* and has one of its members the *OWS metadata* document that describes the contract. Figure 3.12 describes the relations involved.

The next concepts have been identified in the conceptualization and shall be formalized: *OWSCommunity*, *OWSContract*, *OWSMetadata*, *ExtendedLiteral*, and *GetCapabilities*. *GetCapabilities* is a specialization of *Interaction* that will be defined later.

$$OWSContract \equiv Contract \sqcap = 1\, owns.GetCapabilities \sqcap \\ \exists specifies.OWSCommunity \tag{3.17}$$

An OWS contract is the contract of the OWS community that form an OWS server. The characteristics OWS communities are available through a *GetCapabilities* interaction.

$$\begin{aligned}&OWSCommunity \equiv Community \sqcap \exists composedOf.OWSServiceMetadata \sqcap \\ &\quad \exists composedOf.SpatialService \sqcap \exists isSpecifiedBy.OWSContract && (3.18)\\ &OWSCommunity \sqsubseteq \forall hasKeywords.ExtendedLiteral \sqcap \forall title.rdf\colon PlainLiteral \sqcap \\ &\quad \forall abstract.rdf\colon PlainLiteral && (3.19)\end{aligned}$$

The OWS community is a community specified by an OWS contract composed at least by a spatial service and an OWS service metadata document. An OWS service metadata document provides titles, abstracts and keywords that describe the community.

$$\begin{aligned}&OWSMetadata \sqsubseteq Metadata \sqcap \exists describes.OWSContract \sqcap \\ &\quad \exists location.xs\colon anyURI \sqcap \forall location.xs\colon anyURI \sqcap \exists serviceTypeVersion.xs\colon string \sqcap \\ &\quad \forall serviceTypeVersion.xs\colon string \sqcap = 1\, serviceType.xs\colon string \sqcap \forall serviceType.xs\colon string \sqcap \\ &\quad \leq 1\, updateSequence.\, (xs\colon hexBinary \sqcup xs\colon string) \sqcap \\ &\quad \forall updateSequence.\, (xs\colon hexBinary \sqcup xs\colon string) && (3.20)\end{aligned}$$

A service metadata document shall be the normal response to a client from performing the *GetCapabilities* operation, and shall contain metadata that describes the contract appropriate to the specific server. A service metadata document has as datatype properties its location in the Web, the identifier of the type of service, the identifiers of the versions supported and the update sequence (or the synthetic digest).

$$\begin{aligned}&ExtendedLiteral \sqsubseteq \exists literalForm.rdf\colon PlainLiteral \sqcap \\ &\quad \forall literalForm.rdf\colon PlainLiteral && (3.21)\end{aligned}$$

The extended literal is a placeholder for keywords found in a service metadata documents that cannot be easily mapped into concepts.

Interactions and roles. The ontology shall answer which interactions are offered by the system and which roles can play the objects involved or referenced in an interaction. The behaviour of a community is such that it meets its objective. The collective behaviour of a community may be specified in terms of one or more of the following elements in a contract: the *interactions* of the

Table 3.7: Roles imported from Béjar (2009).

Type	Definition in Béjar (2009)	Rationale for re-use
Contributor	They contribute and/or withdraw the assets, i.e. datasets or services, they own or control. A contribution is understood as a way to make some assets available to the users of an SDI, i.e. they are findable and there is a way to get or use them. It does not require the assets are for free and it may be necessary to get a license from the contributor.	The OGC Web services common specification (Whiteside, 2007) explicitly says “the service provider section of a service metadata document contains metadata about the organization operating this server”. The term is used in a broader sense, i.e. accessible to any public user.
Spatial asset	Any useful or valuable spatial information resource that can be made accessible to the users of an SDI.	The term is co-opted but in a broader sense, i.e. accessible to any public user.
Spatial asset metadata	A type of spatial asset that provides information, i.e. a structured description about another spatial dataset.	This concept describes the purpose of an OGC Web service metadata document.

community, the *roles* of the community, the assignments of roles to interactions and the *policies* that apply to the roles and interactions. The policies are discussed in the next section.

Interactions are the observable actions associated to an object that take place with the participation of its environment. Thus, an interaction is a piece of shared behaviour, with no necessarily a single initiator. A *role* identifies those aspects of the behaviour of an object required to form part of an interaction. The role links those aspects of an instance of an interaction as constraints in the behaviour on an actual object. That is, once linked, the actual object *plays* a role in an instance of the interaction. The role is named *interface role* if plays with objects that are not members of the community that owns the role. The role is named *resource* if the object is an essential asset with respect to the behaviour. If the object participates in the action, the role is named *actor*. If an actor can initiate the interaction, it is named *initiator*. If an actor responds to the actions of other objects in the interaction, it is named *responder*. If the object is referenced in the action, the role is named *artefact*. Roles can have additional information including the named actions or *operations* in which the behaviour can be decomposed, the structure of the information involved in the behaviour or additional facts related to that role. The concept *spatial role* identifies roles related with spatial data. Béjar (2009) proposes a model for SDIs that uses the enterprise language of RM-ODP. The service metadata can contain metadata about some of the roles described in that work; in particular, the roles *contributor*, *spatial asset* and *spatial asset metadata* (Table 3.7). These concepts are included in the conceptual model as specializations of the role *spatial role*.

The analysis of the interface specifications and the documents about the OGC web service architecture (see Whiteside, 2005, 2007) allows identifying four top-level categories of interactions in geographic model/information management services:

- **GetCapabilities**. The *get capabilities* interaction retrieves service metadata about the capabilities provided by any server that implements an OGC Web service interface specification. When a service instance plays the role of service in this interaction, the service participates with the operation *GetCapabilities*.
- **GetDataSubset**. The *get data subset* interaction gives access to a subset or a gridded part of the data available in the server. When a service instance plays the role of service in this interaction, the service participates with operations named as *GetRecords*, *GetFeatures*, *GetFeatureInfo*, *GetCoverage* or *GetMap*. Although the name of the operation and the type of resource are different, these operations not change the basic semantics of get data subset. There are two main variants:
  - **Get<X>Subset** gives access a subset of records (*GetRecords* in CSW) or features available (*GetFeatures* in WFS, *GetFeatureInfo* in WMS)
  - **Get<X>Part** gives access to a gridded part of a dataset (*GetCoverage* in WCS, *GetMap* in WMS).
- **GetResourceById**. The *get resource by id* interaction allows clients to retrieve one or more identified resources, including datasets (e.g. *GetRecordById*, *GetGmlObjec*t) and resources that describe datasets (e.g. *DescribeCoverage*, *DescribeRecord*, *DescribeFeatureType*) or parameters (e.g. *GetDomain*). The next variants can be identified:
  - **Get<X>ById** returns an identified instance of a dataset (*GetRecordById* in CSW, *GetGmlObject* in WFS).
  - **Get<X>Metadata** returns additional metadata (*DescribeCoverage* in WCS).
  - **Get<X>Model** returns the information model of a type as a XML Schema (*DescribeRecord* in CSW, *DescribeFeatureType* in WSF).
  - **Get<X>Domain** returns the current domain of values a data type (*GetDomain* in CSW).
- **Manage**. The *manage* interaction allows clients to manage the data served by the server. Typical manage actions are add, modify or delete a resource, lock a resource and harvest remote resources. The next variants can be identified:
  - **<X>Transaction** allows adding, modifying and removing resources (*Transaction* in CSW and WFS).

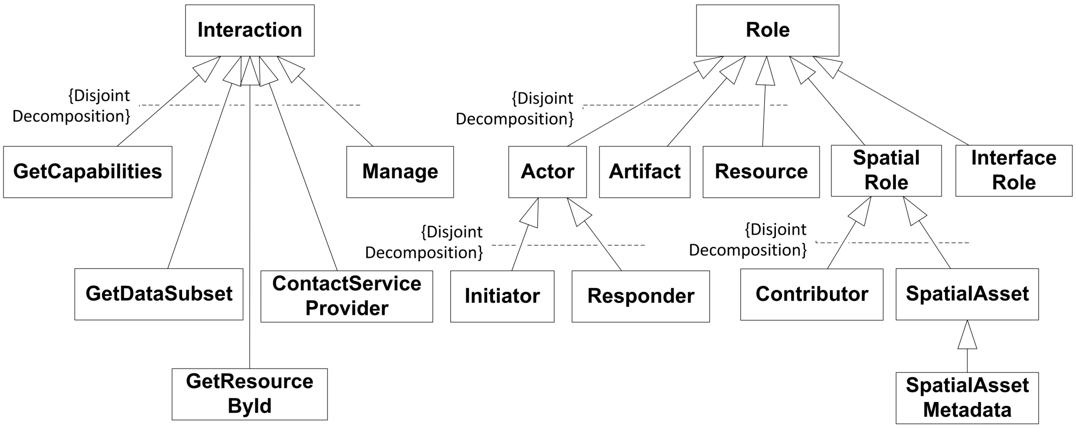


Figure 3.13: Class hierarchy of interaction and role concepts.

- **Lock<X>** allows locking a set of resources to prevent modification or deletion (*Lock-Feature* in WFS).
- **Harvest<*X*>** describes an interaction where a client requests a service to retrieve new and modified resources from a specified location, often on a regular basis (*Harvest* in CSW). This behaviour triggers interactions where the server behaves as client requesting resources from other servers.

These interactions are included in the conceptual model to provide a basis for further specialization. Additionally, the service metadata document contains metadata about the organization operating this server to provide service provider contact information. It is widely acknowledged the importance of service provider and customer communications. The service metadata is an essential resource in the *contact service provider* interaction as it contains a business card of the service provider meanwhile the service instance and the datasets plays the role of referred *artefacts*. Figure 3.13 presents the elements related with interactions and roles in the conceptual model.

The next concepts have been identified in the conceptualization and shall be formalized: *Interaction*, *Role*, *InterfaceRole*, *Operation, Actor, Initiator, Responder, Artefact, Resource, InformationType, SpatialRole, Contributor, VCard, SpatialAsset, SpatialAssetMetadata, GetCapabilities*, *GetDataSubset*, *GetResourceById*, *Manage* and *ContactServiceProvider.* The concepts *Operation* and *InformationType* are defined later. The concept *VCard* is imported from the vCard ontology represents a business card.

$$Interaction \sqsubseteq \geq 2\, hasRole \sqcap \forall hasRole.Role \tag{3.22}$$

The interactions are the observable actions, i.e. something with happens, associated to an object that take place with the participation of its environment. There are at least two roles in an interaction.

$$Role \sqsubseteq \forall isPlayedBy.Object \tag{3.23}$$

$$Role {\equiv} \exists isRoleOf.Interaction \tag{3.24}$$

$$isPlayedBy \equiv plays^{-1} \tag{3.25}$$

$$hasRole \equiv isRoleOf^{-1} \tag{3.26}$$

$$Func\left(isRoleOf\right) \tag{3.27}$$

A role is a formal placeholder in the specification of an interaction. It identifies those aspects of the behaviour of some object required for it to form part of the interaction and links them as constraints on an actual object in an instance of the interaction. An object can play several roles. Each role instance identifies only one interaction. Thus, the inverse of the relation *has role* (*is role of*) is functional.

$$InterfaceRole \sqsubseteq Role \sqcap \forall hasOperation.Operation \sqcap \exists hasOperation.\top \tag{3.28}$$

An interface role is the part of the behaviour identified by an interface. An interface role participates in an interaction where are identified the behaviour of the external objects. Interface roles are manifested through operations.

$$Actor \sqsubseteq Role \tag{3.29}$$

$$Artefact{\sqsubseteq}Role \tag{3.30}$$

$$Resource \sqsubseteq Role \sqcap \forall hasType.InformationType \tag{3.31}$$

$$Initiator \sqsubseteq Actor \tag{3.32}$$

$$Responder \sqsubseteq Actor \tag{3.33}$$

$$disjoint\left(Actor, Resource, Artefact\right) \tag{3.34}$$

$$disjoint\left(Initiatior, Responder\right) \tag{3.35}$$

An actor is a role that identifies an object that participates in an action. An actor can initiate or respond in an interaction. An artefact is a role that identifies an object referenced in the action. A resource is a role that specifies an object essential for an action. Resources may be typed. Actor, artefact and resource are concepts pairwise disjoint. Initiator and responder are disjoint concepts.

$$SpatialRole \sqsubseteq Role \tag{3.36}$$

$$Contributor{\sqsubseteq}SpatialRole \sqcap \forall hasBusinessCard.vcard\colon VCard \tag{3.37}$$

$$SpatialAsset \sqsubseteq SpatialRole \tag{3.38}$$

$$SpatialAssetMetadata \sqsubseteq SpatialAsset \tag{3.39}$$

$$disjoint\left(Contributor, SpatialAsset\right) \tag{3.40}$$

$$disjoint\left(SpatialAssetMetadata, Actor \sqcup InterfaceRole\right) \tag{3.41}$$

A spatial role is a role related with spatial data. A contributor role describes the behaviour of an object that contributes and/or withdraws the assets, i.e. datasets or services, they own or control. A contribution is understood as a way to make some assets available to the users of an SDI, i.e. they are findable and there is a way to get or use them. It does not require the assets are free and it may be necessary to get a license from the contributor. A contributor may have a business card describing its name, address, organisation, telephone and email. A spatial asset is a role played by any useful or valuable spatial information resource that can be made accessible to the users of an SDI. Contributors and spatial assets are disjoint concepts. A spatial asset metadata is a type of spatial asset that provides information, i.e. a structured description about another spatial dataset or resource. By its nature, a spatial asset metadata is neither an actor nor an interface role.

$$\begin{aligned}&GetCapabilities \equiv Interaction{\sqcap} = 1\, hasRole\left(Initiator \sqcap \forall isPlayedBy.Agent\right) \sqcap\\ &\quad = 1\, hasRole.Responder{\sqcap} = 1\, hasRole.\left(Responder \sqcap SpatialAsset{\sqcap}\right.\\ &\quad \geq 1\, hasOperation.Operation \sqcap \forall name.\{"GetCapabilities", "Capabilities"\}{\sqcap}\\ &\quad \left.\forall isPlayedBy.SpatialService\right) \sqcap \exists hasRole.\left(Resource \sqcap SpatialAssetMetadata{\sqcap}\right.\\ &\quad \left.\forall isPlayedBy.OWSMetadata\right)\end{aligned} \tag{3.42}$$

The above concepts allow defining with precision the interactions that could happen in a system. The *GetCapabilities* interaction retrieves service metadata about the capabilities provided by any server that implements an OGC Web service interface specification. An agent initiates this interaction. The responder is played by a spatial service that becomes in a spatial asset. This spatial service participates in the interaction through an interface role with an operation named *GetCapabilities* (or *Capabilities* in WMS 1.0.0 (see Doyle et al., 2001)). The *GetCapabilities* interaction gives access to some service metadata, which is considered an essential resource in this interaction.

$$
\begin{aligned}
&GetDataSubset \sqsubseteq Interaction \sqcap = 1\, hasRole.Responder \sqcap \\
&\quad = 1\, hasRole\, (Initiator \sqcap \forall isPlayedBy.Agent) \sqcap = 1\, hasRole.\, (Responder \sqcap \\
&\quad SpatialAsset \sqcap InterfaceRole \sqcap \forall isPlayedBy.SpatialService) \sqcap \\
&\quad \exists hasRole.\, (Resource \sqcap SpatialAsset \sqcap \forall isPlayedBy.\, (SpatialDataset \sqcup Metadata)) \sqcap \\
&\quad \exists hasRole.\, (Artifact \sqcap SpatialAssetMetadata \sqcap \forall isPlayedBy.OWSMetadata)
\end{aligned}
\tag{3.43}
$$

The *GetDataSubset* interaction gives access to a subset or a gridded part of the data available in the server. When a service instance plays the role of service in this interaction, the service participates with operations named as *GetRecords*, *GetFeatures*, *GetFeatureInfo*, *GetCoverage* or *GetMap*. Although the name of the operation and the type of resource are different, these operations not change the basic semantics of get data subset. The essential resource in the interaction is a spatial dataset or metadata. The OGC Web service metadata plays the role of referenced artefact (e.g. the client uses the service metadata to find the endpoint of the operations). The *GetResourceById* interaction allows clients to retrieve one or more identified resources, including datasets (e.g. *GetRecordById*, *GetGmlObject*) and resources that describe datasets (e.g. *DescribeCoverage*, *DescribeRecord*, *DescribeFeatureType*) or parameters (e.g. *GetDomain*). The definition of *GetResourceById* is the same as *GetDataSubset* (see equation 3.43). The *Manage* interaction allows clients to manage the data served by the server. Typical manage actions are add, modify or delete a resource, lock a resource and harvest remote resources. The definition of *Manage* is the same as *GetDataSubset* (see equation 3.43).

$$
\begin{aligned}
&ContactServiceProvider \equiv Interaction \sqcap = 1\, hasRole\, (Initiator \sqcap \forall isPlayedBy.Agent) \sqcap \\
&\quad = 1\, hasRole.\, (Responder \sqcap Contributor \sqcap \forall isPlayedBy.Agent) \sqcap \exists hasRole.\, (Artifact \sqcap \\
&\quad \sqcap SpatialAsset \sqcap \forall isPlayedBy.\, (SpatialService \sqcap SpatialDataset \sqcap Metadata)) \sqcap \\
&\quad \exists hasRole.\, (Resource \sqcap SpatialAssetMetadata \sqcap \forall isPlayedBy.OWSMetadata)
\end{aligned}
\tag{3.44}
$$

The *ContactServiceProvider* is an interaction related with the contact of a client to the service provider about the service, the datasets or the metadata that describes both. The OGC Web service metadata plays an essential role as resource because provides the business address of the service provider.

The concepts *GetCapabilities*, *GetDataSubset*, *GetResourceById*, *Manage* and *ContactServiceProvider* are pairwise disjoint.

$$disjoint\,(GetCapabilities, GetDataSubset, GetResourceById, Manage,\\ ContactServiceProvider) \tag{3.45}$$

Policies. The `ServiceIndentification` section and the `Contents` sections of a service metadata can include optional statements about the fees and access constraints related to the behaviour of the system. Typically, `ServiceIdentification` contains statements about the access constraints that should be observed to assure the protection of privacy or intellectual property, and any other restrictions, fees or terms applicable on retrieving or using data from or otherwise using this service instance. The `Contents` section contains statements that apply only to part of the data. When these statements are no present, it is fair to assume that no access constraints, fees or terms are imposed. Reserved values, such as *none* (case insensitive), are used to express explicitly that the system does not have access constraints or fees.

The scope of the enterprise module includes the policies that can be described in the OGC Web service metadata of a system. The ontology shall be able to answer which policies applies to the OGW system. A *policy* is a named aspect of the specification or the constraints of a system's behaviour that can be changed to tailor a single system or that can evolve during the lifetime of the system. The choice in force at any particular instant is the *policy value*. In this sense, a service metadata defines the current policy values for the *access policy* and the *fees policy* on retrieving or using data from or using this service instance. Thus one might say that an object has an *access policy* with policy value *none*. The description of policy rules in *service metadata* is narrative. Thus, the definition of the rules in the conceptual model includes a description attribute used for display the rule to a human. Policies may apply to a *community* of objects as a whole, to particular objects that fulfil *roles* in a configuration, to instances of *roles* and to instances of *interactions*. A contract *owns* the policies that constraint the behaviour of the community of objects specified by the contract. That is, the policies have sense in the context of that community of objects for the purpose of the community expressed in the contract. The conceptual model adds two classes of policies, access policy and fees policy, with a none policy concept to provide a basis to model the statements about fees and access constraints found in *service metadata*.

The next concepts have been identified in the conceptualization and shall be formalized: *PolicyObject*, *Policy* and *PolicyValue*.

$$PolicyObject \sqsubseteq \forall specifies\,(Community \sqcap Interaction \sqcap Role \sqcap Object) \tag{3.46}$$

A policy object identifies the specification of behaviour, or constraints on behaviour, of communities, interactions, roles and objects.

$$Policy \sqsubseteq PolicyObject \sqcap = 1\, name.xs\colon string \sqcap \forall hasValues.PolicyValue \tag{3.47}$$

A policy is a set of rules related to a particular purpose. This set of rules can be named (e.g. *access*, *fees*).

$$PolicyValue \equiv \exists isValueOf.Policy \tag{3.48}$$

$$PolicyValue \sqsubseteq \forall isValueOf.Policy \sqcap \forall name.xs\colon string \sqcap \\ \leq 1\, name \sqcap \forall literalForm.rdf\colon PlainLiteral \tag{3.49}$$

$$disjoint\,(PolicyValue, Policy) \tag{3.50}$$

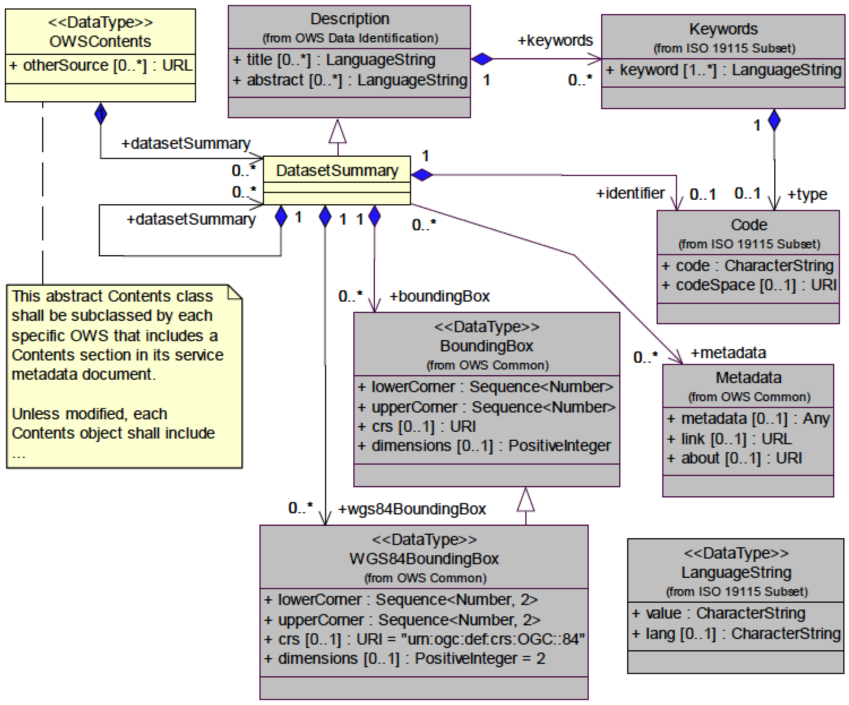
$$hasValue \equiv isValueOf^{-1} \tag{3.51}$$

A policy value is a policy that can be in force at some particular time. The policy value is disjoint from the concept policy. Policy values can be named (e.g. *none* for a none policy value). If the policy value is derived from an OGC Web service metadata document, the datatype property literal from can store the literal description of the policy value.

3.4.5 Information types

The focus of many OGC specifications is the definition of the content of the information that is being processed by geographic services and exchanged between geographic services and clients. The semantics of the information is defined in the specifications endorsed by OGC meanwhile the structure of the information is defined in information models developed by OGC. The semantics and the structure of the information depend on the type of geographic service. The OGC Service Architecture (OGC SA, Percivall, 2002) defines a basic taxonomy of geographic services (Table 3.8a). The most popular services today are geographic model/information management services (e.g. WMS, WMTS, WFS, WCS, CSW, SOS). These services are characterized by providing access to geographic datasets or metadata and conceptual schemas related to geographic datasets. This access can take the form of a geographic image in WMS and WMTS. Model/information management services are characterized by the type of content acceded (Table 3.8b).

As the OGC Web services Common Specification (Whiteside, 2007) explains, the *service metadata* of a *service instance* may contain a `Contents` section with metadata about the data served by the service instance (Figure 3.14). This metadata describes the type of content that can be returned by the service after processing the original dataset. The type of content acceded should not to be confused with the underlying information model of any of the *datasets* accessed through the *service instance*. For example, given a *dataset*, a feature access service may describe in the `Contents` section



<<DataType>>
OWSContents
+ otherSource [0..*] : URL
Description
(from OWS Data Identification)
+ title [0..*] : LanguageString
+ abstract [0..*] : LanguageString
+keywords
1
0..*
Keywords
(from ISO 19115 Subset)
+ keyword [1..*] : LanguageString
+datasetSummary
0..*
0..*
DatasetSummary
1
+datasetSummary
1
1 1
0..*
+identifier
0..1
0..1
+type
Code
(from ISO 19115 Subset)
+ code : CharacterString
+ codeSpace [0..1] : URI
This abstract Contents class shall be subclassed by each specific OWS that includes a Contents section in its service metadata document.
Unless modified, each Contents object shall include ...
0..*
+boundingBox
<<DataType>>
BoundingBox
(from OWS Common)
+ lowerCorner : Sequence<Number>
+ upperCorner : Sequence<Number>
+ crs [0..1] : URI
+ dimensions [0..1] : PositiveInteger
0..*
+metadata
Metadata
(from OWS Common)
+ metadata [0..1] : Any
+ link [0..1] : URL
+ about [0..1] : URI
0..*
+wgs84BoundingBox
<<DataType>>
WGS84BoundingBox
(from OWS Common)
+ lowerCorner : Sequence<Number, 2>
+ upperCorner : Sequence<Number, 2>
+ crs [0..1] : URI = "urn:ogc:def:crs:OGC::84"
+ dimensions [0..1] : PositiveInteger = 2
<<DataType>>
LanguageString
(from ISO 19115 Subset)
+ value : CharacterString
+ lang [0..1] : CharacterString

Figure 3.14: `OWSContents` section UML class diagram (pg. 35, Whiteside, 2007).

the different feature types that can be requested to the service. The feature type characterizes exclusively the information model of each of the feature instances returned by the service instance. The accessible content to the same dataset through a web map service is described in the `Contents` section as layers. The layer characterizes the properties of each map returned by the service portraying the dataset contents.

The types of information exchanged between a user and an OWS system compose the scope of the module. The ontology shall be able to answer which types of information are exchanged.

The behaviour of a system depends on a shared understanding of the information items that the objects that participate in the system communicate when they play roles in the interactions. Thus, this shared understanding of the information in an interaction is modelled in our conceptual model as *information types* associated with the behaviour or the state of the involved object when plays the role. An *information type* is a set of predicates characterizing a collection of information objects relevant in the system. The OGC *service metadata* can identity the *information model* where the *information type* is defined. An *information model* is a set of predicates that describe the constraints (*invariant schema*), the configuration (*static schema*) and the behaviour (*dynamic schema*) of the information objects to which it applies. The characteristics and the standardization of OGC information models are discussed in *Implemented Standards* (Section 3.4.8).

The conceptual model defines a minimal set of attributes and relations that an *information type*

Table 3.8: Taxonomy of relevant OGC Web services identified in the OGC SA (Percivall, 2002) and its content types.

(a) Taxonomy of Geographic services in OGC SA.

Geographic service	**Description**	**Example**
Human interaction services	Management of user interfaces, graphics, multimedia, and for presentation of compound documents	Map access service
Model/Information management services	Management of the development, manipulation, and storage of metadata, conceptual schemas and datasets	Map access service, Feature access service, Coverage access service, Sensor access service, Catalogue service
Workflow/Task management services	Support of specific tasks or work-related activities conducted by humans	Sensor Planning Service
Processing service	Perform large-scale computations involving substantial amounts of data	Processing Service
Communication services	encoding and transfer of data across communications networks	Messaging Service
Management system	Management of system components, applications and networks	None identified

(b) Examples of geographic model/information management services and its main content type.

Model/information service	**Implementation standard**	**Content type**
Feature access service	WFS	Feature type: information about real word phenomena that may be requested as structured content.
Map access service	WMS, Web Map Tile Service (WMTS, Masó et al., 2010)	Layer: basic unit of geographic information that may be requested as a map, i.e. encoded as image.
Coverage access service	WCS	Coverage: Feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain.
Sensor access service	SOS	Observation: It is a way to organize sets of sensor observation groupings.
Catalogue service	CSW	Record: Records that conform to the schema of a metadata information model. Described in the OperationsMetadata section.

should have: *identifier*, *title*, *abstract*, *keywords*, *bounds* (spatial and temporal), *metadata*, and a reference to its *information schema*. Information types can be logically aggregated (*information aggregate*) for forming hierarchies in some services (e.g. map access services). An instance can be the same time an *information aggregate* and an *information type*, that is, it can be composed of information types and can have an identifier. This minimal model is based in the contents of the `DatasetSummary` described in the common specification (Whiteside, 2007), and the organization of geographic information in datasets and aggregation datasets of ISO 19115 (ISO/TC 211, 2003b).

The next concepts have been identified in the conceptualization and shall be formalized: *InformationType*, *InformationModel. InformationAggregate*, *Layer, FeatureType, Coverage, Observation, Record, TemporalEntity,* and *GeographicEntity.* The concepts *Operation* and *InformationType* are defined later. The concept *InformationModel* is defined later. The concept *TemporalEntity* is imported from the Time Ontology, and represents time. The concept *GeographicEntity* is imported from the Geo-Net Ontology described in Chapter 4, and represents space.

$$\begin{aligned} & InformationType \sqsubseteq \forall name.xs\colon string \sqcap =1\, name \sqcap \\ & \quad \leq 1\, isSpecifiedBy.InformationModel \sqcap \forall hasKeywords.ExtendedLiteral \sqcap \\ & \quad \forall title.rdf\colon PlainLiteral \sqcap \forall abstract.rdf\colon PlainLiteral \sqcap \\ & \quad \forall coverage.\left(tm\colon TemporalEntity \sqcup gn\colon GeographicEntity\right) \end{aligned} \tag{3.52}$$

An information type is a named set of predicates characterizing a collection of information objects relevant in the system. The information type may be populated with descriptive information found in an OGC Web service metadata, that is, title, abstract, keywords and spatial and temporal coverage. An information model may specify the information type.

$$\begin{aligned} & InformationAggregate \sqsubseteq \forall name.xs\colon string \sqcap \leq 1\, name \sqcap \\ & \quad \forall hasKeywords.ExtendedLiteral \sqcap \forall title.rdf\colon PlainLiteral \sqcap \\ & \quad \forall abstract.rdf\colon PlainLiteral \sqcap \exists composedOf.\top \sqcap \forall composedOf.InformationType \sqcup \\ & \quad \exists subset.\top \sqcap \forall subset.InformationAggregated \sqcup \exists superset.\top \sqcap \\ & \quad \forall superset.InformationAggregated \end{aligned} \tag{3.53}$$

$$disjoint\left(InformationType, InformationAggregate\right) \tag{3.54}$$

Information aggregates is a way to organize a set of information types into a hierarchy. This concept is disjoint from the information type.

$$Record \sqsubseteq InformationType \tag{3.55}$$

$$Layer \sqsubseteq InformationType \sqcap \exists coverage.gn\colon GeographicEntity \tag{3.56}$$

$$Coverage \sqsubseteq InformationType \sqcap \exists coverage.gn\colon GeographicEntity \tag{3.57}$$

$$FeatureType \sqsubseteq InformationType \sqcap \exists coverage.gn\colon GeographicEntity \tag{3.58}$$

$$Observation \sqsubseteq InformationType \sqcap \exists coverage.gn\colon GeographicEntity \tag{3.59}$$

$$disjoint\,(Record, Layer, Coverage, FeatureType, Observation) \tag{3.60}$$

In order to detail the representation of different management services, the ontology include the concepts record, layer, coverage, feature type and observation. These concepts are pairwise disjoints.

3.4.6 Operations

The OGC Service Architecture provides a framework for that enables users to access and process geospatial data across generic computing interfaces. OGC standards are defined for multiple distributed computing platforms while maintaining common geospatial semantics across the underlying technology. OGC standards define *services*, *interfaces* and *operations*. A *service* is a distinct part of the functionality that is provided by an entity through interfaces. An *interface* is a set of operations that characterize the behaviour of an entity. An *operation* is 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. Each OGC service metadata describes that an entity can play the *role* of entity that enables to access and process geospatial data in an *interaction*. Additionally, the OGC service metadata asserts in the `OperationsMetadata` section that this entity supports a number of named operations that are conform to the specification of an *interface* when plays the role of spatial service (Figure 3.15).

The scope of the module is the specification of the actions that can be called to execute in a system. The ontology shall be able to answer which operations are supported by a system and its signature, which are the logical address of the operations of the system in each DCP, and which are the interaction style, i.e. how the requests are send.

An *interface* is an abstraction of the *behaviour* of an object that consists of a subset of the interactions of that object together with a set of constraints on when they can occur. Each interface considers only the interactions of that interface hiding all other interactions. The part of the behaviour identified by an interface is modelled by an *interaction* that has associations with each of the roles that express the *interface role* of the object on the one hand and the *roles* that identifies the behaviour of the external objects on the other. The interface role should be specified by an implementable standard or equivalent specification. Specifications are discussed in *Implemented standards* (Section 3.4.8). A specification in this section acts as source of constraints to interfaces

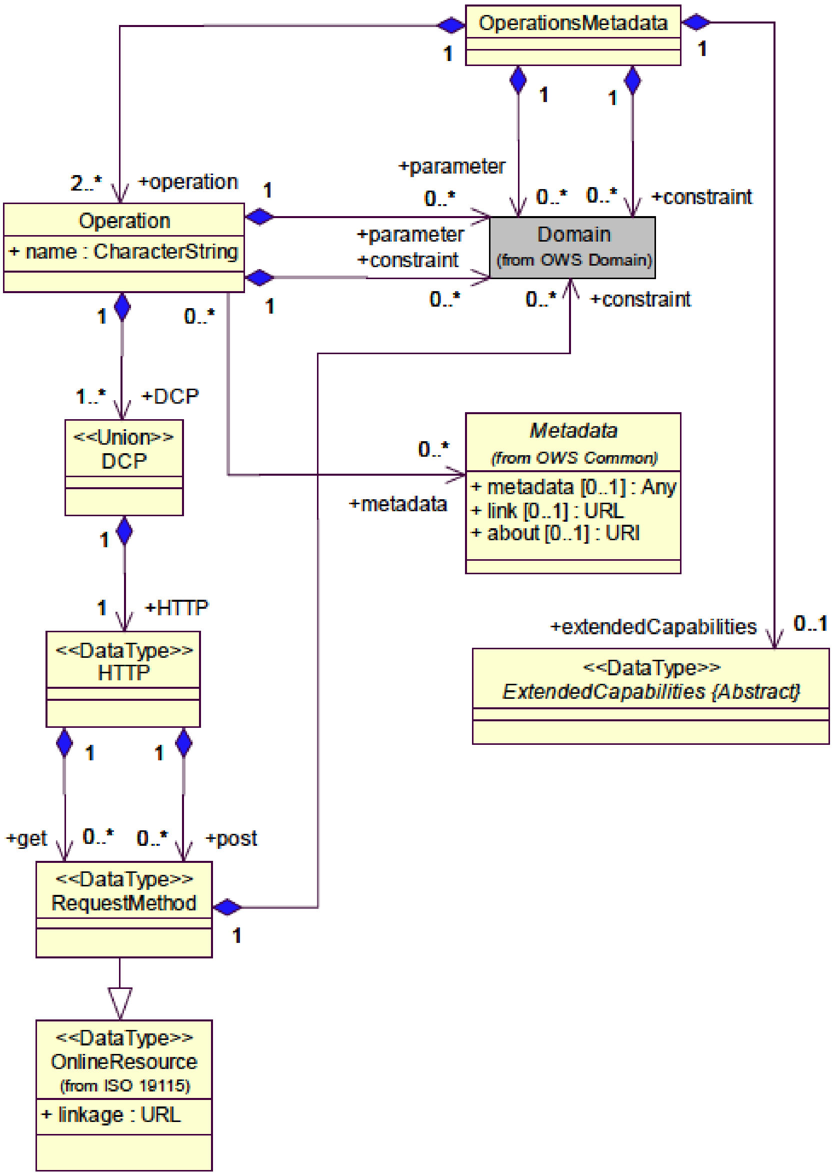


Figure 3.15: `OperationsMetadata` section UML class diagram (pg. 30, Whiteside, 2007).

and operations.

An interface role has a number of operations. An *operation* is a named interaction between a client object and a service object consisting of an invocation and optionally a termination. An ordered sequence of message types exchanged between clients and services can describe the interaction. The common specification (Whiteside, 2007) uses the term *message* to identify the content of requests, responses and exceptions in different bindings and encodings, and allows specifying operations that only accept requests encoded as a message. The term *message* is used in other approaches to the formal modelling of interfaces, such as Pedrinaci et al. (2010). Subsequently, the conceptual model the conveyance of information as *message types* that are equivalent to the operation signature. A message that flows from the client object to the service object is named *input message*. A message originated in a service object is named *output message*. A message can be also named *exception message* if they are send as a consequence of some fault in the service object or in the client object.

Each message may carry a *typed structured payload* (e.g. the image that a operation *GetMap*). The payload can be decomposed in other logical parts (e.g. the parameters of the operation). The allowed *literal* values for each logical part can be explicitly defined. These values can be associated with other object in the system when the value in the part acts as a surrogate of those objects. For example, the parameter LAYERS in the operation *GetMap* represents map layers as lists the map layers returned by a *GetMap* request; the allowed values are the identifiers of the layers that acts as surrogate of the layers in this context. The description of a message can be completed with information extracted from the interface specification. This information includes messages and *mandatory* parts not explicitly included in the available service metadata. An optional part can be modelled as specialization of the relation between a message and its parts. Some parts are conditional, i.e. they require or exclude the presence of other parts in the message. Conditional parts can be modelled as parts that appear only in messages found in some sequence of interactions.

The next concepts have been identified in the conceptualization and shall be formalized: *Operation*, *Binding, Message* and *Payload*. The concept *Binding* is defined later.

$$\begin{aligned}
&Operation \equiv \forall name.xs\colon string \sqcap = 1\, name \sqcap \\
&\quad \exists hasFirst.Message \sqcap \forall hasMessage.Message \sqcap \\
&\quad \forall hasBinding.Binding && (3.61) \\
&hasFirst \sqsubseteq hasMessage && (3.62) \\
&hasInput \sqsubseteq hasMessage && (3.63) \\
&hasOutput \sqsubseteq hasMessage && (3.64) \\
&hasException \sqsubseteq hasMessage && (3.65)
\end{aligned}$$

An operation is a named interaction between a client object and a service object consisting of an invocation and optionally a termination. The first invocation is represented by the role has first. The operation may be bound to a DCP. The different kinds of messages that flows using the operation are classified with the roles *has input* that represents flows from the client to the server, *has output* that represents flows form the server to the client and *has exception* that qualifies an input or output message.

$$Message \sqsubseteq \forall hasPayload.Payload \sqcap \leq 1\, hasPayload \sqcap \forall hasNext.Message \tag{3.66}$$

$$hasMessage \sqsupseteq hasMessage \circ hasNext \tag{3.67}$$

The message conveys the information between the client and the server carrying a payload. The order of messages is made explicit with the role *has next*. If an operation has a relation to a message through a property chain formed by the roles *has message* and *has next*, the operation is also related with the message through the role *has message*.

$$\begin{aligned} Payload \sqsubseteq\ & \forall name.xs\colon string \sqcap \leq 1\, name \sqcap \forall allowedType.InformationType \sqcap \\ & \forall allowedValue.\,(\exists name.xs\colon string \sqcup \exists namespace.xs\colon anyURI \sqcup \\ & \quad ExtendedLiteral) \sqcap \forall composedOf.Payload \end{aligned} \tag{3.68}$$

$$mandatoryPart \sqsubseteq composedOf \tag{3.69}$$

A payload may be named (e.g. an input parameter), typed (e.g. the return value), structured (e.g. a set of parameters of a request) and constrained (e.g. the allowed values in an input parameter). Mandatory parts of a message are signalled with the role mandatory part.

3.4.7 Distributed platform bindings

The scope of the module is the identification of the mechanisms for distributed computing useful identified in the OGC specifications. The ontology shall be able to answer which DCP are used by the system, which are the logical address of the operations of the system in each DCP, and which are the interaction style, i.e. how the requests are send.

One of the goals of the OGC Service Architecture Abstract Specification (Percivall, 2002) is the development of a service architecture that can use DCP for service interaction in the geospatial context. The OGC Web Services Common Specification (OWS CS, Whiteside, 2007) defines a binding to the DCP formed by the Internet hosts that offers its resources as remote procedure calls (RPC). Specific implementation standards can specify additional DCP, how to assert the logical addresses of the operations, and how to use the DCP to send interaction request. For example, the WMTS 1.0 standard (Masó et al., 2010) defines a binding in the so-called RESTful style (Richardson and Ruby, 2007), and the WFS 2.0 standard (Vretanos, 2010) defines a binding to a message based DCP where the messages are expressed in XML using the Simple Object Access Protocol (SOAP, see Curbera et al., 2002).

A distributed interaction consists of multiple autonomous objects that reside in distributed nodes that communicate through a network. Given a DCP, the communication between distributed objects is made through a virtual channel between platform endpoints that provide access to the distributed object interfaces. Once the channel is established, client object and service objects *marshal* its requests into information conveyed by the channel. Marshalling is the process of transforming the memory representation of an object to a data format suitable for storage or transmission. It simplifies complex communication, using platform objects to communicate. The reverse of marshalling is unmarshalling.

On the other side, the messages are unmarshalled into operation requests and operation responses. The description of the mechanisms required to support distributed interaction is simplified under the assumption that few DCPs are relevant for OGC Web services (and Web services in general). Given a well-known transport *platform* (e.g. Internet hosts that support HTTP), a basic but complete description requires only the description of the bindings between the operations of an object and a platform for the purpose of distributed interaction. A *binding* describes a concrete message format or *encoding schema* (e.g. KVP) and the transmission protocol in a *platform* that is applied to one or several operations[5]. The binding can define (or share) an *endpoint* or port that defines a logical address or range of logical addresses in the naming context of the platform for the binding. The endpoints thus are the alternate places where the operations, and subsequently the interface, are provided. The message format or encoding are the rules required to marshal and unmarshal messages. Some platform bindings require additional information. For example, a binding to the DCP comprising Internet hosts that support HTTP (*HTTP platform*) requires the specification of the HTTP verb or method (e.g. GET, POST) used to establish the communication channel.

The next concepts have been identified in the conceptualization and shall be formalized: *Binding, HttpBinding, Platform, Endpoint* and *Encoding.*

$$\begin{aligned} \mathit{Binding} \equiv\; & \exists usesEndpoint.\top \sqcap \forall usesEndpoint.Endpoint \sqcap \exists withEncoding.\top \sqcap \\ & \forall withEncoding.Encoding \sqcap = 1\, toPlatform \sqcap \forall toPlatform.Platform \end{aligned} \tag{3.70}$$

A binding component describes given a platform, a collection of endpoints and concrete message formats that can be used to communicate through the platform. The concept encoding is an atomic

[5] The terminology used here is closer to WSDL 2.0 (Chinnici et al., 2007) and OGC (Whiteside, 2007) terminologies. In ODP terms, a *binding* is a channel. Our "binding" describes the participation of an object in a *channel*, provides an identifier to the object for the purpose of interaction in the *channel*, and instruct the *stub* of the object in the channel how to interpret the interactions conveyed by the channel, and performs any necessary message transformations based on this interpretation.

concept. The ontology defines instances for the concepts that identify encodings used in OGC Web service standards: KVP (see Whiteside, 2007), XML (see Whiteside, 2007), SOAP (see for instance in Vretanos, 2010) and RESTful (see for instance in Masó et al., 2010).

$$\begin{aligned}&HttpBinding \equiv Binding \sqcap \forall toPlatform.\{PlatformHTTP, PlatformREST\} \sqcap \\ &\quad \forall verb.\{"GET", "POST", "PUT", "DELETE\} \sqcap = 1\, verb\end{aligned} \tag{3.71}$$

The HTTP binding allows specifying the HTTP verb used to establish the communication channel in platforms based in HTTP that requires the specification of the HTTP verb. The above formula introduces the instances *PlatformHTTP* and *Platform REST*. *PlatformHTTP*, *PlatformREST* alongside with *PlatformSOAP* are instances defined in the ontology for the representation of the following DCPs:

- *PlatformHTTP*. Internet hosts that support HTTP interactions in a RPC style. This is the standard DCP in OGC Web services (see Whiteside, 2007).

- *PlatformSOAP*. Internet hosts that exchange SOAP messages (Curbera et al., 2002). Bindings to this platform are available in recent OGC Web service standards, such as WFS 2.0 (Vretanos, 2010).

- *PlatformREST*. Internet hosts that support HTTP interactions oriented to resources or REST (Fielding, 2000). The concept REST is discussed in chapter 4. Bindings to this platform are available in recent OGC Web service standards, such as WMTS (Masó et al., 2010).

$$Endpoint \sqsubseteq \exists uriTemplate.xs\colon string \sqcap = 1\, uriTemplate \tag{3.72}$$

An endpoint defines the particulars of a specific endpoint at which a given service is available in a binding. The endpoints thus are in effect alternate places at which the service is provided. The OGC Web service metadata provides a URL that acts as the prefix of the effective URI of the endpoint. For example, if the endpoint for a *GetCapabilities* operation for a WCS is `http://www.example.com/service?dummy=param&` and the encoding used is KVP the effective request is `http://www.example.com/service?dummy=param&request=GetCapabilities&service=WCS`, where the order and the case of *request* and *service* is ignored. The server should be able to identify the later URI as a request to the endpoint defined by `http://www.example.com/service?dummy=param&`. This situation is captured in the ontology by means of the datatype role URI template.

3.4.8 Implementable standards and information models

OGC has developed several documents and information models to address different kinds of interoperability challenges since 1994. OGC publishes *Implementation Standards*, *Abstract Specifications*, *Best Practices*, *Discussion Paper*s and *White Papers.* The standard baseline of OGC consists of *Implementation Standards*, *Abstract Specifications* and *Best Practices.* An *Abstract Specification* describes a platform-independent model for an application environment for geospatial data and service products and interoperable geoprocessing. An *Implementation Standard* describes a platform-dependent *interface*, an *encoding*, an *application schema,* or a *profile* based on an *Abstract Specification* or a domain extension to an *Abstract Specification.* A *Best Practices* document contains discussion about the use or the implementation of a baseline document. The development of an *Implementation Standard* or a *Best Practices* documents often result in an *information model*, usually in the form of UML models and *XML Schema* documents. *Implementation Standards* and *Abstracts Specifications* documents are documents containing a consensus within OGC. *Best Practices* documents are an endorsement of the OGC of the content of the document. *Discussion Papers* are not endorsed by OGC. *White Papers* states a position of OGC on a subject. OGC Web services are defined using non-proprietary Internet standards, such as HTTP, URLs, MIME types and XML. Recently, OGC Web services are becoming defined using enterprise web service standards (WSDL, SOAP). OGC has endorsed specifications not developed within OGC working groups, such as GeoRSS (Reed, 2006) and KML (Wilson, 2008).

The focus of the module is the documents that form the standard baseline of OGC standards and its information model. The ontology shall be able to answer which standards and information models that a system that implements OGC Web services supports.

Users may describe an implementation of a system based in OGC Web services in terms of objects (hardware, software, and network products) that are conforming to *implementable standards.* An *implementable standard* is a specification in sufficient detail of the common terms, concepts, encoding and techniques of shared or required by a collection of objects. The next concepts, relations and attributes can be found in OGC service metadata documents and shall be represented in the ontology. OGC *implementable standards* are named resources, sometimes with a *version* number, whose digital representations can be located elsewhere. An implementable standard can replace previous implementable standards. Implementable standards can have profiles. A *profile* consists of an agreed-upon subset and interpretation of a specification. An implementable standard can define several representations of these information items as concepts, relationships, constraints, rules, and operations. This representation is an *information model* of the implementable standard. Information models can be related each other by means of *importing* and *merging* models. *Information models* often takes the form of *XML Schema* documents and UML model which are maintained by OGC in an official XML schema repository[6].

[6] `http://schemas.opengis.net/`

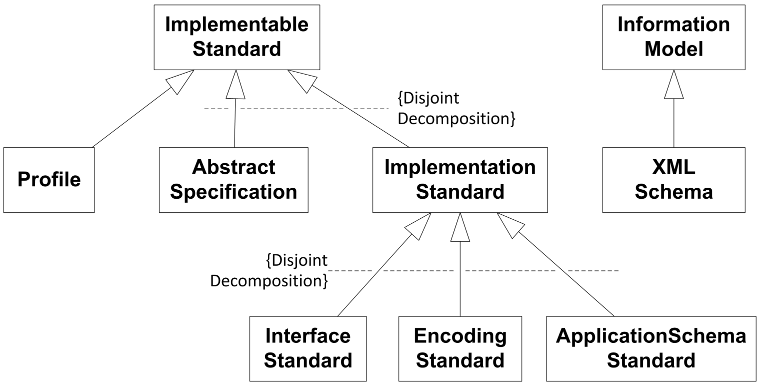


Figure 3.16: Taxonomy of implementable standards and information models.

An object can *conform* to an implementable standards and information models. Conformance connotes a similarity to the implementable standard and information model within some allowed range. An object may be *compliant* with an implementable standard or an information model if follows these in a very specific and verifiable way. In the case of OGC implementable standards, compliance means that the object passes the abstract tests established in the standard.

The concept model includes the two types of implementable standards that exist in OGC: *abstract specifications* and *implementation standards*. An abstract specification describes platform-independent properties of the objects, that is, the conceptual foundation. Meanwhile, an implementation standard defines platform-dependent properties. An implementation standard should be based in abstracts specification. Although the same document can contains an abstract specification and several implementation standards (e.g. the OGC Catalogue Service Nebert et al. (2007)), the concepts *abstract specification* and *implementation standard* shall be disjoint concepts.

The ontology shall include three disjoint subtypes of the concept *implementation standard*: *interface standards*, *encoding standards* and *application schema standards.* The concepts interface standard and encoding standard are auto explicative. An *application schema* is a term used in OGC to identify a domain or specific XML Schema. An *application schema standard* defines a set of XML schemas, policies, and guidelines defined for a particular application or domain. Figure 3.16 presents the class hierarchy of this module derived from theses guidelines.

The next concepts have been identified in the conceptualization and shall be formalized: *ImplementableStandard, Profile, AbstractSpecification*, *ImplementationStandard*, *EncodingStandard*, *ApplicationSchemaStandard*, *InformationModel* and *XMLSchema.*

$$ImplementableStandard \sqsubseteq \forall replaces.ImplementableStandard \sqcap \forall name.xs: string \sqcap$$

$$= 1\, name \sqcap \forall version.xs\colon string \sqcap = 1\, version \sqcap \forall location.xs\colon anyURI \sqcap$$
$$\exists location.xs : anyURI \tag{3.73}$$

An implementable standard is a named and versioned specification in sufficient detail of the common terms, concepts, encoding and techniques of shared or required by a collection of objects.

$$AbstractStandard \sqsubseteq ImplementableStandard$$
$$ImplementationStandard \sqsubseteq ImplementableStandard \sqcap \forall defines.InformationModel \sqcap$$
$$\exists isSpecifiedBy.\top \sqcap \forall isSpecifiedBy.AbstractStandard \tag{3.74}$$
$$disjoint\,(AbstractStandard, ImplementationStandard) \tag{3.75}$$

An abstract standard is a document containing platform independent OGC consensus, technology dependent standard for application programming interfaces and related standards. An implementation standard is a document containing platform dependent OGC consensus, technology dependent standard for application programming interfaces and related standards derived from an abstract standard.

$$Profile \equiv ImplementableStandard \sqcap \forall profileOf.\top \sqcap$$
$$\exists profileOf\,(AbstractStandard \sqcup ImplementationStandard) \tag{3.76}$$

A profile consists of an agreed-upon subset and interpretation of an abstract standard or implementation standard.

$$InterfaceStandard \sqsubseteq ImplementableStandard \sqcap$$
$$\forall specifies\,(OWSServiceMetadata \sqcup OWSContract) \tag{3.77}$$
$$EncodingStandard \sqsubseteq ImplementableStandard \sqcap \forall specifies.InformationType \tag{3.78}$$
$$ApplicationSchemaStandard \sqsubseteq ImplementableStandard \sqcap$$
$$\forall specifies.InformationType \sqcap \exists defines.\top \sqcap \forall defines.XMLSchema \tag{3.79}$$
$$disjoint\,(InterfaceStandard, EncodingStandard, ApplicationSchemaStandard) \tag{3.80}$$

The above equations define the properties required for specifying interface standards, encoding standards and applications schema standards.

$$
\begin{aligned}
&InformationModel \sqsubseteq \forall replaces.InformationModel \sqcap \\
&\quad \forall namespace.xs\colon string \sqcap = 1\, namespace \sqcap \forall version.xs\colon string \sqcap \leq 1\, version \sqcap \\
&\quad \forall location.xs\colon anyURI \sqcap \exists location.xs : anyURI \sqcap \\
&\quad \forall import.InformationModel \sqcap \forall merge.InformationModel && (3.81)\\
&Trans\,(import) && (3.82)\\
&Trans\,(merge) && (3.83)\\
&import \sqsupseteq merge \circ import && (3.84)
\end{aligned}
$$

An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse.

$$
\begin{aligned}
&XMLSchema \sqsubseteq InformationModel \sqcap \forall import.XMLSchema \\
&\quad \sqcap \forall merge.XMLSchema && (3.85)
\end{aligned}
$$

In addition, the implementation of the ontology should include a defined set of instances that represents the official implementation standards and information models used in OGC, including deprecated models.

3.5 A service in OntoOWS

This section contains an example of the application of OntoOWS for the description of a service from its service metadata document. The running example is a service instance that implements the WMS Interface Standard (IS) version 1.3.0. The next URL is an example of a *GetCapabilities* request encoded in KVP in a HTTP GET request with values appropriate for a WMS 1.3.0 service instance:

```
http://www.idee.es/wms/WMS-Cantabria/WMS-Cantabria?REQUEST=
GetCapabilities&SERVICE=WMS&ACCEPTVERSION=1.3.0
```

Below is the structure of the service metadata document that can be retrieved:

```
<WMS_Capabilities version="1.3.0" ...>
 <Service>
    ...
    <ContactInformation>...</ContactInformation>
```

```
      ...
    </Service>
    <Capability>
      <Request>...</Request>
      ...
      <Layer ...>
         ...
      </Layer>
    </Capability>
  </WMS_Capabilities>
```

Although the structure differs syntactically from the structure presented in Section 3.4.1, the semantics are the same. The section `Service` contains the service identification (`ServiceIdentification`). The section `ContactInformation` contains the service provider description (`ServiceProvider`). The section `Request` contains metadata about operations (`OperationsMetadata`). Finally, the section `Layer` contains the contents for this kind of service (`OWSContents`).

OGC Web service communities and contracts. In a WMS 1.3.0, the root tag and the section `Service` provides information to identify the service as WMS. The version is found as an attribute of the root tag ("1.3.0"). See the example below:

```
<WMS_Capabilities version="1.3.0"
  xmlns="http://www.opengis.net/wms" ...>
  <Service>
     <Name>WMS</Name>
  </Service>
  ...
</WMS_Capabilities>
```

Once the XML document is identified as a valid OGC Web service metadata document, the extraction of information can start. The extraction of information can begin with the definition of individuals that represent the service metadata, the contract, the community, the spatial service and the spatial dataset accessed.

```
OWSMetadata(cantabria_sm)
OWSContract(cantabria_cnt)
OWSCommunity(cantabria_com)
SpatialService(cantabria_srv)
SpatialDataset(cantabria_data)
```

Next, the description is supplemented with statements about properties found in the service metadata that describe the service metadata and the community

```
location(cantabria_sm, "http://www.idee.es/...")
serviceType(cantabria_sm, "WMS")
serviceTypeVersion(cantabria_sm, "1.3.0")
title(cantabria_com, "WMS-Cantabria")
```

Finally, the individuals can be related. That is, the service metadata describes the contract, the contract specifies the community, and the community is composed by a spatial service, a spatial dataset and the service metadata.

```
describes(cantabria_sm, cantabria_cnt)
specifies(cantabria_cnt, cantabria_com)
composedOf(cantabria_com, cantabria_sm)
composedOf(cantabria_com, cantabria_srv)
composedOf(cantabria_com, cantabria_data)
```

Interactions and roles. The service metadata of the running example provides metadata that allows several interactions. For example, the section `ContactInformation` in `Service` provides a business card of the service provider and enables the contact with the service provider. Below, the section `Request` describes the availability of three operations (*GetCapabilities*, *GetMap* and *GetFeatureInfo*).

```
<Request>
  <GetCapabilities>...</GetCapabilities>
  <GetMap>...</GetMap>
  <GetFeatureInfo>...</GetFeatureInfo>
</Request>
```

For example, the *GetMap* operation manifests a *GetDataSubset* interaction. Hence, the definition of this interaction as individual is added to the description of the service, and then, the interaction is related with the service instance, the service metadata and the spatial dataset defined above.

```
GetDataSubset(cantabria_gds)
hasRole(cantabria_gds, cantabria_gds_ir)
isPlayedBy(cantabria_gds_ir, cantabria_srv)
hasRole(cantabria_gds, cantabria_gds_data)
isPlayedBy(cantabria_gds_data, cantabria_data)
hasRole(cantabria_gds, cantabria_gds_sm)
isPlayedBy(cantabria_gds_sm, cantabria_sm)
```

The interpretation of the above statements depends on the definition of this interaction in equation 3.43 and the statements about the objects identified by *cantabria_srv*, *cantabria_data* and *cantabria_sm*. The statements that involve the role *cantabria_gds_ir* say that the service instance is a responder in the interaction, a spatial asset, and it interacts with the environment with an interface with at least an operation. The statements that involve the role *cantabria_gds_data* say that the spatial dataset is a spatial asset, an essential resource in the interaction and may be associated with an information type. Finally, the statements that involve the role *cantabria_gds_sm* say that the service metadata is an artefact referenced in the interaction and, for this interaction, is the metadata of a spatial asset.

All the defined interactions and roles are owned by the contract:

```
owns(cantabria_cnt, cantabria_gds)
owns(cantabria_cnt, cantabria_gds_ir)
owns(cantabria_cnt, cantabria_gds_data)
owns(cantabria_cnt, cantabria_gds_sm)
```

Policies. The Service section of a WMS 1.3.0 contains information about the service fees. The running example says that the use of this service is free ("gratuito" in Spanish) but it is not allowed its use in value added services ("no está permitido(...)" in Spanish).

```
<Service>
  ...
  <Fees>gratuito</Fees>
  ...
</Service>
```

The example contains a policy value of a fees policy whose description is "gratuito". This can be described with OntoOWS as follows:

```
Policy(fees)
name(fees, "Fees")
PolicyValue(cantabria_fees)
literalForm(cantabria_fees, "gratuito")
hasValue(fees, cantabria_fees)
specifies(cantabria_fees, cantabria_com)
owns(cantabria_cnt, cantabria_fees)
```

The above statements say that there is a policy named *Fees* with a value *gratuito* owned by the contract that specifies the behaviour of the community.

Information types. Below is an excerpt of the Layer section of the service metadata:

```
<Layer noSubsets="1">
    <Title>WMS-Cantabria</Title>
    ...
    <Layer>
       <Name>Topografia5000</Name>
       <Title>Topografía 5000</Title>
       <BoundingBox CRS="EPSG:23030"
          minx="349347.31" miny="4734326.55"
          maxx="488182.33" maxy="4818234.60"/>
       ...
    </Layer>
  </Layer>
```

This snippet identifies a named layer (a layer with `Name` tag) and a category layer (a layer without `Name` tag). Category layers help users to arrange layers into hierarchies. The named layer is defined as individual of the information type class *Layer*.

```
Layer(cantabria_topografia5000)
InformationAggregate(cantabria_wms_cantabria)
```

Named layers can be accessed using *GetMap* operations. The operation *GetMap* manifested the *GetSubsetData* interaction and then the *cantabria_gds_data* role. Therefore, the layer is related with the *GetSubsetData* interaction through the *cantabria_gds_data* role as follows:

```
hasType(cantabria_gds_data, cantabria_topografia5000)
```

Then, the properties of the layer can be described. Simple properties as name and title are covered by OntoOWS. Complex properties, such as the bounding box, are described using the Geo-Net vocabulary.

```
name(cantabria_topografia5000, "Topografia5000")
title(cantabria_topografia5000, "Topografía 5000")
gn:GeographicEntity(cantabria_topografia5000_bbox)
coverage(cantabria_topografia5000, cantabria_topografia5000_bbox)
```

The category layer can also be described:

```
title(cantabria_wms_cantabria, "WMS-Cantabria")
composedOf(cantabria_wms_cantabria, cantabria_topografia5000)
```

Operations. The Request section contains the definition of the operations.

```
<GetMap>
   <Format>image/png</Format>
    ...
</GetMap>
```

The definition of an operation is straightforward.

```
Operation(cantabria_getmap)
name(cantabria_getmap, "GetMap")
```

However, the interface role that gives sense to this operation requires to be defined. The analysis of the possible interactions has related the *GetMap* operations with *GetDataSubset* interactions. Thus, it is possible to assert:

```
hasOperation(cantabria_gds_ir, cantabria_getmap)
```

Each definition may describe the allowed parameters and its values. For example, in the running example, the *GetMap* operation allows the value *image/png* as format. The parameter layer, defined in the standard, allows identifying the named layers that are requested. The *GetMap* request can be defined as a message whose payload is composed by a part named layer whose allowed value is the layer defined above, and a part named format whose allowed value is a extended literal. This allows to late binding the extended literal to a more precise definition (e.g. a MIME type ontology).

```
hasFirst(cantabria_getmap, cantabria_getmap_req)
hasInput(cantabria_getmap, cantabria_getmap_req)
hasPayload(cantabria_getmap_req, cantabria_getmap_kvp)
composedOf(cantabria_getmap_kvp, cantabria_getmap_layer)
name(cantabria_getmap_layer, "LAYER")
allowedValue(cantabria_getmap_layer, cantabria_topografia5000)
composedOf(cantabria_getmap_kvp, cantabria_getmap_format)
name(cantabria_getmap_format, "FORMAT")
ExtendedLiteral(image_png)
literalForm(image_png, "image/png")
allowedValue(cantabria_getmap_format, image_png)
```

The typical response is a graphical image whose meaning is defined by one of the allowed layers. Thus, the request can be related with the response, and then, the payload with the available named layers.

```
hasNext(cantabria_getmap_req, cantabria_getmap_resp)
hasOutput(cantabria_getmap, cantabria_getmap_resp)
hasPayload(cantabria_getmap_resp, cantabria_getmap_image)
allowedType(cantabria_getmap_image, cantabria_topografia5000)
```

Distributed platform bindings. The running example describes the simplest binding supported by OGC Web services. The following code is an excerpt of the section of the service metadata that describes the binding of an operation.

```
<GetCapabilities>
  <DCPType><HTTP><Get>
     <OnlineResource xlink:type="simple"
       xlink:href="http://www.idee.es/wms/WMS-Cantabria/WMS-Cantabria"/>
  </Get></HTTP></DCPType>
</GetCapabilities>
```

This snippet says that the operation *GetCapabilities* uses to support distributed interaction the distributed computing platform comprising Internet hosts that support HTTP. This binding uses the HTTP verb GET to convey the information. The online resource URL intended for HTTP GET requests is in fact an URL prefix to which additional parameters may be appended in order to construct a valid operation request. This binding can be described as follows.

```
HttpBinding(cantabria_binding)
toPlatform(HttpBinding, PlatformHTTP)
verb(cantabria_binding, "GET")
withEncoding(cantabria_binding, KVP)
withEndpoint(cantabria_binding, cantabria_endpoint)
uriTemplate(cantabria_endpoint, "http://www.idee.es/wms/WMS-Cantabria/↵
  WMS-Cantabria{?kvp*}")
```

The URI template draft of W3C (Gregorio et al., 2010) is used here for representing the URI that captures all the KVP requests that should be managed by the endpoint. However, the encoding of the URI template string is outside of the scope of the OntoOWS ontology.

The three operations in the running example have the same binding. This is described by asserting:

```
hasBinding(cantabria_getcapabilities, cantabria_binding)
hasBinding(cantabria_getmap, cantabria_binding)
hasBinding(cantabria_getfeatureinfo, cantabria_binding)
```

Implementable standards and information models. The implementable standards and standard information models should be pre-defined. Below are the statements that define the OGC Service Architecture abstract specification (Percivall, 2002), the WMS 1.3.0 implementation standard (de la Beaujardiere, 2006), and the XML schema that defines its response to a *GetCapabilities* request.

```
AbstractSpecification(as_web_services)
name(as_web_services, "Topic 12: OpenGIS Service Architecture")
version(as_web_services, "4.3")
InterfaceStandard(wms_1_3_0)
name(wms_1_3_0, "WMS")
version(wms_1_3_0, "1.3.0")
isSpecifiedBy(wms_1_3_0, as_web_services)
defines(wms_1_3_0, wms_getrequest_response)
XMLSchema(wms_getrequest_response)
namespace(wms_getrequest_response, "http://www.opengis.net/wms")
version(wms_getrequest_response, "1.3.0")
location(wms_getrequest_response, "http://schemas.opengis.net/wms/↵
  1.3.0/capabilities_1_3_0.xsd")
```

Given the above definitions, these specifications can be related with the service metadata document and the contract.

```
specifies(wms_1_3_0, cantabria_sm)
specifies(wms_1_3_0, cantabria_cnt)
specifies(wms_getrequest_response, cantabria_sm)
```

3.6 Ontology implementation

OntoOWS is implemented as an ontology with 62 classes, 41 object properties, 13 data properties, 7 individuals, 73 class axioms, 24 property axioms, and 1 data property axioms. The ontology is encoded in OWL in Appendix A. The ontology is identified with the URI `http://purl.org/iaaa/sw/ontoows` and its concepts has their names prefixed by the string `http://purl.org/iaaa/sw/ontoows#`.

The expressivity of the ontology is $\mathcal{SOIQ}(\mathcal{D})$. That is, the ontology uses an attributive description logic language with transitive properties. Additionally, some classes are defined by enumeration of individuals or by restriction of values. Some properties have inverses and have qualified cardinality restrictions. Finally, the data properties use explicit datatypes and data values. Figure 3.17 shows the complete class hierarchy of the ontology.

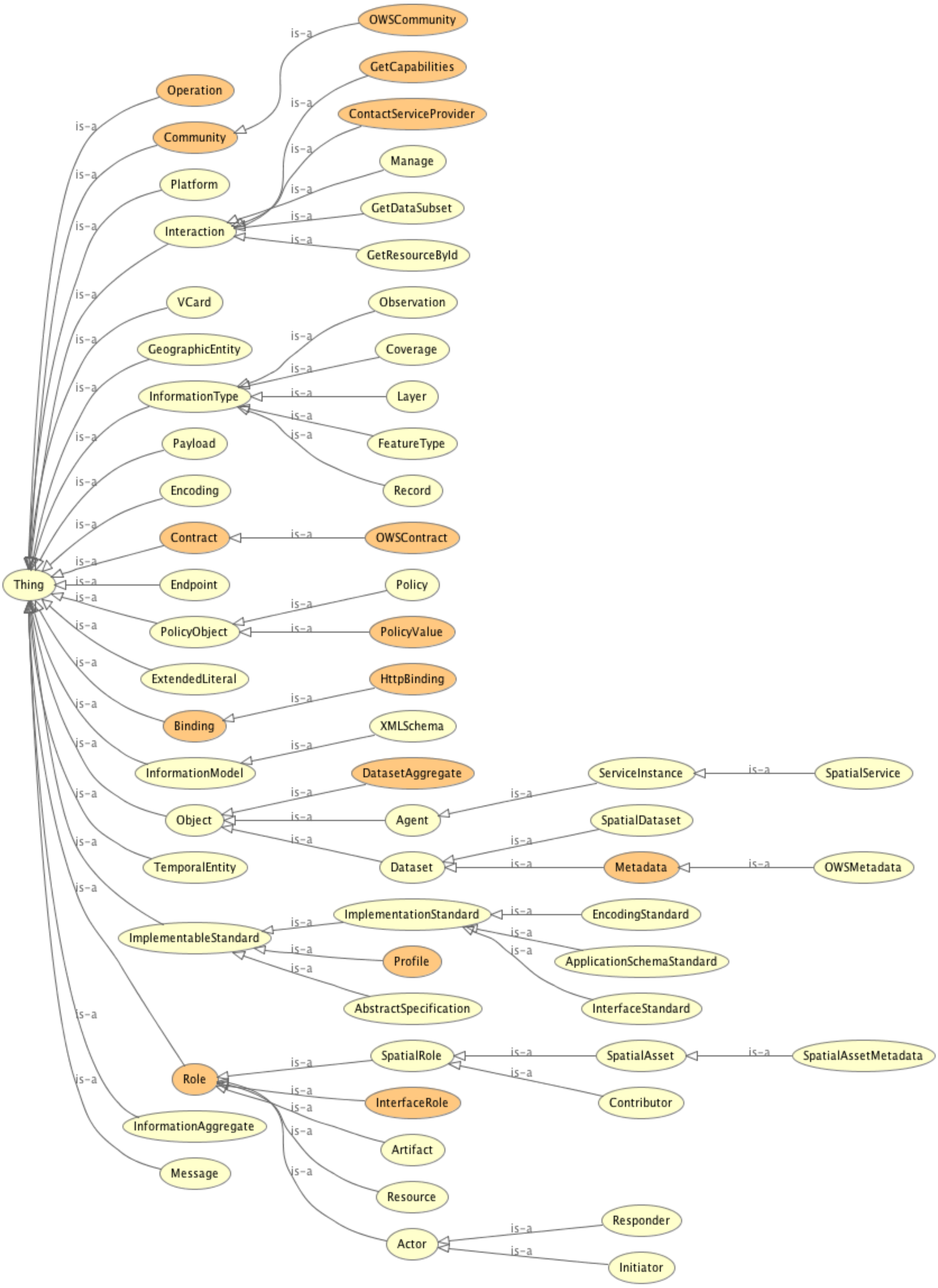


Figure 3.17: OntoOWS class hierarchy.

3.7 Summary of the Chapter

This chapter introduces the approach of this thesis for the modelling OGC Web services. This approach is based in the mapping of concepts found in the OCG Web service metadata documents of each Web service into an ontology derived from concepts of the OGC specifications. The development of the ontology, named OntoOWS, requires a clear delimitation of the scope of the ontology and the use of a well-defined procedure for its development. This thesis proposes the use of an adaptation of the Methontology framework for the construction of the ontology.

The OntoOWS ontology provides a framework that helps to translate in a consistent way OGC Web service metadata encoded into assertions about the service instances, allowing further enrichment. These assertions include not only information about the interfaces, the data types and the platform bindings but also expected behaviour, policies, and technological choices. It is out of the scope of the thesis to detail the implementation of the transformation of OGC Web service metadata documents into knowledge representation models based in the OntoOWS ontology. Nevertheless, part of the chapter is dedicated to provide an overview of the transformation procedure.

Chapter 4

Minimum content model

4.1 Introduction

Once a Geospatial Web service has been discovered by the Web crawler, its content could be surfaced. Chapter 5 analyses a technological approach based in the Semantic Web. This chapter provides a minimum content model suitable for surfacing the content behind WFS servers. WFS servers return feature data encoded in Geography Markup Language (GML, Portele, 2007). However, the detail of the spatial representations in GML is beyond the needs of most Web use cases. An adequate balance between simplicity and usefulness can be reached with a simplified formalization of geospatial content based in the gazetteer data structure (Hill et al., 1999).

This chapter presents the Geo-Net ontology. The purpose of this ontology is to describe simple geographic features using semantic constructs. Figure 4.1 shows the different modules of the example application outlined in the introductory chapter that may use Geo-Net as one of their reference models. Geo-Net is an extension of the GKB (Chaves et al., 2005) metamodel, which is based in the gazetteer data structure and was developed by the project GREASE[1] (GREASE, Silva et al., 2006). Geo-Net was presented in Lopez-Pellicer et al. (2009) and was developed as part of the project GREASE-II.

The development of a content model for metadata is out of the scope of this thesis. However, it makes sense to describe in this chapter the use of the Dublin Core vocabulary (Powell et al., 2007) for interoperate between different metadata schemas in the Geospatial Web. The role of the Dublin Core vocabulary is similar to the role of Geo-Net in the different modules of the example application outlined in the introductory chapter.

The use of formal gazetteers as base for a minimum common model for the publication of knowledge about simple features in machine processable form has sense in the context of OGC Web services and SDIs. For example, Egenhofer (2002) put the gazetteer in the research agenda of the Semantic

[1] `http://xldb.fc.ul.pt/wiki/Grease`

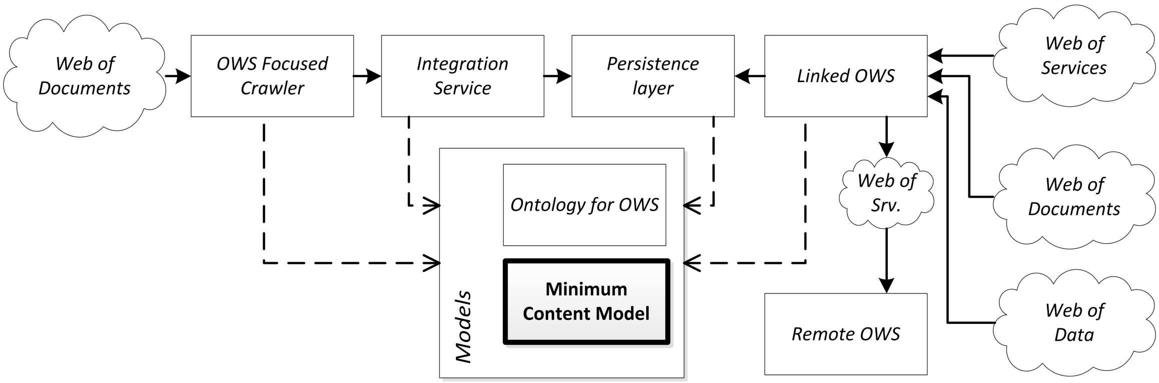


Figure 4.1: The role of the minimum content models in the example application outlined in the introductory chapter.

Geospatial Web in the role of provider of the semantics of geospatial labels. Projects that involved the retrieval of geographical information, such as SPIRIT (Abdelmoty et al., 2007), and use of SDI services, such as DIGMAP (Martins et al., 2007), have required the support of formal gazetteers with different level of expressivity.

This chapter is organized as follows. First, Section 4.2 analyses what is a gazetteer and its role as minimum content model shared by different kind of providers. Next, Section 4.3 summarizes the requirements of the Geo-Net ontology. Section 4.4 presents the conceptual model, the formalization and the implementation of the Geo-Net ontology. Section 4.5 provides examples of the application of Geo-Net. Next, Section 4.6 introduces the use of Dublin Core as minimum content model for metadata. Finally, the main contributions of this chapter are summarized.

4.2 Gazetteers

A *gazetteer* is an artefact that contains authoritative facts about features. A *place* or *geographic feature* is any relatively permanent part of the natural or man-made landscape or seascape that has recognizable identity within a particular cultural context (Orth and Payne, 2003). Different kinds of *named geographic features* may be found in gazetteers. The traditional categories are natural landscape features, populated places and localities, civil and political divisions, administrative areas, transportation routes and constructed features. New ones categories range from devices to utilities service areas.

The *spatial reference system* (SRS) of average users is not coordinate-based but place-name based. Users describe and retrieve data using conventional place-names whose themes range from more concrete ones (natural landscape features, populated places and localities, transportation routes and constructed features) to less tangible ones (civil and political divisions, administrative areas, cadastral parcels and postal addresses). A *place-name*, *toponym* or *geographic name* is an official or conventional name used in everyday language to refer or to identify a place, phenomenon or area that has a cultural recognized identity (UNGEGN, 2006). However, the concept place-name might have a slightly different intension and extension in different application domains. For instance, the intension of the place-name concept in the geographic information context includes their use as a identifier (ISO/TC 211, 2003a) or as a metonymic substitute of a coordinate-based representation of location (Markert and Nissim, 2002); in the same context, its extension broadens to include codes such as "*1*" (ISO/TC 211, 2003a) and hierarchical specified place names such as "*New York City, New York State, USA*" (Axelrod, 2003).

Several SDI experts recognize that data in the SDI should also be indexed using place-name SRSs to increase the recall and/or the precision of place-names constrained queries (Nebert, 2004; Rose, 2004). Therefore defining and supporting a reliable conventional place-name vocabulary is an issue on the SDI development. This vocabulary should deal with problems such as which place-names should be used, which is the role of synonyms, how wide is the area referred by a place-name or recognizing which place the user is asking for. The *conventional wisdom* says that a specialized artefact named *gazetteer* is able to deal with this complex task. However, it seems that it does not exist a unified opinion about the gazetteer content model. Instead, there are different frameworks that identify the most important types of information that a gazetteer should contain (Hill et al., 1999; ISO/TC 211, 2003a; UNGEGN, 2006).

The gazetteer architecture is a core subsystem of some SDI architectures. A good example is the OGC Geospatial Portal Reference Architecture (Rose, 2004). This reference model documents a core set of standard-based services that a geoportal architecture should have: portal services to provide a single point of access, catalogue services to locate geospatial services and data, portrayal services to present data to the user and data services to provide data content and data processing. Its gazetteer subsystem is composed of a gazetteer client that provides users the capability to explore a spatially organized collection of named features, a gazetteer service that allow users to query well-known place-names to retrieve named features, and a gazetteer content model. OGC proposes a refactored ISO 19112 model as gazetteer. This gazetteer should be published using WFS servers (Fitzke and Atkinson, 2006). However, some reference SDIs, such as US Geodata.gov[2], offer non-WFS gazetteer services. Other SDIs publish their own gazetteer content model using WFS (e.g. CGDI, GeoConnections, 2007). Even in these cases the content model is far from ISO 19112. For example, the *Spanish SDI Working Group* has developed the *Spanish Gazetteer Model* (Modelo Español de Nomeclátor, MEN, Rodríguez-Pascual et al., 2006) based on the Alexandria Digital Library Gazetteer content model (ADL Gazetteer, ADL, 2004) that has become the recommended content model in Spain for both gazetteer services and sharing standard gazetteer data between administrations.

One might ask why SDIs have not widely adopted the proposed OGC best practice. Some clues

[2] `http://gos2.geodata.gov/wps/portal/gos`

could be found in the role given to the place-name by SDI users. It has been detected some of these roles through the development of the MEN. For instance, some toponymists have proposed that variant spellings must be out the national gazetteer because they could not be standardized. Other experts such as government officials responsible for cadastral management did not agree with the content of some fields, as they appeared to be useful for naive exploratory search and not for expert data retrieval. Finally, local land managers have criticized that the spelling variants were considered as alternative place-names instead of attributes of the place-name from which the spelling variants are derived. After analysing the arguments, these roles appear to be grouped as follow:

- **Geographic identifier**. GIS sees the gazetteer as the container of a spatial reference system made of location types and their respective location instances. A location instance record refers to a place and contains a preferred place-name written form, which acts as unique identifier in a context (e.g. "*Nowhere, Oklahoma*"), alternative place-names written forms (e.g. "*Madrid*", "*Madrit*", "*Majerit*") and a spatial footprint described using a coordinate reference system (e.g. Denmark footprint as "*West 7.9°, South 54.3°, East 13.2°, North 57.8°*" or a place-name reference system (e.g. Denmark footprint as "*Hovedstaden, Midtjylland, Nordjylland, Sjælland and Syddanmark*"). These place-names and the footprint may be applied for data retrieval as query constraint values. This is the conventional SDI gazetteer behaviour. It was considered that a conventional gazetteer service might offer some explicit thesauri functionality (Atkinson, 2001), however these efforts were discontinued. An example of gazetteer with geographic identifiers is the gazetteer defined on the ISO 19112:2003. These geographic identifiers must provide an unambiguous identification. That requires rewriting many place names in the databases by adding an application in form of higher feature name (e.g. "*Paris*" become "*Paris, Texas*" to be distinguished from "*Paris, New York*" in an ISO 19112 compliant town gazetteer). These refactored geographic identifiers are applied as a replacement of complex geometries on databases for location and data retrieval purposes. For example, a land parcel can replace the geometry that identifies its location by a compact, readable, multipurpose representation: its address (e.g. "*2400 Jefferson Road, Paris, Texas*").

- **Encyclopaedia entry**. Digital Libraries treat the gazetteer as a geographic dictionary, an encyclopaedia of places (Hill et al., 1999), thesauri of places (J. Paul Getty Trust, 2007) or geospatial ontology whose content could be used to georeferencing other resources in accordance to some guidelines (e.g. GOMWG (2005) describes how to use the *Getty Thesaurus of Geographic Names* (TGN) place-names to describe the extension of a resource in absence of standard gazetteer data). These include not only typical SDI artefacts such as maps, imagery and aerial photography but also any kind of resource (e.g. archaeological records). These place-names and the footprint may be then applied for information retrieval (Jones et al., 2001). This kind of gazetteer, created mainly by librarians (e.g. ADL Hill et al., 1999) and historians (e.g. *China Historical GIS* – Berman, 2003), arise from the necessity of cataloguing resources that may contain historical, contemporary and even future space references. In this context the gazetteer could appear as a geographic encyclopaedia, such as the ADL, or as a thesaurus, such as the TGN. These are cited as sources of geographic names used in processes of metadata creation as the Digital Library gazetteer is applied to georeferencing resources indirectly. For example, the spatial coverage of a norm of the Zaragoza City Council (Spain) published in its Web portal[3] could be described as "*World; Europe; Spain; Aragón; Zaragoza; Zaragoza*" according to TGN or "*Zaragoza; Spain*" indicating that it is of type populated place according to ADL.

- **Standardized administrative identifier**. It is widely recognized that a national geographic name standardization programme produces savings in time and money by increasing operation efficiency organizations (UNGEGN, 2006). A standardization programme also provides to an authority means to reinforce its authority and to recover and to disseminate a cultural heritage. Typical standardization programme results are the standard gazetteer and the concise gazetteer. This kind of datasets is often published through a SDI. Even there exist an international recommendation to establish a gazetteer as key component of a National SDI (UNCHS, 2001). The gazetteer is the sharing vehicle of the established official geographic names and their applications. The reference model that serves as base to these gazetteers has been established by the United Nations working group on geographic name standardization, the United Nations Group of Experts on Geographical Names (UNGEGN, 2006). Although UNGEGN is not a geographic names board, its recommendations about the gazetteer content are followed by place names boards of United Nations members. The goal of standard gazetteers is different from the previous ones. They aim to solve economic and political necessities. The economic one is derived from the reduction of costs caused by errors in the use of geographic names by administrations, companies and citizens. The political one is caused for the need of increasing the national self-esteem by protecting and/or recovering the linguistic heritage and by reinforcing the government authority on the territory by settling the official names. Among others, this result is seen in form of maps in which appear the official and/or standardized names. This data has a virus nature: changes spread through the country organizations. Applied to data retrieval may fail: usually the data uses non-standardized geographic names. May include styled data to ease the production of official products such as maps.

- **Graphic symbol**. In the cartography process the written form of each place-names obtained from a gazetteer is applied to an entity shown graphically on a map. Both graphics and written forms that represent each entity used to vary depending on the context: scale, audience, output device, etc. A toponymic guideline contains the rules that would enable cartographers to apply correctly the place-names on maps (UNGEGN, 2006). These guidelines should contain or refer to a toponym list or gazetteer but only as far as its content could be of benefit

[3] http://www.zaragoza.es/

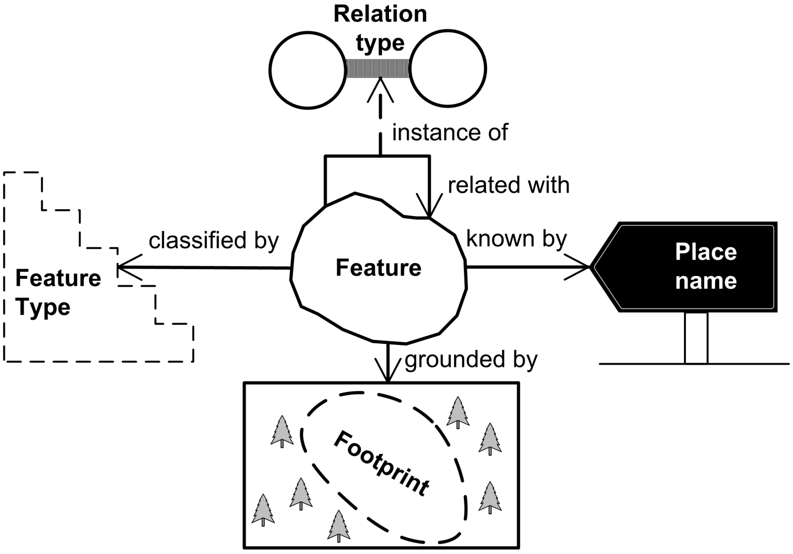


Figure 4.2: Elements that appears in gazetteers.

to the cartographic process. This is usually a list of standardized geographic names of an administrative unit where next to each official name there is sufficient additional information that allows locating its written form in a map and identifies the graphical representation of the feature. Therefore, the allocation in the map is dependent of criteria such as readability. Hence, it is quite difficult to trace the relationships with other graphical representations in the map. This gazetteer possibly acts as a portrayal of other gazetteers. Its styled content is explicitly included or implicitly stated by a toponymic guide and it may consider spatial footprint and geographic name changes if context properties such as scale change.

Each of the above communities gives a different role to the gazetteer. However, its core properties are the same: a feature is known by its place names, grounded by its footprints, classified by feature types and related with other features with typed relations (see Figure 4.2). Hence, the gazetteer might provide a minimum content model for simple named features stored in data repositories accessible using the Geospatial Web.

4.3 Requirements

The specification of the Geo-Net ontology is partially based in the structure of the requirements template proposed by Suárez-Figueroa et al. (2009). Geo-Net is a lightweight ontology. As a consequence the list of requirements is short. This specification defines:

- **Purpose**. The purpose of building the ontology is to provide a minimum content model for the descriptions of features encoded in GML that the WFS interface standard can return.

- **Scope**. The universe of discourse is the simple named features that can be considered as part of a gazetteer and found in WFS servers.
- **Level of formality**. The maximum level of complexity is given by OntoOWS (see Section 3.3). That is, portions of the ontology should be amenable to implementation using production rule systems, such as DLJena (Meditskos and Bassiliades, 2008) or OWLIM (Kiryakov et al., 2005). In addition, the maximum level of complexity should be $SROIQ(D)$, which is supported by popular reasoners such as Pellet (Parsia and Sirin, 2004) and HermiT (Motik et al., 2009b).
- **Intended uses**. The intended use in the context of this thesis is to provide a user with machine processable description of the contents of an OGC WFS service instance (Chapter 5).
- **Intended end-users**. The users in the context of this thesis are the semantic proxy described in Chapter 5, and the developer of both systems.
- **Non-functional requirements**. This specification identifies the following requirements:
  - **Multilingual**. The ontology should be able to represents human readable text in different languages.
  - **Spatial representation**. The ontology treats the spatial representation of features as generic data values.
- **Functional requirements**. This specification identifies the next requirements in relation with the description of simple named features:
  - **Concepts**: Features, place names, feature types, footprints and relation types
  - **Relations**: Features are known by place names, classified by feature types, grounded by footprints, and related each other
  - **Constraints**: Features are classified by enumeration, footprints grounds only one feature, and relations between features are typed

4.4 Ontology

This section presents the conceptual model of the Geo-Net ontology, its formalization and its implementation.

4.4.1 Conceptual model

This section presents the conceptual model behind the minimum content model for geospatial knowledge, which extends the conceptual model presented in Chaves et al. (2005). This model is based on ideas presented in Hill (2000), Manov et al. (2003) and Fu et al. (2005). The geospatial knowledge falls in three broad categories (Mark, 1993):

- **Declarative geographic knowledge**. This knowledge is the set of facts that may or may not be associated with a clear and crisp idea of where are these named places are located on Earth. In the phrase *Lisbon is the capital of Portugal,* we express a set of facts that are independent of where on Earth *Lisbon* and *Portugal* are located.
- **Configurational geographic knowledge**. This knowledge is the set of facts that range from basic topological relations to complete coordinate descriptions. In the phrase *Lisbon is part of Portugal and is located at* $38° \, 42' \, N$, $9° \, 11'$, we express a basic configuration as a topological relation and a location described with coordinates.
- **Procedural geographic knowledge**. The rules and clues that allow a person to perform a spatial task using geographic knowledge.

The minimum content model that provides Geo-Net only conceptualizes declarative and configurational knowledge. It is composed by the concepts (see Figure 4.3):

- **Geographic features** (*Feature*). Each geographic feature is required to be partially specified in terms of the declarative geographic knowledge, that is:

  1. A feature can be instantiated in the conceptual model if it has at least a place name or a label and a type.
  2. Two different features can have the same name and type signature.

  These constraints represent the fact that people can recognize place names in the context of a communication with little context. Place names are inherently ambiguous, and they do not need to be georeferenced to be acknowledged. The roles *type* and *name* relate a *Feature* with feature types and place names, respectively.

- **Place names** (*PlaceName*). The linguistic content of the place name is captured in *lemma*, a canonical form of the name, and language. In some cases, the name that identifies the feature is just a label. For example, the name of a digital resource that describes the city of Lisbon can have as name *Fernando Pessoa: Lisbon, what the tourist should see* (`http://www.shearsman.com/pages/books/catalog/2008/pessoa_lisbon.html`). Only *real* place names deserve the qualification of place names, and, hence, if a kind of feature does not have real place names, the place name is not needed to the instantiation. This conceptual model allows the specification of place names not explicitly related to features.
- **Types** (*FeatureType*). They classify geographic features. Classification is the process of assigning elements or units to classes carrying some kind of geographic meaning according to

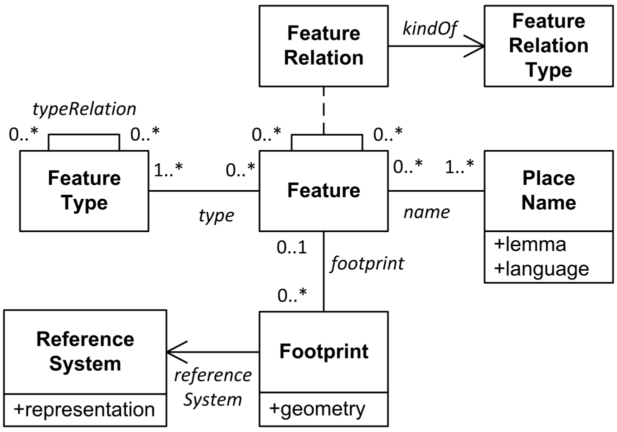


Figure 4.3: Conceptual model of Geo-Net.

some criteria. Our conceptual model does not intend to enforce a typing schema. We assume that the set of features classified by a type is defined by explicit enumeration, that is, a set of statements that assert membership. A set defined by enumeration allows describing that these features share a set of properties without stating which or how fuzzy they are. The main drawback of this assumption is that the system cannot use the properties associated to each feature type to infer or verify knowledge. The relation *typeRelation* describes binary semantic relationships among instances of the *FeatureType* class. The relation *typeRelation* provides the minimum machinery for supporting taxonomies of feature types. This conceptual model allows the specification of types not explicitly related to features.

- **Relations between features** (*FeatureRelation*). They are classified by the concept *FeatureRelationType*. Relationships document the configurational geographic knowledge derived from the footprints. These include relations such as *part of*, *adjoint to* and *connect to*. Relationships also describe declarative geographic knowledge, such as *capital of*, *administrative division of* and *former part of*. At least, relationships should be documented to determine if they are pre-calculated from spatial relationship or they extend beyond the configurational knowledge.

- **Locations** (*Footprint*). The spatial description of a location is captured in the field *geometry* and complemented by a *ReferenceSystem*. The footprint of physical and mental features might be as complex as a survey description of the boundaries of the feature or as simple as a pinpoint. This configurational knowledge is restricted by a single rule: a footprint cannot ground more than one feature. This constraint assumes that footprints are tightly bounded to only one feature. However, this conceptual model allows the specification of footprints not explicitly related to features.

The description of time is not explicitly considered in the Geo-Net conceptual model. It is only a framework to identify the above concepts and properties. The description of its temporal dimension should be done with appropriate vocabularies, such as the temporal vocabulary described in Gutierrez et al. (2007). This issue is part of the future work identified in Chapter 6.

4.4.2 Formalization

The superstructure of the formalization Geo-Net is composed by the concepts *GeographicConcept*, *InformationDomain*, *Source*, *GeographicEntity*, *FeatureType* and *ReferenceSystem*.

$$\begin{align}
&disjoint\,(GeographicConcept, InformationDomain, Source) \tag{4.1}\\
&GeographicEntity \sqsubseteq GeographicConcept \tag{4.2}\\
&FeatureType \sqsubseteq GeographicConcept \tag{4.3}\\
&ReferenceSystem \sqsubseteq GeographicConcept \tag{4.4}\\
&disjoint\,(GeographicEntity, FeatureType, ReferenceSystem) \tag{4.5}
\end{align}$$

The concept *GeographicConcept* represents the concepts related with geographic features introduced in the conceptual model described in this chapter. Application vocabularies can extend the conceptual model using this concept. The concept *GeographicEntity* represents concepts that can be used to represent a location in the Earth, directly or indirectly. The *GeographicEntity* concept has been reused in the OntoOWS ontology (see Chapter 4) for the representation of the spatial coverage of Web services. The rest of concepts are presented below. Some concepts include recommendations of use that reflect lessons learned during the implementation of the Geo-Net in different scenarios (Cardoso et al., 2009; Lopez-Pellicer et al., 2009, 2010c).

Feature. A *feature* represents any meaningful object that can be grounded directly or by reference. The formalization of this concept is as follows:

$$\begin{align}
&Feature \sqsubseteq GeographicEntity \sqcap \exists name.PlaceName \sqcap \forall name.PlaceName \sqcap\\
&\quad \exists type.FeatureType \sqcap \forall type.FeatureType \sqcap \forall footprint.Footprint \sqcap \tag{4.6}\\
&\quad \forall relation.Feature \tag{4.7}\\
&relation \equiv relation^{-1} \tag{4.8}\\
&\top \sqsubseteq 1 \leq footprint^{-1}. \tag{4.9}\\
&disjoint\,(Feature, PlaceName, Footprint) \tag{4.10}
\end{align}$$

The concept *Feature* is a typed named *GeographicEntity* with optional footprints. This constraint reflects the assumption that each geographic feature must be partially specified in terms of declarative knowledge, that is, a feature needs to have at least a place name and a feature type. The role *footprint* relates a feature with its footprints. The inverse of the role *footprint* is functional. This constraint enforces the conceptual restriction that footprints are not shared among features.

Features can be related each other with the binary role *relation*. This point differs from the conceptual model presented before. The n-ary relation *FeatureRelation* that involves two *Features* and a *FeatureRelationType* is formalized with the binary role *relation* between features that can be subclassed to represent different feature relation types. Application vocabularies can extend the conceptual model using this role. The Geo-Net-PT 02 ontology presented in the Section 4.5 is a good example of the extension of the role *relation*.

Feature types. The Geo-Net vocabulary does not intend to enforce a typing schema: the set of features classified by a type is defined by explicit enumeration. A set defined by enumeration allows describing that these features share a set of properties without stating which or how fuzzy they are. Hence, the instances of the concept *FeatureType* represent sets of features. The intended meaning of the role *type* is to assert that a feature shares characteristics with other features classified by the same feature type. The conceptual model defines the relation *typeRelation*. This conceptual relation is implemented with the role *relation*. This relation provides the minimum machinery for implementing taxonomies.

$$FeatureType \sqsubseteq GeographicConcept \sqcap \forall relation.FeatureType \tag{4.11}$$

Relationships. Relationships record declarative geographic knowledge, such as *capital of*, *administrative division of* and *former part of*, and configurational geographic knowledge, such as *part of*, *adjoint to* and *connect to*. The relationships are represented in Geo-Net with the role *relation* that has been introduced above.

Place names. A *place name* represents proper names of one or more features. The definition of proper name is not restrictive. For example, authorities can define schemas of codes for identifying features. These codes, known as geographic codes, are considered proper names in this vocabulary. The concept *PlaceName* is formalized as follows:

$$\begin{aligned} &PlaceName \sqsubseteq GeographicEntity \sqcap \exists lemma.xs\colon string \sqcap \\ &\quad \forall languageCode.xs\colon string && (4.12) \\ &Func(lemma) && (4.13) \end{aligned}$$

$$Func\,(languageCode) \tag{4.14}$$

The pair of roles *lemma* and *languageCode* is a key for the concept *PlaceName*. A consistent use of the role *lemma* relates a *PlaceName* individual to a unique lexical form considered the canonical representation of the denoted place name. The range is a typed literal with type *xs:string*. The canonical form must be a Unicode string, used in communication, indivisible, and without additional context information. That is, the lemma identifies the lexical form used as place name, such as *Paris*, not the lexical form of the proper name of a specific feature, such as *Paris, Texas*.

Place names can contain a proper name part and a generic part that may hint the feature type. However, the inference of the generic part is ambiguous. For example, the place name with the lemma *Rio Douro* can refer to the *Douro* River or an administrative division with the name *Rio Douro*. The recommended best practice for applications is to avoid practices such as splitting the lemma in parts to infer possible feature types or any other context information. Place names often require for their understanding an additional context in the form of additional details. Application vocabularies can represent other lexical forms, such as the hierarchical form *Paris, Texas*, by extending the concept *PlaceName*. If the implementation requires the persistence of values in other character encodings, such as ASCII, these never must be stored in the property lemma.

The role of lemma as key raises an additional problem. Names are vague and ambiguous, not only because language users can use them to refer to vague places, but also because they can use different case, spellings and abbreviations. For example, *Olissippo*, *Olisipo*, *Ulisipo*, *Olisponna* but also *OLISIPO* and *olisipo* are different strings, but they all are versions of the historical name of Lisbon during the Roman Empire. The decision of how a string is matched to a name is dependent of how the conceptual model is implemented. The recommended practice is to formalize the method that transforms a string to a form considered canonical. This method can be as simple as the lower case function, and as complex as a complete lemmatization.

A consistent use of the role *languageCode* relates a *PlaceName* individual to a well-known string value that identifies the language of the denoted place name. The range is an *xs:string* literal. A best practice is the use of identifiers from ISO 639-3:2007. ISO 639-3:2007 is a standard that aims to define three-letter identifiers for all known human languages.

A place name can play several roles along its existence as a communication tool that often reveals significant patterns of environment, settlement, colonization, exploration, organization, historical facts and folk etymology. Application vocabularies derived from the Geo-Net vocabulary can create specialized relations to represent these roles.

Footprints and reference systems. A *footprint* is a location on a surface. The footprint of a feature might be as complex as a survey description of the boundaries of the feature, or as simple as a pinpoint. A *footprint* to be fully specified requires the specification of a unique reference system for all the points stored in the geometry field. The list of coordinate points takes the form $(x,\ y)$

if the georeferencing system is based on a planar, Cartesian or two-dimensional reference system, or (*latitude*, *longitude*) if the georeferencing system is based on a three-dimensional reference system. A planar reference system requires a map projection that describes how to transform the planar coordinates onto the Earth's surface and vice versa. Both planar and three-dimensional reference systems need the definition of a geodetic datum. That implies a necessary relation with a concept that represents the *reference system*. This concept should provide directly or indirectly the geodetic datum, and, if needed, the projection. The concepts *Footprint* and *ReferenceSystem* shall be formalized.

$$Footprint \sqsubseteq GeographicEntity \sqcap \exists referenceSystem.\top \tag{4.15}$$

$$\top \sqsubseteq \leq 1 referenceSystem.ReferenceSystem \tag{4.16}$$

These definitions are lightweight. In addition, the Geo-Net ontology defines the unqualified data roles *geometry* and *representation*. The role *representation* is a key of the concept ReferenceSystem. The pair of roles *referenceSystem* and *geometry* is a key for the concept *Footprint*. The values and data types of these roles depend on the use of the ontology.

A consistent use of the role *geometry* as part of the key should relate a *Footprint* with a canonical geometrical description of the footprint shape. Its data value should be encoded in one of the following encoding schemes for representing geometry data: Well-Know Text (WKT), and Geography Markup Language (GML). WKT and GML are GIS industrial standards specified by Open Geospatial Consortium (OGC). WKT syntax for 2D and 3D geometries is described in the OGC SFA specification (Herring, 2006), and GML is described in (Portele, 2007). GML is also an ISO standard (TC 211, 2007a). As a best practice, crisp 2D geometries, such as points, lines, polygons and combinations of these, should be represented as WKT geometries with coordinates in 2D. Fuzzy 2D geometries should be represented as WKT geometries with coordinates in 3D (see Jones et al., 2008).

A consistent use of the role *representation* as a key of the concept *ReferenceSystem* should relate a *ReferenceSystem* with a canonical description of the georeferencing system encoded in WKT. The WKT syntax for coordinate reference systems is described in the OGC CTS specification (Daly, 2001).

Structure and provenance. The Geo-Net vocabulary provides support to organise data into collections. These collections are called information domains, and are represented by the concept *InformationDomain*. This concept disjoints with other concepts of the ontology. The role *inDomain* marks the assignment of a resource to an information domain. Information domains allow the partitioning of data in coherent collections. I is also possible the assignment of a geographic concept to multiple information domains. This enables the organisation of the geographic information in overlapping collections if necessary. The potential users of a dataset cannot evaluate the authority of the claimed facts without a proper description of its provenance. The concept *Source* represents sources whose contents have been added to a dataset described with the Geo-Net vocabulary. The information model applied to the description of a source should be similar to the applied to describe sources in the geographic information domain, such as the detailed in Kresse and Fadaie (2004); Nogueras-Iso et al. (2004). The concept *Source* disjoints with other Geo-Net concepts. The provenance of items of information is stated through the role lineage that relates a concept or statement with its source, a resource of type *Source*.

Next, the formal definition of the structure and provenance in Geo-Net:

$$\top \sqsubseteq \forall inDomain.InformationDomain \tag{4.17}$$

$$\top \sqsubseteq \forall lineage.Source \tag{4.18}$$

4.4.3 Implementation

Geo-Net is implemented as an ontology with 9 classes, 7 object properties, 4 data properties, 9 class axioms, 7 property axioms, and 4 data property axioms. The ontology is encoded in OWL in Appendix B. The ontology is identified with the URI `http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net` and its concepts has their names prefixed by the string `http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net#`.

The expressivity of the ontology is $\mathcal{ALCIF}(\mathcal{D})$. That is, the ontology uses an attributive description logic language with transitive properties. Additionally, some properties are functional and have inverse. Finally, the data properties use explicit datatypes and data values. Table 4.1 presents the terms of the Geo-Net vocabulary. The Geo-Net vocabulary, encoded in OWL 1, is available from `http://xldb.di.fc.ul.pt/wiki/Geo-Net-PT_02` together with vocabularies and datasets based on it, such as the Geo-Net-PT vocabulary and the Geo-Net-PT 02 ontology. A dataset described by the Geo-Net vocabulary is interpreted as a collection of individuals and a collection of relations between these individuals, where both collections are constrained by the axioms of the Geo-Net vocabulary.

4.5 Application

The Geo-Net ontology has been tested in the development of geospatial knowledge bases and knowledge management systems in the project GREASE-II[4]. This section presents its application into the extension of the knowledge management system known as GKB system, and in the development of the geospatial ontology of Portugal Geo-Net-02 PT.

[4] `http://xldb.fc.ul.pt/wiki/Grease`

Table 4.1: Classes and properties of the Geo-Net vocabulary.

	Term	**Description**	**Specializes**
Class	*gn:GeographicConcept*	root concept	-
	gn:FeatureType	classifier	*gn:GeographicConcept*
	gn:GeographicEntity	entity	*gn:GeographicConcept*
	gn:Feature	feature	*gn:GeographicEntity*
	gn:PlaceName	proper name	*gn:GeographicEntity*
	gn:Footprint	footprint	*gn:GeographicEntity*
	gn:InformationDomain	collection	-
	gn:Source	provenance metadata	-
	gn:ReferenceSystem	spatial reference system	-
Object	*gn:name*	has name	-
property	*gn:type*	is classified by	-
	gn:relation	is related with	-
	gn:inDomain	is member of	-
	gn:lineage	has provenance metadata	-
	gn:referenceSystem	has reference system	-
Datatype	*gn:lemma*	has a canonical form	-
property	*gn:languageCode*	belongs to language identified by	-
	gn:geometry	has a canonical encoding of its shape	-
	gn:representation	has literal description of reference system	-

4.5.1 Extension of the GKB system

The GKB system is a relational repository developed in the project GREASE. GKB is based on a domain independent meta-model for integrating geographic knowledge collected from multiple sources presented in Chaves et al. (2005). GKB includes tools for the serialization of the knowledge in Semantic Web representation languages. The development of the Geo-Net ontology helped in the extension of the GKB system to enhance the support of geospatial descriptions. This extension was presented in Lopez-Pellicer et al. (2009) and Lopez-Pellicer et al. (2010c).

Architecture. The architecture of the extended GKB system is represented in Figure 4.4. Its structure can be analysed taking into account how a client application can interact with the managed content. It is possible to interact with an ontology instance managed by the GKB system in three different paradigms:

- **Relational**. A relational database schema accessible through SQL interfaces. This interaction mode entails that the client application must query and reason following all the rules and assumptions formalized in the Geo-Net vocabulary. The components of the relational persistence system are the tailored schema, the storage that implements it, and the API that provides interaction with the repository. Both the relational storage and its API must support the OGC Simple Feature specification for spatial datatypes (Herring, 2006). The tailored schema is a platform-dependent model of the GKB metamodel that instances one or more information domains. These information domains can be modified to implement application requirements. The client applications can use the GKB instance by querying the relational persistence system.

- **Object oriented**. An object-oriented data-structure accessible through an API. The API works as a contract between the application and the GKB by exposing the domain model. The components of the object-oriented persistence system are the content access framework, the access API, and the object-relational mapping (ORM) library. The ORM library must support code generation from schemas and standard spatial datatypes. The back-end of the object-oriented persistence system is the relational persistence system. The content access framework is an object-oriented library, which has been semi-automatically generated from the tailored schema of the relational-based persistence system. The access API is a lightweight set of interfaces that partially implements the GKB metamodel and the additional classes, and provides graph navigation methods. The process that generates the content access framework is instructed to implement the access API in the generated code and wire the access API methods to fields of the tailored schema. The advantage of the object-oriented persistence system over the relational is that the generated code structure is closer to the GKB metamodel, and the ORM library can provide extra functionalities, such as caching and transparent fetching of data. The main disadvantage is the potential complexity of wiring the code and the additional overload of the ORM library to some tasks.

- **Resource oriented**. A knowledge representation described by an OWL ontology, which uses the Geo-Net vocabulary, accessible through a semantic aware interface, such as a SPARQL end-point (Seaborne and Prud'hommeaux, 2008), or serialized in a semantic format, such as RDF/XML (Beckett, 2004). The Geographic Ontology Serializer (GOS), the resource-based storage, and the RDF/OWL API that manages the resource storage compose the resource oriented persistence system. The GOS is described in detail below. The resource-oriented persistence system employs the object-oriented persistence system as back-end database. Applications can access the resource-oriented persistence system using a SPARQL end-point or the RDF/OWL API. In addition, the resource-oriented system can be used to create RDF

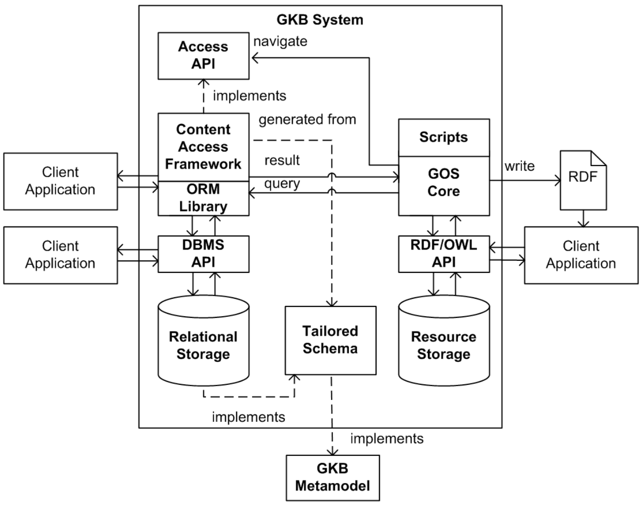


Figure 4.4: The architecture of the GKB System.

serializations of the ontologies managed by GKB.

Geographic ontology serializer. GOS is a GKB module that takes an ontology hosted in a GKB instance and creates its representation in one of the available RDF serialization formats. The main components of GOS are:

- **Access API** implements the current GKB metamodel based on the Geo-Net vocabulary. This API is the facade for any serializable content.

- **Content Access Framework** provides an ORM to relational storage instances of a specific GKB schema. The content access framework is generated using reverse engineering techniques and implements the Access API.

- **Application Script** selects the contents to be serialized using the Access API and provides specific mappings between a GKB tailored schema and an application vocabulary describing its semantics.

- **GOS Core** orchestrates the creation of serializations of GKB instances. The GOS Core creates and manages an RDF staging area associated to the execution of an application script. This component is also responsible for writing out in the desired RDF format.

Users can run application scripts in the GOS Core. The application script can use GOS core to query the Content Access Framework using its Access API, or use the default Geo-Net mappings implemented in the GOS Core.

Table 4.2: Descriptive statistics of Geo-Net-PT 02 concepts; this table only considers direct assertions.

Concept	**Adm.**	**(%)**	**Phy.**	**(%)**	**Net.**	**(%)**
Features	386,067	92.9	5,676	1.4	23,666	5.7
Names	265,044	97.0	8,266	3.0	-	0.0
Footprints	4,597	100.0	-	0.0	-	0.0
Types	62	69.7	25	28.1	2	2.2
other	4,597	58.9	3,207	41.1	-	0.0
Total	660,637	94.2	17,174	2.4	23,668	3.3

Technical details. The relational system of choice for GKB is PostgreSQL[5] 8.3.6. The support for geographic objects to the PostgreSQL is provided by PostGIS 1.3.6[6]. PostGIS implements the persistence of geographic objects following the OGC SFA specification. The language for the development of the object-oriented and resource-oriented persistence systems is Java SE 1.6. The ORM library of choice for the object-oriented persistence system is Hibernate Core[7] 3.3. The support for geographic objects is provided by the extension Hibernate Spatial[8]. The storage system of choice for the resource-oriented persistence system is the RDF-based file storage TBD[9]. TDB is a non-transactional file-based RDF storage optimized for large-scale models. TDB provides for large-scale storage and query of RDF dataset. TDB is a component of Jena, a framework for building in Semantic Web applications. Jena and TDB are open source projects initially developed in the HP Labs.

4.5.2 Geo-Net-PT 02

The Geo-Net-PT 02 (Lopez-Pellicer et al., 2009), a geospatial ontology of Portugal, is an authoritative geographic knowledge dataset created with the extension of the GKB system described in the previous section. This ontology is available in the XLDB Node of Linguateca[10]. Geo-Net-PT 02 is the evolution of the Geo-Net-PT-01 ontology (Chaves, 2008) enriched with data from the physical domain (Rodrigues, 2009).

The GKB repository. Geo-Net-PT 02 defines 701,209 concepts, most of them administrative features and place names (see Table 4.2). The Geo-Net-PT 02 GKB repository contains of three

[5] `http://www.postgresql.org/`
[6] `http://postgis.refractions.net/`
[7] `http://www.hibernate.org/`
[8] `http://www.hibernatespatial.org/`
[9] `http://www.openjena.org/wiki/TDB`
[10] `http://xldb.di.fc.ul.pt/wiki/Geo-Net-PT_02_in_English`

GKB instances: geo-administrative, geo-physical and network. The geo-administrative instance includes human geography features, such as administrative regions. The geo-physical instance includes physical geography features, such as natural regions and man-made spots. The network instance stores data about Web sites. Each of these instances contains only data about Portugal. The content of the geo-administrative and network domains is the same as Geo-Net-PT 01 but mapped to fit the Geo-Net vocabulary. The geo-administrative and geo-physical features are organized in 81 feature types. Postal code, street layout and settlement are the most common feature types found in the geo-administrative domain. Hydrography and touristic resources, such as museums and hotels, are the most common feature types found in the geo-physical domain. The geographic descriptions are in 5 different coordinate reference systems, and there are two different coordinate reference systems for footprints located in Portugal's mainland (ETRS 1989 TM06-Portugal, Lisboa Hayford Gauss IGeoE). Geo-Net-PT 02 has 21 different sources. The main source is CTT (`http://www.ctt.pt`), Portugal's mail services, which provides mainly addresses. Other relevant local sources are Fundação para a Computação Científica Nacional (FCCN, `http://www.fccn.pt`), Agência Portuguesa do Ambiente (APA, `http://www.apambiente.pt`) and Instituto Geográfico Português (IGP, `http://www.igeo.pt`). FCCN provides data about Web sites. APA provides data about features in the geo-physical domain. IGP provides official data about administrative features. Finally, Wikipedia is a complementary source that provides ancillary data about the administrative structure.

The Geo-Net-PT vocabulary. This vocabulary is an extension of the Geo-Net vocabulary. The goal of this extension is to annotate specific characteristics of the Geo-Net-PT 02 data. The prefix *gnpt:* identifies terms of the Geo-Net-PT vocabulary. The prefix *gnpt:* maps to the string `http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net-pt#`. The terms of the Geo-Net-PT vocabulary are summarized in Table 4.3.

The Geo-Net-PT vocabulary defines three subproperties derived from the Geo-Net property *name*: *gnpt:preferred*, *gnpt:alternative* and *gnpt:identifier*. The property *gnpt:preferred* is applied to mark the main label for the geographical feature in an information system. The notion of preferred name implies that a geographical feature can only have one such name per language. The property *gnpt:alternative* is applied to relate a *Feature* with a resource instance of the class *PlaceName* not considered preferred. Finally, the property *gnpt:identifier* is an inverse functional property asserts that the place name uniquely identifies that feature.

A place name is not a safe feature identifier for management or administrative purposes. Authorities can define schemas of codes for identifying features. These codes are known as geographic codes. In the context of Geo-Net-PT, there are some administrative features, such as administrative divisions, and physical features, such as rivers, that have known geographic codes. These codes can be found in databases and specialized documents. The class *gnpt:GeographicCode*, which is subclass of *PlaceName*, represents these codes. The controlled vocabulary where the geographic code is

Table 4.3: Classes and properties of the Geo-Net-PT vocabulary.

	Term	Description	Specializes
Class	`gnpt:GeographicCode`	alphanumeric identifier	`gn:PlaceName`
Object property	`gnpt:preferred`	has as preferred name	`gn:name`
	`gnpt:alternative`	has as alternative name	`gn:name`
	`gnpt:identifier`	is identified in some Schema	`gn:name`
	`gnpt:identifies`	identifies in some Schema	-
	`gnpt:scope`	spatial content about	`gn:relation`
	`gnpt:isLocatedOn`	is on the land surface of	`gn:relation`
	`gnpt:isAdjacentTo`	is adjacent to	`gn:relation`
	`gnpt:isConnectedTo`	is connected to	`gnpt:isAdjacentTo`
	`gnpt:isPartOf`	is part of	`gn:relation`
	`gnpt:hasPart`	has as part	`gn:relation`
	`gnpt:inSchema`	specified in	`gn:lineage`
Datatype property	`gnpt:population`	is inhabited by *literal value* people	-

defined is identified by the functional property *gnpt:inSchema*, which is a subproperty of *lineage*.

One of the goals of the Geo-Net-PT vocabulary is to describe conceptual, hierarchical and topological relations between features. These relations are subproperties of the Geo-Net property *relation*. This vocabulary includes the properties *gnpt:scope*, *gnpt:isLocatedOn*, *gnpt:isAdjacentTo*, *gnpt:isConnectedTo*, *gnpt:isPartOf* and *gnpt:hasPart*. The property *gnpt:scope* declares that the subject of the relation is a resource that identifies a real world object, such as a book, or a digital resource, such as a Web page, whose content has a spatial extent described by the object of the relation. The property *gnpt:isLocatedOn* asserts that the subject is a feature located on the land surface of the target feature. The property *gnpt:isAdjacentTo*, which is symmetric, states that its subject is a feature touching another feature. The property *gnpt:isConnectedTo*, which is a symmetric subproperty of *gnpt:isAdjacentTo*, asserts that its subject is a feature attached to another feature such that objects may flow between them. The property *gnpt:isPartOf*, which is transitive, declares that its subject is a feature that is a physical or logical component of the object feature. Finally, the property *gnpt:hasPart*, which is transitive and inverse of *gnpt:isPartOf*, asserts that its subject is a feature has as physical or logical component of the feature designated by the object. The Geo-Net-PT vocabulary also has the datatype property *gnpt:population*, which asserts the inhabitants of the location. The value of this property is used to rank features.

The Geo-Net-PT 02 serialization. The Geo-Net-PT 02 serialization for the Web uses the Geo-Net and Geo-Net-PT vocabularies, and resuses concepts form known vocabularies, such as SIOC[11], FOAF[12] and Basic Geo (Brickley, 2004). The prefixes *sioc:*, *foaf:* and *geo:* identifies concepts imported form SIOC, FOAF and Basic Geo respectively. The serialization procedure uses the RDFS properties *rdfs:label* and *rdfs:comment* to provide a human readable name for the serialized resources, and *rdfs:seeAlso* for linking to related resources. The use of these vocabularies eases the reuse of the Geo-Net-PT 02 by other communities. The details of the serialization are described below:

- **Features**. They are serialized as *Feature* instances, classified by a *FeatureType*, with a human readable *rdfs:label*, members of an *InformationDomain*, and information about their provenance captured with the *lineage* property. Features from the geo-administrative and the geo-physical information domains contain as many labels as place names. These labels are derived from the relations with place names asserted in gnpt:preferred and alternative. Also, the features can be reused as spatial references. For that reason, the class *geo:SpatialThing* is reused to describe these resources as having spatial extent. 390,664 administrative and physical features and footprints are classified as *geo:SpatialThing.* Finally, if a detailed description of the spatial extent is available as a footprint, the Geo-Net property *footprint* points to it. Features from the network domain represent Web sites and Internet domains. Each network feature is annotated as a resource of type *sioc:Space.* Its label is based on the URL of the Web page or on the registered name of the Internet domain. If the network feature describes a Web page, the *rdfs:seeAlso* links the feature to it.

- **Place names**. They are serialized as *PlaceName* instances, with its *lemma*, its *languageCode* and a human readable label. Also, they are tagged as members of one of the information domains. Only place names from the geo-administrative and the geo-physical information domains are serialized. Some geographic codes are explicitly serialized as resources of type *gnpt:GeographicCode.* These names are related to features by the properties *gnpt:identifies* and *gnpt:identifier.* Also, the property *gnpt:inSchema* links the name with the schema where the geographic code is defined.

- **Feature types**. They are serialized as instances of the class *FeatureType*, with a human readable label and members of an *InformationDomain* instance. Feature types are also serialized using the SKOS Core Vocabulary (Miles and Bechhofer, 2009) in a separate RDF dataset.

- **Feature relationships**. The serialization only asserts direct relationships. The serialized relationships include: administrative hierarchy containment, spatial proper containment, administrative and spatial adjacency, spatial connectivity, relative location, geographic scope web resources and location of the owner of a Web resource. The geographic scope and the

[11] `http://rdfs.org/sioc/spec/`

[12] `http://xmlns.com/foaf/spec/`

location are special relationships. The geographic scope of a Web resource is derived from its content (Silva et al., 2006). The geographic scope is described by means of a property *gnpt:scope* that links a feature resource in the network domain to a feature resource in the geo-administrative domain. 23,666 network features are classified as *sioc:Space*. The owner of a Web resource is described using the concept *sioc:User*, and the ownership by the property *sioc:has_owner*. Geo-Net-PT 02 identifies 22,980 owners of domains, which are classified as *sioc:User* instances. The location of the user is described using the property *foaf:based_near*.

- **Footprints**. They are serialized as *Footprint* instances in an *InformationDomain* with a human readable label, based on the label of the feature that locates, and a lineage description. The geometry description is encoded in GML. Footprints that have spatial extent can be pinpointed with a geodetic point. These characteristics are made explicit using the class *geo:SpatialThing* and the properties *geo:lat* and *geo:long*.

- **Reference systems**. They are serialized as instances of the class *ReferenceSystem*. The description of the reference system can be retrieved from a registry, such as the EPSG Geodetic Parameter Registry (`http://www.epsg-registry.org`), if the URI has the form `urn:ogc:def:crs:{authority}::{id}` where *authority* is the name of a known authority and *id* is the identifier of the coordinate reference system in its registry. The property *rdfs:Also* links the resource with a GML description of it in that registry. If the registry does not contain the description of the reference system used, it is explicitly described in the ontology. The geodetic datum and, if needed, the map projection, are encoded in GML as the value of the Geo-Net property *representation*.

- **Sources**. They are serialized as resources of type *Source* with a rich metadata description. The property *rdfs:seeAlso* links the source, when available, with an online resource where the original data can be found or is described in detail.

4.6 Content model for metadata

The development of a content model for metadata is out of the scope of this thesis. However, it makes sense to describe in this chapter the wide use of the Dublin Core model as interoperable core metadata in the Geospatial Web.

The annotation of geographic resources is based on the concept of metadata. Metadata are information and documentation that enable data to be understood, shared and exploited effectively by all users over time. As mentioned in Nebert (2004), the geographic metadata help geographic information users to find the data they need and determine how to use. One of the main goals of the creation of geographic metadata is the reuse of organization's data by publishing its existence through catalogue metadata records that conveys information about how to access and use the data (FGDC, 2000). There are different metadata standards that often only define syntactic serializations (see Nogueras-Iso et al., 2005). The most used metadata standards used for describing geospatial resources are the MARC 21 format (NDMSO, 1999), the FGDC Content Standard (MDWG, 1998), the ISO 19115/19119 (ISO/TC 211, 2003b, 2005), the Darwin Core (DCTG, 2010), the ADN (DLESE, 2004) and the MODS (NDMSO, 2005).

A simple approach to the publication and interchange of geospatial metadata in the Semantic Web is a mapping from these metadata standards to RDF data models with shared semantics. The Dublin Core Metadata Initiative (DCMI) has developed a core set of metadata elements that provides a small and fundamental group of text elements through which most resources can be described and catalogued. Using only 15 base text fields, a Dublin Core metadata record can describe physical resources such as books, digital materials such as video, sound, image, text files and web pages. Metadata records based on Dublin Core are intended to be used for cross-domain information resource description and have become standard in the fields of library science and computer science (see Greenberg et al., 2001; Nogueras-Iso et al., 2004; Payette and Lagoze, 2009). The abstract model of DCMI has evolved from a list of terms to a RDF based model. The DCMI abstract model (DCAM) has a reference model formalized in terms of the semantics of the RDF abstract model since 2005 (Powell et al., 2007). There are several geographic metadata crosswalks to the Dublin Core vocabulary. The use of well-known Dublin Core crosswalks to implement uniforms mappings from geographic metadata schemas to the RDF data model often follows these steps:

- Apply a metadata crosswalk from the original metadata schema to the Dublin Core vocabulary.
- Add additional metadata such as provenance of the record, original information model or crosswalk identification.
- Transform the Dublin Core description into a RDF data model by applying a profile for expressing the metadata terms using RDF (Nilsson et al., 2008).

The main drawback of this approach is that many of the available crosswalks were defined prior to the change of the DCAM model in 2005. That is, the crosswalks often define a mapping from complex metadata schemas to the original 15 base text fields. For example, the first two columns of 4.4 contain a standard crosswalk of ISO 19115 to 15 Dublin Core terms. Note that an ISO 19115 record can have more than 300 elements distributed in sections and subsections. The last two columns shows the corresponding DCMI recommended RDF properties and its range. Some mappings are easy to implement without loss of meaning (e.g. *data*, *title*). But other mappings, such as the mapping of the *coverage* term, are more complex. The key point is that the properties identified for the mapping in the source schema are textual properties (e.g. *OrganisationName*) or data type properties (e.g. *EX_GeographicBoundingBox*). These crosswalks were not designed as a mapping from concepts to concepts. They were designed as a mapping from text field to text field.

Table 4.4: CWA 14857: Crosswalk ISO 19115 Core – Dublin Core; the prefix *dct:* maps to the `http://purl.org/dc/terms/` namespace; the entities *Agent*, *Location*, *MediaType*, *LinguisticSystem* and *RightsStatement* of RDF property range are DCMI terms classes.

ISO 19115:2003	**DC property**	**RDF property**	**RDF range**
MD__Metadata.identificationInfo. MD__DataIdentification.credit	Contributor	*dct:contributor*	*Agent*
MD__Metadata.identificationInfo. MD__DataIdentification.extent.EX__Extent. geographicElement. EX__GeographicBoundingBox	Coverage	*dct:spatial*	*Location*
MD__Metadata.identificationInfo. MD__DataIdentification.citation. CI__Citation.CitedResponsibleParty. CI__ResponsibleParty.OrganisationName [role="originator"]	Creator	*dct:creator*	*Agent*
MD__Metadata.identificationInfo. MD__DataIdentification.citation.CI__Citation.date. CI__Date	Date	*dct:modified*	*Typed literal*
MD__Metadata.identificationInfo. MD__DataIdentification.abstract	Description	*dct:abstract*	*Plain literal*
MD__Metadata.distributionInfo.MD__Distribution. distributionFormat.MD__Format.name	Format	*dct:format*	*MediaType*
MD__Metadata.MD__Distribution. MD__DigitalTransferOption.onLine CI__OnlineResource.linkage.URL	Identifier	*dct:identifier*	*Plain literal*
MD__Metadata.identificationInfo. MD__DataIdentification.language	Language	*dct:language*	*LinguisticSystem*
MD__Metadata.identificationInfo. MD__DataIdentification.citation.CI__Citation. CitedResponsibleParty..CI__ResponsibleParty. OrganisationName[role="publisher"]	Publisher	*dct:publisher*	*Agent*
-	Relation	*dct:relation*	*Resource*
-	Rights	*dct:rights*	*RightsStatement*
MD__Metadata.dataQualityInfo.DQ__DataQuality. lineage.LI__Lineage.source.LI__Source.description	Source	*dct:source*	*Resource*
MD__Metadata.identificationInfo. MD__DataIdentification.topicCategory	Subject	*dct:subject*	*Resource*
MD__Metadata.identificationInfo. MD__DataIdentification.citation.CI__Citation.title	Title	*dct:title*	*Plain literal*
MD__Metadata.hierarchyLevel	Type	*rdf:type*	*Class*

This mapping implies a drastic lowering of the semantics conveyed in the resulting RDF model. This area has been identified as target of future research in Chapter 6.

4.7 Summary of the chapter

This chapter introduces the use of the gazetteer data model for the formalization of a minimum content model. The formalization, named Geo-Net, is an extension of the GKB metamodel. The Geo-Net ontology provides a simple and flexible framework that helps to translate in a consistent way descriptions of features found in WFS servers. The Geo-Net ontology captures the elements that define a feature: its place names, its footprints, its feature types and its relations. It is out of the scope of the thesis to detail the implementation of the transformation of descriptions of features encoded in OGC encoding standards into knowledge representation models based in the Geo-Net ontology. Nonetheless, part of the chapter is dedicated to introduce tools that help to use the Geo-Net ontology and results consequence of transformation procedures. In addition, the chapter highlights the paralelism between the role of Geo-Net as minimum content model for data, and the role of Dublin Core as de-facto minimum content model in the Geospatial Web.

Chapter 5

Linked OGC Web services

5.1 Introduction

This chapter describes an approach to the publication of the knowledge about OGC Web services that winds together OGC Services and Linked Data (Berners-Lee, 2006) in a single network. Figure 5.1 shows how the example application outlined in the introductory chapter may use the approach presented in this chapter to make available the OGC Web services found with an advanced geospatial crawler.

This approach is the result of the relative failure of OGC Web Services. The OGC envisioned that OGC Web Services would lead to the creation of an automated world wide public interoperable e-market where geospatial providers and geospatial consumers trade data and services (McKee and Kottman, 1999). This market does not exist yet although *geoportals* may be though to some extent as a replacement (Maguire and Longley, 2005).

This failure is not exclusive of OGC Services. W3C Web services, that is, Web services based in SOAP and WSDL, were conceived in the same epoch, as an opportunity led by Microsoft, IBM, BEA, Sun and SAP among other to support in the Web their products for distributed computing. They also envisioned that W3C Web services would lead to the creation of an automated world wide public e-market where service providers and service consumers trade services (Ferris and Farrell, 2003). Today, W3C Web services are mainly available for private use in enterprises tied to its *Enterprise Service Bus* (Papazoglou et al., 2007).

Although technical limitations can be the cause of the disappointing implantation OGC Web services, the motivations behind their relative failure to deliver services on the Web could be more social that technological. Public OGC Web services, as happened with W3C Web Services, are not conceived as tools that serve functionality to service consumers, and*, at the same time,* enable the communication among service consumers.

A recent work about Web Services highlights:

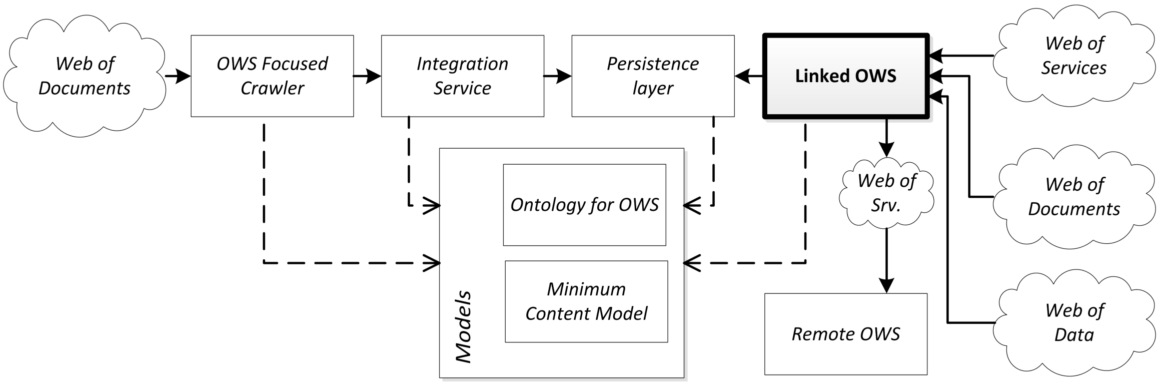


Figure 5.1: The role of the Linked OWS engine in the example application outlined in the introductory chapter.

> *The major revolution behind Web 2.0 is not on the use of particular technologies such as AJAX as initially believed, but rather on realising that, on the Web, value largely resides on the data about and the communication between people and this value is subject to the network effect.* *Pedrinaci and Domingue (2010)*

The *network effect* (Hendler and Golbeck, 2008) is the intuition that when the number of people in a network in the Web grows, the connectivity increases. Social connectivity applied to the Web implies that people can link to each other's content. Rich linked content can attract more people to the network, feeding back into the net grow of the network. The network effect has paved the way to the Social Web (e.g. Facebook), new business models based on the Web (e.g. Wikipedia) and movements for the publication data on the Web (e.g. Linked Open Data initiative).

This chapter proposes to reduce the cost of integrating OGC Web services to current applications and communities based in the network effect using a solution based in Linked Data principles. Linked Data is a consequence of the application of the principles of the Representational State Transfer architectural style (REST, Fielding, 2000). Section 5.2 presents the REST architectural style, and its derivatives including the Semantic Web, Linked Data, and the so-called RESTful Web services. Section 5.3 analyses different business cases where the use of Linked Data and RESTful Web services may help the integration and reuse of OGC Web services and its contents. Section 5.4 describes the *Linked OWS engine*, a system that uses a knowledge base that contains descriptions of OGC Web services using the ontology introduce in chapter. This engine winds Linked Data and OGC Services. Section 5.5 describes the use of this engine in two applications related with some business cases. Finally, the contents of this chapter are summarized.

5.2 Semantics and interactions in REST

The purpose of this section is the introduction of a set of concepts related with the REST architectural style for distributed hypermedia systems. These concepts include the current consensus in the semantics of the HTTP protocol, the Semantic Web, the Linked Data initiative, and hints that might help to identify a Web service as RESTful, that is, a Web service conforms with the constraints of the REST architectural style.

5.2.1 The Representational State Transfer

REST is an architectural style for *distributed hypermedia systems* strongly related with the design of the Hypertext Transfer Protocol specification versions 1.0 and 1.1 (HTTP, Berners-Lee et al., 1996; Fielding et al., 1997). The first version of REST was developed between 1994 and 1995 with the name *HTTP object model* as a means for communicating concepts during the writing of the HTTP 1.0 specification. This architectural style was iteratively improved with feedback from the rapid development of the Web.

The purpose of REST is to explain the nature of systems like the Web, and to advice in the design of applications in these platforms. From this point of view, a well-designed Web application is a network of web pages where the user progresses through the application by selecting links, resulting in the next page being transferred from a server to the application and rendered for their use. The name *Representational State Transfer* intents to describe such behaviour using state machine concepts. That is, a well-designed Web application is a virtual state machine where each link represents a state transition and each retrieved page represents a next state. The focus of the work of Fielding was the identification of the most relevant constraints to the above generic scheme of interaction that should help to maximize performance, scalability, simplicity, modifiability, visibility, portability and reliability of a distributed hypermedia system.

A REST system is a distributed platform composed by *origin servers*, *gateways*, *proxies* and *user agents* that transfers *representations* of *resources* with *hypermedia* links between *user agents* and *origin servers* making use of *gateways* and *proxies* as intermediates (see Table 5.1 and Figure 5.2). At any particular time, a *user agent* can either be in transition between application states or being in an application state. A *user agent* in an application state is able to interact only with its user. The *user agent* begins sending requests to *origin servers* when it is ready to make a transition to a new application state. The *origin servers* process the requests and return appropriate responses. The *user agent* is in transition between application states meanwhile there are outstanding requests. The interaction involves the transfer of representations of resources whose identifiers are within the namespace governed by the servers. The *user agent* should assume that the *representation of a resource* returned by an *origin server* captures the state of the resource when it was requested. REST describes a hypermedia system. Therefore, the representation may contain references, i.e.

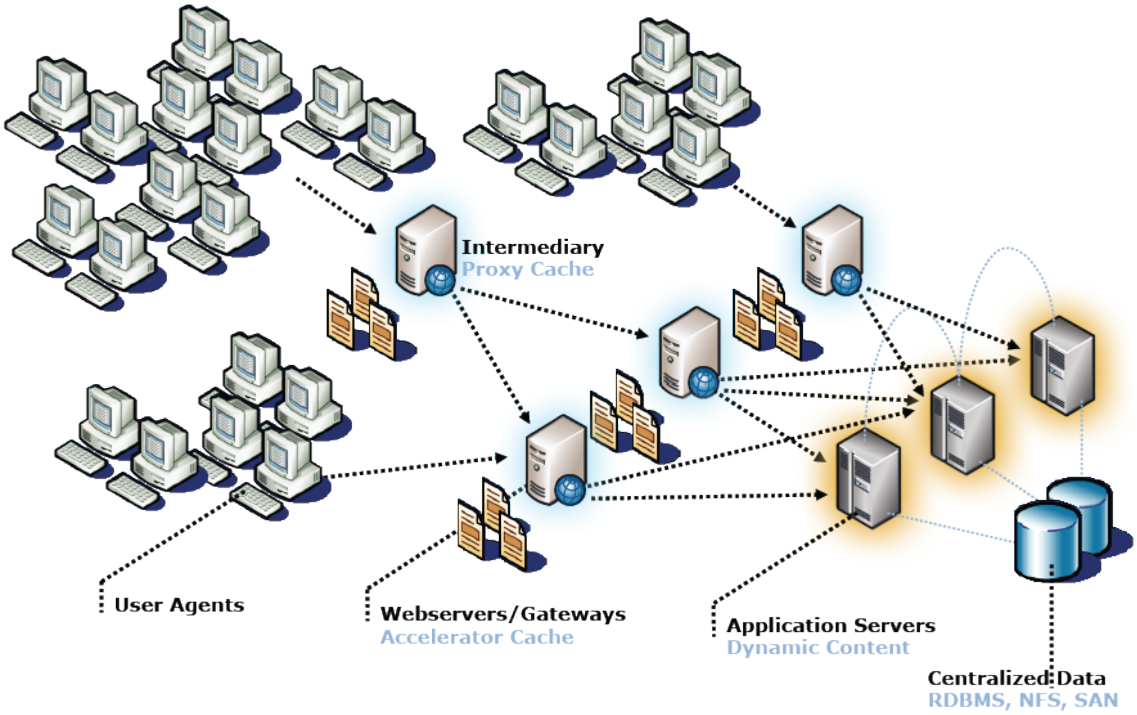


Figure 5.2: Web implementation of the REST architectural style (Fielding, 2008b).

hypermedia links, to resources that can be requested next time the *user agent* decides to initiate a new transition.

The analysis of Fielding concluded that only six constraints were relevant in the design of the distributed hypermedia system (see Table 5.2). First, the design should maintain a clear separation of client and servers. Next, the communication between clients and servers should be stateless. Then, proxies and gateways should offer cache services. Following this, the interaction interface should be uniform across the design. Next, each component should have knowledge only of the immediate layer. Finally, the system should allow the optional use of code on demand.

The REST architectural style was used to identify problems with HTTP/1.0, and has influenced the design of HTTP/1.1 and revisions of the URI specification (Berners-Lee et al., 2005). However, HTTP/1.1 presents some mismatches respect to the REST architectural style (e.g. cookies, HTTP headers can be extended at will), and the URIs syntax cannot avoid that a URI could encode session state. That is, there are distributed systems that can use the Web, HTTP and URI in ways that do not match the REST model. In fact, REST is a just summary of properties of the Web that are considered central in its design that when added to the design of a distributed system should result in optimum behaviour when the system is deployed in the Web.

Next sections will introduce topics related with REST about the intended semantics of URIs and

Table 5.1: Key concepts for REST architectures (Fielding, 2000).

Concept	Definition
user agent	A *user agent* (or *client*) initiates a request about a resource and becomes the ultimate recipient of the response.
proxy	A *proxy* is an intermediary selected by a *client* to provide encapsulation of other services (e.g. cache).
gateway	A *gateway* (or *reverse proxy*) is an intermediary imposed by the network or the *origin server* that provides encapsulation of other services.
origin server	An *origin server* governs the namespace for a resource, that is, it is the definitive source for representations of the resource and must be the ultimate recipient of any request that intends to retrieve a representation or modify the value the resource.
resource	A *resource* is a conceptual temporal mapping to a set of values that are considered equivalents at a time t. The values of the mapping are *resource representations* and *resource identifiers*.
representation	A *representation* is a sequence of bytes, plus representation metadata to describe those bytes.
identifier	An *identifier* that identifies a *resource* which some resolver can translate partial or complete into a network address managed by an *origin server*.
hypermedia	An *hypermedia* system is a non-linear system concerned with the location of information, performing information requests and rendering information where the information items contains navigable references to other information items.

the HTTP protocol, expressing meaning in the Web and REST interactions. These topics are the technological and semantic basis for the publication and access platform to information about OGC Web services and contents described in this chapter.

5.2.2 Resource oriented semantics

The formalization of URI and HTTP semantics is an open question with relevance to application design, interface semantics, resource description, testing, accountability, authorization, trust, provenance, and so on. There is a sparkling debate on the semantics of HTTP requests, HTTP responses, resources, URIs (as identifiers), and representations. The debate is around which is the most accurate interpretation of the different definitions of these concepts in the HTTP/1.1 (Fielding et al., 1997), the specification of the URI syntax (Berners-Lee et al., 2005), and in the works of the W3C's Technical Architecture Group (TAG). Two TAG documents are widely cited as informal reference of semantics: the *Architecture of the World Wide Web* (AWWW, Jacobs and Walsh, 2004), and the compromise resolution about the range of the HTTP dereference function (httpRange-14, Fielding, 2005). The ongoing revision of HTTP, also known as HTTPbis (Fielding et al., 2010), is trying to

Table 5.2: Key constraints for REST architectures (Fielding, 2000).

Constraint	Definition
client-server	Separation of concerns between the client and the server in the sense that the server is the only responsible for the resources.
stateless	Session state is entirely kept on the client.
cache	The server or the client must label implicitly (e.g. HTTP constraints) or explicitly as cacheable or non-cacheable the data within the response to a request.
uniform interface	The interface is simple, visible, uniform and decoupled from the services provided, and the information is transferred in a standardized form using resource identifiers, resource representations, representation metadata, and cache control data.
layered	A client cannot discovers whether it is connected directly to an *origin server*, or to an intermediary along the way.
code on demand	The representation of a resource may be code that extends the functionality of the client; this is an optional feature in the REST architecture in the sense that proxies and gateways may prevent the transfer of code.

resolve the interpretation of the basic message semantics. This section summarizes current agreements and disagreements on the URI and HTTP semantics in the TAG that are relevant for this chapter:

- **Identity crisis**. In the AWWW document, the term *resource* has an unlimited scope and includes "real things". The term *information resources* identify those *resources* that all of their essential characteristics (from the point of view of the *origin server*) can be conveyed in a message. For example, a document located by an URI is an information resource. As URIs can identify resources and information resources, how a machine can know if the URI identifies a real thing or a document? This issue was not clearly addressed in the HTTP/1.1 and required the following clarification (httpRange-14): given a GET request, if the response status code is success (2xx class), then the *resource* identified by the effective request URI is an *information resource.* However, if the response status code is *303 See Other*, the *resource* identified by the URI could be any *resource (thing).* Finally, if the response status code is error (4xx class), the nature of the *resource* is unknown. Logicians and Web architects do not have reached a satisfying solution yet. This issue it is far from being considered closed (see Halpin and Presutti, 2009).

- **Redirection semantics**. In HTTP/1.1, a redirection status code (3xx) indicates that a second request needs to be taken by the user agent in order to fulfil the request. There are three relevant cases: *permanent redirection* (*301 Moved Permanently*), *temporal redirection* (*302 Found*, *307 Temporary Redirect*) and *resource without representation* (*303 See Other*).

Permanent redirection means that the target resource has been assigned a new permanent URI, indicated by a URI in the *Location* header field, and any future references to this resource should use the new URI. Temporal redirection means that the target resource resides temporarily under a different URI, indicated by a URI in the *Location* header field, the second request should use the temporal URI, and the user agent should continue to use the effective URI for future requests. Resource without representation means that the origin server directs the user agent to a different resource, indicated by a URI in the *Location* header field. HTTPbis clarifies that the *Location* URI indicates a resource that is descriptive of the target resource. In permanent and temporal redirection, the *Location* URI is a substitute reference for the effective request URI. The *Location* URI in resource without representation is not a substitute reference for the effective request URI.

- **Methods semantics**. The main methods of HTTP are GET, POST, PUT and DELETE. These methods represent very abstract operations: The GET method retrieves a representation of the target resource; the PUT method stores content at the effective request URI; and the DELETE method requests that the origin server delete the target resource. The definition of the POST method in HTTP/1.1 raised interpretation doubts: "*the origin server accept the entity enclosed in the request as a new subordinate of the resource*". *Subordinate* here includes annotations, blocks of data in a data-handling process, and appending a database. HTTPbis clarifies the definition: "*the origin server accept the representation enclosed in the request as data to be processed by the target resource*". In HTTP/1.1 and HTTPbis, the server determines the actual function performed by the POST method. A GET request is expected to be *safe*, that is, it should not involve a modification of the state of the resource managed by the origin server accountable to the user agent. This does not mean that a GET request should never have side effects. It is possible to consider, for example, a resource that represents an auto-increment counter that is updated in each GET request. In such cases, the side effects are accountable to the nature of the resource. The methods PUT, DELETE, and all safe methods are *idempotent*. The property of *idempotence* means that, aside from error or expiration issues, the effect of multiple identical change requests of the same resource do not change its state. Table 5.3 summarizes the characteristics of the main methods of the HTTP protocol.

5.2.3 Expressing meaning in the Web

The Semantic Web (Berners-Lee et al., 2001) is an initiative led by W3C, with participation from a large number of researchers and industrial partners, whose goal is the development of a common framework that should allow data to be meaningfully shared and reused across application, enterprise, and community boundaries using the Web as platform. The Web was initially designed as a Web of documents. Humans can use the Web of documents to carry out complex tasks, such as the discovery of a web mapping service with information about an emergency disaster, or the finding the

Table 5.3: Main methods of the HTTP protocol.

Method	Idempotent	Safe	Definition
GET	✓	✓	The *user agent* requests the *origin server* returns whatever information (in the form of a representation) currently corresponds to the target *resource*.
POST	✗	✗	The *user agent* requests the *origin server* accepts the representation enclosed in the request as data to be processed by the target *resource*.
PUT	✓	✗	The *user agent* requests the *origin server* accepts the representation enclosed in the request as data to be stored at the target *resource*.
DELETE	✓	✗	The *user agent* requests the *origin server* deletes or moves to an inaccessible location the target *resource*.

Spanish word for *geographic feature*. User agents can perform these activities under the direction of their users because the Web content is mainly composed by human-oriented resources (e.g. Web pages, images, movies, music). The early vision of the Semantic Web is a Web where the information can be interpreted by user agents, that is, machine processable with formal semantics, so user agents can performs tedious tasks on user's behalf returning the same results that a user directed work. This idea remained largely unrealized (see Shadbolt et al., 2006). Today, the Semantic Web has found a niche named Linked Data related with the publication as graph, the augmentation with links to other resources, and the reuse of large datasets (e.g. sensor information, personal data, bibliographic records).

The research program of the Semantic Web is organized in a stack known as the *layer cake* (Figure 5.3). This stack comprises *resource identification, resource representation, data model, ontology languages, query languages, rule languages, unifying logic, proofs, cryptography, trust, user interfaces and applications*. The bottom layers of the semantic stack are *resource identification* and *resource representation*, which are closely related with the REST architectural style. The URI specification, and its internationalized version IRI (Duerst and Suignard, 2005), provides a mean for uniquely identifying resources. W3C standardized *resource representation* syntaxes encode syntactically resource descriptions. Traditionally, XML was *the* representation syntax in the Semantic web. Today, there are alternatives not based in XML (e.g. OWL 2 Functional-Syntax, Motik et al., 2009a) or based in annotation formats (e.g. RDFa, Adida et al., 2008). The relation between representation and resource using the HTTP protocol in the context of the Semantic Web is the cause of the identity crisis described in the previous section.

The middle layers of the semantic stack have been materialized in W3C recommendations. The *data model* is the *Resource Description Framework* (RDF, Carroll and Klyne, 2004), which enables the representation of resources as statements in the form of subject-predicate-object expressions.

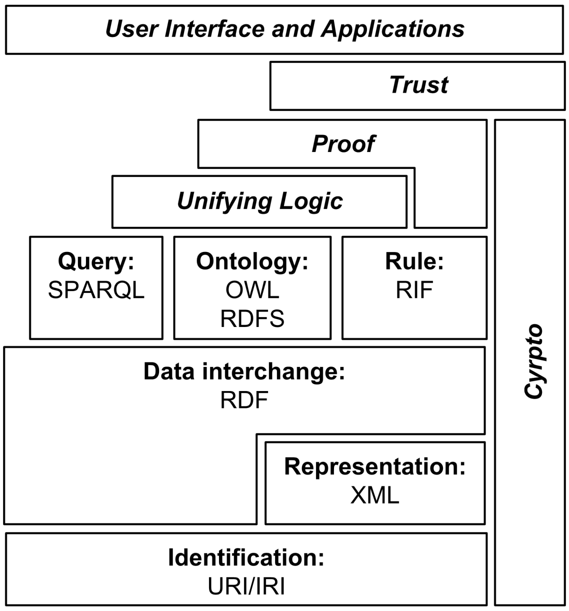


Figure 5.3: The semantic layer cake (circa 2010)

The nodes of the graph are resources, named or blank, and values, also known as literals. Each named node has an associated URI that uniquely identifies the node. The rules of the arcs, known as triples, are:

- The subject, that is, the origin of the arc, is a resource.
- The property or predicate, that is, the label of the arc, is a named re-source.
- The object, that is, the target of the arc, is a resource or a literal.

There are two kinds of literals: plain and typed. A plain literal is a character string that optionally has a tag that documents the language of the character string. A typed literal is a pair composed by a value encoded as a character string, and the data type, which defines both the semantics of the value and the syntax of the encoding. *Ontology languages* provide the constructs required to interpret the statements. The *RDF Schema* (RDFS, Guha and Brickley, 2004) makes possible to create hierarchies of classes and properties. The family of Web Ontology Languages (OWL1, Welty et al., 2004, OWL2, OWL WG, 2009) adds additional constraints (e.g. cardinality, value restriction, transitivity). The family of knowledge description languages named Description Logics (Baader et al., 2003) has influence heavily the design of OWL languages. *Query languages* allow querying any RDF-based dataset. SPARQL (Seaborne and Prud'hommeaux, 2008) was recommended by W3C as query language in 2008. *Rule languages* allow describing things that cannot be described or computed efficiently with ontology language construct. The Rule Interchange Format (RIF, Paschke et al., 2009) is a language that allows the interchange of different kind of rules (logic rules, positive datalogs, production rules).

Top layers of the semantic stack contain technologies that are not yet recommended by W3C, proofs of concepts and ideas. These technologies include a *unifying logic* that relates the semantics of data model, query languages, ontology languages and rule languages. *Proof* languages will check if the statements made are true or false. *Trust* engines will use *cryptography* and inference engines to ascertain our belief in the information. The final layer will contain W3C recommendations about the use of Semantic Web technologies in *user interfaces and applications*.

The application of the Semantic Web is the Web of Data, a web of things in the world, described by data on the Web. However:

> *The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data.* Berners-Lee (2006)

Berners-Lee (2006) has recommended a set of non-normative rules that have become known as the Linked Data principles:

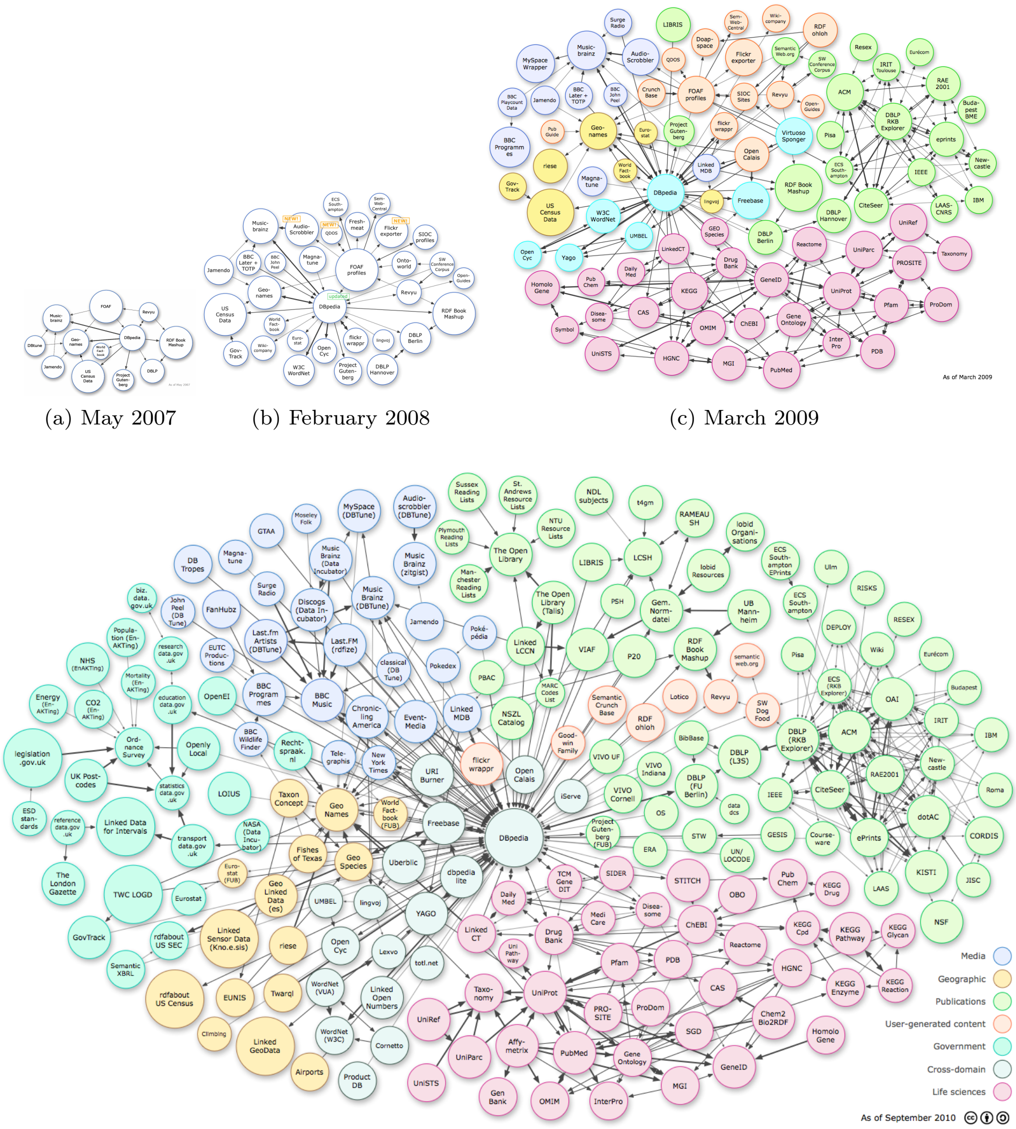
1. Use URIs for naming things. If an application or system does not use URIs for naming things, it is outside of the Semantic Web.

2. Use HTTP URIs so that people can look up those names. The use of an alternative naming authority not based in the established Domain Name System adds an extra level of complexity to the name lookup. This principle disallows the use of domain URI schemas, such as the OGC URI schema (Reed, 2008).

3. Provide useful machine-readable representation when a user agent dereferences a HTTP URI used for naming things. That is, the origin server should return the information at least in a resource representation syntax standardized by W3C. Even if the HTTP URI identifies a resource part of a large dataset that can be queried using a SPARQL endpoint, the HTTP URI should provide useful information.

4. Provide hyperlinked machine-readable representation so the user agent can discover more things. This is a condition necessary for building an effective Web of Data.

An active community has soared around these concepts. The vision of the Linked Data community is the use of the Web to connect related pieces of data, information, and knowledge not previously linked, or linked using other methods. The Linked Data community has gained momentum since its debut in 2007 (see Bizer et al., 2009). For example, Figure 5.4 shows the evolution of the datasets published as Linked Data by contributors to the *Linking Open Data* community project and other individuals and organisations, and the RDF links among them. Cyganiak and Sauermann (2008),



(a) May 2007 (b) February 2008 (c) March 2009

(d) September 2010

Figure 5.4: Evolution of the Linking Open Data cloud diagram, by R. Cyganiak and A. Jentzsch. `http://lod-cloud.net/` The size of circles correspond to the number of triples in each dataset; very large circles >1B, medium 10M-500K, very small < 10K.

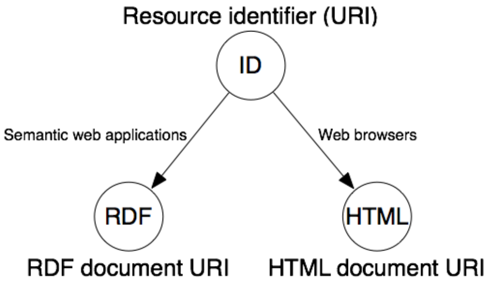


Figure 5.5: Content negotiation of non-information resources (Sauermann and Cyganiak, 2008).

Table 5.4: Parts of an information resource; contents of an information resource named URI_A that describes the resource named URI_B.

Content	Definition
resource description	A set of RDF statements where URI_B is the subject.
backlinks	A set of RDF statements where URI_B is the object.
related descriptions	Any additional information as RDF statements about related resources.
metadata	A set of RDF statements where URI_A is the subject.

Berrueta and Phipps (2008) and Bizer et al. (2007) contain the best practices of the Linked Data community. These best practices clarify how to retrieve the a machine-readable representation in RDF of a resource named with a HTTP URI, and how to avoid problems of ambiguity due to the *identity crisis*. Publishing descriptions of resources as Linked Data means (see Figure 5.5):

- If a user agent dereferences a HTTP URI about an information resource asking for the MIME-type application/rdf+xml, the origin server must return a RDF/XML document that includes a description of the resource, e.g. a set of statements where the HTTP URI is subject.

- If a user agent look up information about a resource identified by a HTTP URI that contains a fragment identifier, the user agent will dereference the HTTP URI without the fragment identifier. The origin server must return a RDF/XML document that includes a description of the resource using the HTTP URI with the fragment identifier.

- If a user agent look up information about a resource identified by a HTTP URI and the origin server returns a response with a HTTP *303 See Other* redirect to a new location, the location contains a description of the resource, e.g. a set of statements where the HTTP URI is subject, with a description of the location.

The best practices recommend that the information resource that contains a description about other

Table 5.5: Data representations.

Format	Media type	Intended users
RDF/XML	application/rdf+xml	Any machine
N3	text/n3	Any machine, casual human reader
Turtle	text/turtle	Any machine, casual human reader
N-triples	text/plain	Any machine (bulk load of data)
XHTML + RDFa	application/xhtml+xml	Browser, user interface

resource should also include backlinks, related descriptions and metadata (see Table 5.4). The representation should be also available for both machines and humans as serialized data in processable formats, such as Turtle (Beckett and Berners-Lee, 2008), or embedded in Web pages using processable annotation schemes, such as RDFa (Adida et al., 2008). Table 5.5 shows some formats that can represent descriptions. The description of resources should use existing vocabularies (e.g. FOAF, SIOC, SKOS, DOAP, vCard, Dublin Core, OAI-ORE or GoodRelations) complemented with additional terms only if these vocabularies do not provide the required terms. It is also recommended that representations and description of additional terms should contain "prose" properties (e.g. *rdfs:comment*, *rdfs:label*) for their use in information retrieval systems. RDF links with other published datasets allows clients to navigate to other datasets. The generation of RDF links is usually automated or semi-automate and can be based on domain identifiers and the similarity of entities.

A variety of tools can help in the publication and discovery of Linked Data. Many of them are services that publish the content of relational databases as Linked Data (Bizer et al., 2009; Hausenblas, 2009a). Large geographical information providers are investigating how Linked Data and other Semantic Web technologies can assist the diffusion of geographic data. For example, Ordnance Survey[1] is developing datasets in RDF and publishing them using the Linked Data principles (Goodwin et al., 2009). The Linked Data community has an increasing interest in the geospatial databases. In particular, the LinkedGeoData project maps OpenStreetMap data into linked data (Auer et al., 2009), and the GeoNames ontology describes the content of the GeoNames database (Vatant and Wick, 2007). A different publication approach is the use of Linked Data wrappers of existing Web APIs (Langegger et al., 2008; Haslhofer and Schandl, 2008). In general, a Linked Data wrapper provides the next functionalities:

- If a user agent dereferences an information resource managed by the wrapper asking for the MIME-type application/rdf+xml, the wrapper rewrites the information resource URI into a request against the underlying API. The wrapper can retrieve the response from a cache. The

[1] `http://data.ordnancesurvey.co.uk/`

results to the API request are mapped to a RDF model and sent back to the user agent. The results may be cached for further use.

- If a user agent look up information about a non-information resource managed by the wrapper, the wrapper returns a response with a HTTP *303 See Other* redirect to an information resource managed by the wrapper. Optionally, the wrapper could rewrite the non-information URI into a request against the underlying API whose results are cached for further use.

A Linked Data wrapper can be thought as a type of RESTful Web Service. RESTful Web Services are presented below and its discussion ends this overview on REST related technologies.

5.2.4 RESTful Web services

A *RESTful Web service* (Richardson and Ruby, 2007) is a Web service implemented according to the principles of the REST architectural style. RESTful Web services have attained increased attention since 2005 by three main reasons: they are the basis of many Web 2.0 APIs (e.g. Amazon.com, eBay, Facebook, Yahoo!, Youtube), there are good chances for serendipitous service reuse (Vinoski, 2008), and many developers repeat the mantra that RESTful Web services are simpler than W3C Web Services (SOAP, WSDL, WS-*) (see the fair review of Pautasso et al., 2008). This repeated mantra has started to influence the OGC Web Services community (Lucchi and Elfers, 2008; Reed, 2009).

Many applications on the Web claims to be RESTful just because all its resources are accessible through an HTTP GET requests, and can be updated through an HTTP POST request. Table 5.6 contains a list of constraints that a RESTful Web service should not violate. Richardson and Ruby (2007), Pautasso et al. (2008) and Fielding (2008a) provide additional information about RESTful Web services. Next, two RESTful Web services that publishes Linked Data are analysed: Pubby and the OAI2LOD Server.

Pubby. Pubby is a Linked Data frontend for local and remote SPARQL endpoints developed in the Freie Universität of Berlin (Cyganiak and Bizer, 2007). Many triple stores offer SPARQL endpoints that can be accessed only by SPARQL client applications. A triple store is a purpose-built database for the storage and retrieval of RDF data. The content of many of these triple stores never has been published as Linked Data or use HTTP URIs for naming resources. The configuration of Pubby defines a simple URI mapping that translate the URIs used into the triple store to a dereferenceable HTTP URIs handled by Pubby. If the server is running at `http://www.example.com/pubby`, then Pubby handles the next list of mappings (see Table 5.7):

Table 5.6: Constraints for RESTful services.

Constraint	Definition
identification	Each identifiable unit of information managed by the RESTful service carries a URI name. The access to the resource is neither guarantee nor implied by the presence of the URI.
functional decoupling	The clients of the RESTful service are instructed dynamically on how to construct appropriate resource URIs (e.g. a HTLM form, a URI template). The client does not require a prior knowledge of the resource names and hierarchies. This does not mean that the URIs should not to be cool. It means that a change in the naming policy should not break the clients.
self-descriptive information	The state of tan application client evolves on the understanding of the media types, link types and mark-up provided in resource representations. The web service documentation should provide documentation about their own media types, vocabulary for defining relations between resources, and mark-up for existing media types.
out-of-band information	The application client must extra ignore information outside the flow of information between the client and the RESTful service, such as procedure or interface declarations.
intermediate friendly	The application clients and the RESTful service make use of the layered architecture of the Web.
uniform interface	The RESTful service should not contain any changes to the HTTP protocol (or the standardized representations) aside from the clarifications to underspecified details (e.g. httpRange-14, Fielding, 2005). For example, GET reads the state of a resource, PUT creates or overwrites a resource, DELETE requests to delete a resource, and POST sends data to be processed by a resource (e.g. inputs for an algorithm).
low entry	The application client requires no prior knowledge of the service beyond the media types, link types and mark-ups used by the RESTful service, and the initial URI of the RESTful service or a bookmarked URI of a resource managed by the RESTful service.
hypermedia	Decisions based in the received representations (e.g. a RDF link), enhanced by client application's manipulations (e.g. from applying rules to a RDF model) or improved by code on demand received from the server (e.g. a JavaScript library) must drive all state transitions of the client application.

Table 5.7: Description of Pubby supported requests.

Pubby request	Semantics	SPARQL req.	Pubby resp.
`/resource/{identifier}`[a]	A resource	–	303 See other
`/page/{identifier}`	A description of a resource	Describe	HTML
`/data/{identifier}`	A description of a resource	Describe	RDF

[a] The variable *identifier* represents a string that concatenated with a given prefix identifies resources within a SPARQL endpoint.

- **Non-information resources**. Pubby performs a *303 See Other* redirect of HTTP URIs identified as non-information resource identifiers (e.g. `http://www.example.com/pubby/resource/WMS-Cantabria`) to the appropriate information resource depending on the MIME type requested.

- **Information resources whose representations are machine processable.** If the client asks for a representation in RDF/XML or N3 format (e.g. `http://www.example.com/pubby/data/WMS-Cantabria`), Pubby maps the URI into the original URI in the triple store, queries to the SPARQL endpoint information about the original URI, translate the URIs in the response to dereferenceable URIs, and returns the information in the requested format. If the description is empty, Pubby returns a *404 Not found* response.

- **Information resources whose representations are human readable**. If the client asks for a representation in HTML (e.g. `http://www.example.com/pubby/page/WMS-Cantabria`), Pubby does the same as above but uses a template engine for rendering a HTML response document.

The configuration of Pubby requires the definition of a resource in the triple store that will act as index, that is, a request to the URI of an instance of Pubby will forward to that resource. Pubby's clients do not require a prior knowledge of the organization of the mappings. They only require the knowledge of a few media formats (HTML, RDF/XML, N3), the location of an instance of Pubby, and a HTTP GET client with support to redirections.

OAI2LOD Server. The OAI2LOD Server is a Linked Data frontend for the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH, Lagoze and Van de Sompel, 2002) developed at the University of Vienna (Haslhofer and Schandl, 2008). This protocol is utilised for the exchange and sharing of metadata in the domain of digital libraries and archives. To guarantee a basic level of interoperability, all OAI-PMH instances must support the unqualified Dublin Core vocabulary and return the metadata in XML. OAI-PMH provides the concept of *Item*, i.e. resource, and the

Table 5.8: OAI-PMH verbs wrapped by OAI2LOD.

Verb	**Description**	**Mandatory parameter**
GetRecord	This verb is used to retrieve an individual metadata record from a repository.	identifier[a], metadataPrefix[b]
ListIdentifiers	This verb is retrieves only headers, which contains the metadata identifiers.	metadataPrefix
ListSets	This verb is used to retrieve the set structure of a repository	–

[a] identifier: A unique identifier unambiguously identifies an item within a repository in URI syntax.

[a] metadataPrefix: The value *oai_dc* identifies the metadata schema for unqualified Dublin Core.

concept of *Set* for grouping related resources and their associated metadata. URIs identify *items* and strings identify *sets* within a repository.

OAI2LOD server provides resolvable HTTP URLs for *items* and *sets*. The description of a resource of set of resources wraps a request to an OAI-PMH instance. Three OAI-PHM operations are wrapped: *ListSets*, *ListIdentifier* and *GetRecord* (see Table 5.8). OAI2LOD transform the XML responses into a RDF model using the Dublin Core vocabulary, and a specific vocabulary for OAI-PMH specific concepts, such as *Item* and *Set*. Table 5.9 shows the requests supported by OAI2LOD and its mapping to OAI-PMH concepts and requests.

The OAI2LOD Server has two major purposes: first, it exposes the metadata entities of the OAI-PMH repositories as dereferenceable Web resources, and second, it provides selective access to metadata via a SPARQL endpoint. This approach makes OAI-PMH metadata accessible not only by Liked Data clients but also for Web clients not being aware of the OAI-PMH protocol.

Pubby and OAI2LOD Server are precursor of the ideas related with the publication of content of this chapter.

5.3 REST publishing of spatial resources

Linked Data and *RESTful Web services* can be applied in a number of different situations. In the context of this thesis, the key to getting value of *Linked Data* and *RESTful Web services* is to identify the reason why they are good for unearthing the spatial resources hidden in the Web. Below are listed a number of generic scenarios known to be successfully delivered by *Linked Data* and *RESTful Web services*. For each scenario it is identified the reason why these technologies are a good fit for implementing the capability. Alternative technical approaches are also described.

Table 5.9: Description of OAI2LOD supported requests.

OAI2LOD request	**Semantics**	**OAI-PMH req.**	**OAI2LOD resp.**
`/`	Links to server contents	–	HTML
`/all`	Links to server contents	–	RDF
`/directory/Set`	The list of sets	ListSets	HTML
`/all/Set`	The list of sets	ListSets	RDF
`/resource/set/{string}`[a]	A set	–	303 See other
`/page/set/{string}`	The description of a set	ListSets	HTML
`/data/set/{string}`	The description of a set	ListSets	RDF
`/directory/Item`	The list of items	ListIdentifier	HTML
`/all/Item`	The list of items	ListIdentifier	RDF
`/resource/item/{identifier}`[b]	An item	–	303 See other
`/page/item/{identifier}`	The description of an item	GetRecord	HTML
`/data/item/{identifier}`	The description of an item	GetRecord	RDF

[a] The variable *string* represents a valid string consisting of URI unreserved characters that identifies a *set* within an OAI-PMH instance.

[a] The variable *identifier* represents a URI that identifies an *item* within an OAI-PMH instance.

Concept-based search

Description: A concept-based search uses information rather than data to search resources across multiple geospatial datasets. Geospatial catalogue applications often use thesauri and spatial maps to capture the semantic intent of the user.

Benefits: The transformation of geospatial assets into RDF models allows the creation of RDF links among the geospatial assets, external knowledge sources and reference ontologies (e.g. OntoOWS). The availability as Linked Data simplifies the development of a concept-search engine.

Alternatives: Geospatial thesauri, topic maps and map-based queries could provide an approximation to concept-based search. Geospatial standards do not provide support to concept-based search.

Contribute data to research

Description: A collection of service metadata, a collection of geographic metadata records or a collection of features may be published as data contributed to the research community. That is, they are published in a way that allows an analysis from different point of views. For example, an information systems researcher may investigate the completeness of the description from an

information retrieval perspective. A geospatial researcher may be interested in the thematic and the spatial distribution of the available information.
Benefits: Linked Data not only simplifies the distribution of information but also conveys the information uniformly in self-described documents. Metainformation, such as provenance, can be part of the document. There is no consensus on how to represent complex geographical data in RDF. Lopez-Pellicer et al. (2010d) proposes that the user agent can negotiate a response in a geospatial format (e.g. a redirect to an OGC service) as a solution for this issue. The researcher is responsible for the extraction of the required views from the Linked Data.
Alternatives: The publisher should create a download point where research data can be downloaded only in a few formats (e.g. the geospatial catalogue of Data.gov[2]). Among these few formats are the standard GML is a format that can encode many of the available geospatial information, and the proprietary ESRI Shapefile.

Contribute data to search engines

Description: A geospatial data provider can award search engines with a privileged access to its assets. The principal search engines (e.g. Google, Yahoo!, Ask.com, Bing, Yandex and Baidu) use the Sitemap protocol (Google, Inc. et al., 2008) for the inclusion of documents. Search engines offers hints to transform portals into search-friendly sites and supports domain publishers to ease the access to its contents (e.g. Google Scholar). Once the resource is indexed, the provider gains in discovery and access to its assets.
Benefits: Linked Data simplifies the crawling of resources. Linked Data wrappers can give access to the contents accessible through geospatial services. Dereferenceable URIs can be advertised using the Sitemap protocol. The semantic Sitemap extension (Cyganiak et al., 2007) developed for semantic search engines can include RDF dumps and SPARQL endpoints. Some non-sematic search engines, such as Google, have begun to index RDFa annotations in XHTML if these annotations follow some conventions.
Alternatives: Transform the geoportals into search engine friendly sites. Create applications that enable a navigational search of geospatial resources.

Crowdsourcing content

Description: A geospatial data provider may supply of data and metadata to volunteers for detecting errors, add new data and improving and enhancing descriptions. Crowdsourcing semantic tags allow a resource be better understood so that search, use or invocation becomes more effective.
Benefits: Linked Data provides dereferenceable URIs that can be reused elsewhere to identify a resource. If crowdsourcing data updates are republished again as Linked Data, the provider can integrate this information. Linked Data enables crowdsourcing edition but not provides yet a solution

[2] `http://www.data.gov/catalog/geodata`

without an additional API (e.g. see Freebase, LinkedGeoData). Without an additional API, the crowdsourcing activity with Linked Data is opportunistic, i.e. volunteers edit whatever they want when they want.
Alternatives: The creation of a full-fledged crowdsourcing initiative oriented to specific information, or generic, with the support of an edit application. The success of such initiative will involve the organization of a Web community (e.g. GeoNames, OpenStreetMap). Geospatial standards do not give support to these activities.

Enhanced data

Description: A reason is to enable to connect geospatial data with related data available on the Web, and use the Web to link data currently linked using other methods. This allow to data to be shared and understood across a variety of scenarios.
Benefits: One of the principles of Linked Data is to provide links to relevant data. Many Linked Data tools give support to automatic or semi-automatic generation of links within the information published (e.g. ODEMapster – Barrasa-Rodriguez and Gómez-Pérez, 2006) and with external sources (e.g. Silk – Volz et al., 2009).
Alternatives: Keywords used in geospatial metadata are often an unordered list of one or more commonly used or formalised word(s) or phrase(s) used to describe the service. Formalised here means a reference to a name authority and/or thesaurus that should contain them.

First Mover

Description: A public organization playing a leadership role in a Spatial Data Infrastructure or a research institute may be a first mover in the use of a technology with geospatial data in relation to the discovery and access. Being a first mover in the publication of geospatial resources may generate publicity and status derived from the activity.
Benefits: Linked Data and RESTful Web services are today on the edge of the technological hype (e.g. LinkedGeoData – Auer et al., 2009, GeoLinked Data – Vilches-Blázquez et al., 2010, Linked Data in SDI – Schade et al., 2010).
Alternatives: Today, a first mover organization could invest in the research of hype technologies, such as cloud computing and the social Web, applied to the Geospatial Web.

Lightweight application development

Description: Geospatial resources deliver added value when reused within the systems of the provider and far beyond its limits. As maps in a travel application, as data in an environmental portal, or as reference data in neogeography initiatives (see Turner, 2006), geospatial resources deserve to be used in new and unexpected manners in third party applications.

Benefits: Linked Data give uniform and simple access to the contents exposed. Site-specific APIs, such as geospatial web services, can be integrated with Linked Data wrappers (see Berners-Lee et al., 2009).

Alternatives: Raw data and geospatial Web services are not prone to be reused in third party applications due to its domain constraints. An alternative approach is the publication of simple APIs and libraries that are oriented to the mainstream developers (e.g. Open Space API[3] provided by Ordnance Survey, UK, and API Géoportail[4] provided by Institut Geographique National, France).

Lightweight in-house application development

Description: A geospatial provider may develop applications on top of its services that access its data. Geospatial standards were developed targeted to domain specific complex applications. However, there are scenarios where a simple application (e.g. a mashups) can fulfil better the requirements (e.g. short development time).

Benefits: An engine aware of the context (see Hausenblas, 2009b) can control the generation of human readable representations of Linked Data. Requests to Geospatial Web services can be integrated in the assembly of the human readable representation (e.g. a Web map service). Linked Data wrappers can give access to other geospatial contents.

Alternatives: There are available libraries (e.g. OpenLayers[5], Mapnik[6]) and development platforms (e.g. GeoDiango[7], GeoMajas[8]) oriented to the production of lightweight Web geospatial applications.

Pattern explorer

Description: A geospatial analyst often uses disparate information to explore and discover patterns present in the data (e.g. geomarketing). A search for patterns involves the analysis of the different connections and the exploration of the data patterns related with these connections.

Benefits: Linked Data not only provides a networked view of the data but also formal semantics and navigable connections with external resources. Inference over Linked Data and graph related measures might help to identify graph patterns.

Alternatives: Statistical algorithms are more suitable when the geospatial data is continuous. These analyses are limited to the dataset under analysis. OGC Web services, such as WPS, can wrap statistical analysis.

[3] `http://openspace.ordnancesurvey.co.uk/openspace/`

[4] `https://api.ign.fr/geoportail/index.do`

[5] `http://openlayers.org/`

[6] `http://mapnik.org/`

[7] `http://geodjango.org/`

[8] `http://www.geomajas.org/`

Publish Open Data

Description: It is often in relation to the mission and strategic goals of the publisher in reference to the dissemination of geospatial information. There is a movement to make public sector geospatial services, data and metadata accessible and free. For example, many geospatial data collected by the USA Federal government is available at no cost and is in the public domain. Geospatial datasets are free in some countries (e.g. Canada, South Africa, Netherlands, Spain) with different degrees of freedom in their license.

Benefits: Linked Data enables the use of the published geospatial information by third party Web applications without requiring to known geospatial standards (e.g. see Ordnance Survey OpenData initiative – Goodwin et al., 2009). A certain degree of semantic interoperability is possible if the description uses well-known vocabularies used in the description of collections of metadata (e.g. Dublin Core – Nilsson et al., 2008, SKOS – Miles and Bechhofer, 2009, Data Catalog Vocabulary – Cyganiak et al., 2010). Additionally, the data is also available as RDF dumps and accessible through a SPARQL endpoint.

Alternatives: To publish in the geospatial portal the geospatial data in a variety of proprietary formats, such as ESRI Shapefile, MrSID Image, Bentley DGN and Autocad DXF, metadata in geospatial metadata schemas, such as ISO 19115, FGDC metadata standard or in a national of regional profiles, and links to service metadata. Data and metadata can be also made accessible through OGC services.

Service and data discovery

Description: A service of discovery of data and services is a key element in a geospatial system. The discovery leads to the reuse of the geospatial resources. However, as more geospatial information is made available online, the ability to describe, organize, and access it has become increasingly difficult. Increasing the discovery points, and making the discovery data reusable in new ways, should increase the opportunities for the discovery of the resources.

Benefits: Linked Data offers to the consumers of geospatial data and services to discovery the resources in by a navigational search. The use of XHTML documents with RDFa annotations allows delivering at the same time human and machine-readable information (e.g. Lopez-Pellicer et al. (2010b)). SPARQL and RESTful wrappers can replace existing services. The intended audience of the publication are not only current users but also casual users that use search engines to discovery information.

Alternatives: Use the geospatial systems that support the discovery of information known as *catalogue services* (OGC, see Nebert et al., 2007), *spatial data directory* (Australian Spatial Data Infrastructure, see ANZLIC, 2003), or *clearinghouse* (USA FGDC, see FGDC, 2005).

5.4 The Linked OWS Engine

The Linked OWS Engine is a Linked Data server that exposes metadata of OGC Web Services as Linked Data and offers a Linked Data wrapper of OGC Web Services operations. Additionally, the engine plays the role of broker of OGC Web Services. It can return OGC service metadata documents where the endpoints are semantic endpoints, that is, self-describable endpoints, where all the operations have a RESTful binding brokered by the Linked OWS Engine. The brokerage has been implemented applying the principles of Linked Data.

5.4.1 Design

The Linked OWS Engine is a stand-alone server based on ideas found in the Linked Data wrappers Pubby (Cyganiak and Bizer, 2007) and OAI2LOD Server (Haslhofer and Schandl, 2008). Figure 5.6 illustrates the main modules of this engine. The *request handler* is the module responsible of managing the URL spaces in accordance with the Linked Data best practices (see Section 5.4.2). The *description manager* module creates a description of resources (see Section 5.4.2; also *OWS requests* of Section 5.4.5). The *collections manager* module is responsible of enabling a navigational search of the contents (see Section 5.4.3) and the access to remote collections as Linked Data (see Section 5.4.4). The *capability rewriter* module adds the functionality of semantic endpoints (see Section 5.4.5) and returns managed OGC service metadata descriptions where all the operations are RESTful operations brokered by Linked OWS Engine (see Section 5.4.6). Finally, the *request rewriter* module is the module responsible of perform POST requests to OGC Web services for RESTful operations when required (see Section 5.4.6).

5.4.2 Linked Data server

The OWS Linked Data server exposes named RDF resources stored in a triple store as Linked Data following the practices described in Section 5.2.3. From this point of view, the system here outlined does not differ from other Linked Data wrappers, such as Pubby and OAI2LOD (see Section 5.2.4). Each relevant resource is identified by an URI in the triple store as the first Linked Data principle says. These URI are not required to be HTTP URIs.

In the context of OGC Services, few concepts have a URI. For example, the standard enforces the use of the URN schema to identify the coordinate reference systems (e.g. `urn:ogc:def:crs:EPSG::4326` that identifies the coordinate reference system WGS 84[9]). Some authorities maintain a repository of these resources. Standardized (e.g. operations name) or arbitrary (e.g. contents name) strings identify the concepts within these repositories. Some of them have a title only for human purposes. Such strings are not URIs.

[9] `http://www.epsg-registry.org/indicio/query?request=GetRepositoryItem&id=urn:ogc:def:crs:EPSG::4326`

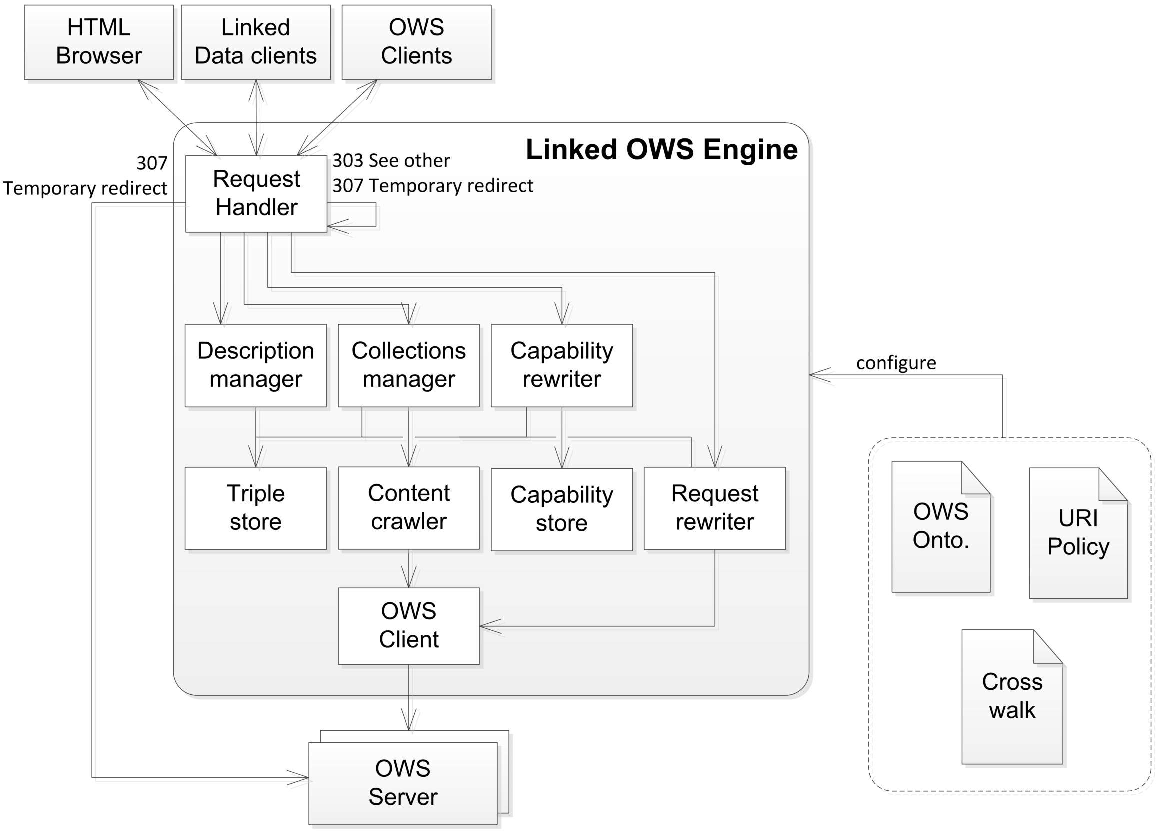


Figure 5.6: The Linked OWS Engine architecture.

The developer of a triple store is responsible of the URI naming policy within the triple store. The URI naming policy is defines how to mint URIs from the available data during the transformation of OGC Service metadata documents into RDF models (see a simple example in Table 5.10).

The second principle says that URIs that identifies resources should be resolvable HTTP URIs. However, the URIs stored in the triple store are required to be neither HTTP URIs nor dereferenceable URIs. One of the main functions of the Linked OWS engine as Linked Data server is to map the URIs in the triple store to HTTP URIs managed by the Linked Data server. A usual solution is to map a common URI prefix found in the triple store to a HTTP URI prefix managed by the server. For example, if the server has as base `http://www.example.com/` and the triple store has a service capabilities identified by the URI `urn:sdi:capabilities:www.idee.es/wms-cantabria/wms-cantabria`, the identifier may become `http://www.example.com/resoure/cap/www.idee.es/wms-cantabria/wms-cantabria`. This mapping should be opaque with respect to the URI naming schema used in the triple store.

The third Linked Data principle is related with delivering useful information and solving the ambiguity between resource and information resource. That is, the server has at least two URI spaces: a

Table 5.10: A simple URI naming schema based in OGC names and locators.

Name or locator	**Identifier in the triple store**
Document URN Form: `urn:ogc:doc:IS:WMS:1.3.0`	Document URN Form: `urn:ogc:doc:IS:WMS:1.3.0`
XML Namespace: `http://www.opengis.net/wms`	XML Namespace: `urn:ogc:ns:WMS`

Schema Location (static, found in XML documents):
`http://schemas.opengis.net/wms/1.3.0/capabilities_1_3_0.xsd`
	Schema Location (static, found in XML documents):
	`urn:ogc:schema:WMS:1.3.0`

Schema Location (dynamic, service request) `http://www.idee.es/IDEE-WFS/ogcwebservice?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&NAMESPACE=xmlns%28ideewfs=http://www.idee.es/wfs%29&TypeName=ideewfs:BDLL200Municipio`	Schema Location (dynamic, service request) `urn:ogc:type:www.idee.es:IDEE-WFS:ogcwebservice?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&NAMESPACE=xmlns%28ideewfs=http://www.idee.es/wfs%29&TypeName=ideewfs:BDLL200Municipio`
HTTP binding endpoint: `http://www.opengis.uab.es/cgi-bin/ICCTiled/MiraMon.cgi`	HTTP binding endpoint: `urn:ogc:endpoint:opengis.uab.es:cgi-bin:ICCTiled:MiraMon.cgi`
Document locator: `http://www.idee.es/wms/WMS-Cantabria/WMS-Cantabria?REQUEST=GetCapabilities`	Document locator: `urn:ogc:doc:www.idee.es:wms:WMS-Cantabria:WMS-Cantabria?REQUEST=GetCapabilities`

space for URIs that identify resources referenced in the triple store (e.g. `http://www.example.com/resource/cap/www.idee.es/wms-cantabria/wms-cantabria`), and a space for URIs that identify documents that describe a resource referenced in the triple store (e.g. `http://www.example.com/data/cap/www.idee.es/wms-cantabria/wms-cantabria`). When a user agent dereferences an HTTP URI that identifies a resource, the server should return a *303 See Other* redirect to the URI that identifies the most adequate representation for the user agent. If the user agent asks for a machine processable representation, the server returns a RDF model about the mapped resource in a RDF serialization format. The document includes backlinks, related descriptions and metadata (see Table 5.4 in Section 5.2.3). If the user agent asks for a human representation, the server returns from a XHTML document with RDFa annotations to a HTML document. A template engine fills predefined templates with the values of a RDF model describing the resource in order to generate human-readable representations. The selection of the template, and the logic within the template, is guided by the content of the RDF model. Table 5.11 shows the basic URI spaces supported by the Linked Data server. A URI template, which syntactically follows the IETF draft specification for URI templates (Gregorio et al., 2010), defines each URI space.

Table 5.11: Basic URI spaces supported.

URI template	Semantics	SPARQL	Resp.
`/resource/{ns}/{URIstring}`	Same as a resource identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*.	–	303
`/data/{ns}/{URIstring}`	A document that should contain a description about the resource `/resource/{ns}/{URIstring}` in machine-readable format.	DESCRIBE	200, RDF
`/page/{ns}/{URIstring}`	A document that should contain a description about the resource `/resource/{ns}/{URIstring}` in human readable format.	DESCRIBE	200, (X)HTML

The last Linked Data principle recommends that the description of a resource should contain links to other related dereferenceable resources. The return of representations that only contain resolvable HTTP URIs fulfils this constraint.

5.4.3 Navigational Search

Linked Data principles do not prescribe an organization of the URI space with directories, or taxonomies, to help clients to browse and narrow down the information they seek. Each server is responsible of its URI space, and this organization must be opaque to the client. The OWS Liked Data server provides a basic three level organization of the resources with pagination support that enables navigational search of all the accessible content (see Table 5.12). All the possible links are described in the representation making the naming structure opaque to the client. The root level describes the *dataset* published as Linked Data. The term *dataset* here refers to a collection of data stored in a triple store, published by the OWS Linked Data server, available as RDF, and accessible though dereferenceable HTTP URIs. The middle level is composed by collections of types, i.e. RDF classes, available in the dataset. The lowest level is composed by collections of instances of types advertised in the middle level.

The implementation of the navigational search requires the definition of a simple vocabulary for Navigation. This vocabulary is encoded in OWL in Appendix B. The vocabulary is identified with the URI `http://purl.org/iaaa/sw/nav` and its concepts has their names prefixed by the string `http://purl.org/iaaa/sw/nav#`. Figure 5.7 provides a graphical description of the concepts of the vocabulary for Navigation. The description of the navigational search uses terms from the *Vocabulary of Interlinked Datasets*[10] (voID, Alexander et al., 2009). The voID is a vocabulary and a set of instructions that enables the discovery and usage of linked datasets. This vocabulary helps the human user to get a quick impression of the kind of data available (e.g. name, short description),

[10] `http://rdfs.org/ns/void`

Table 5.12: Navigational URI spaces supported; page URI space not included.

URI template	**Semantics**	**SPARQL**	**Resp.**
`/resource/dataset`	The dataset published as linked data.	–	303
`/resource/structure {?pageParams*}`	The collection of types found in the dataset. Pagination support (variable *pageParams* may contain a map with limit and offset values).	–	303
`/resource/structure/{ns} /{URIstring} {?pageParams*}`	The collection of instances of the type identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*. Pagination support (variable *pageParams* may contain a map with limit and offset values).	–	303
`/data/dataset`	The description of the dataset published using the voID vocabulary. Links to collection of types with the property `nav:index`.	DESCRIBE	200, RDF
`/data/structure {?pageParams*}`	A document with RDF links to collections of instances. Links to next sibling page with the property `nav:nextPage`. Links to collections of instances with the property `nav :index`.	SELECT	200, RDF
`/data/structure/{ns} /{URIstring}{?pageParams *}`	A document with RDF links to instances of the type identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*. Links to next sibling page with the property `nav: nextPage`.	SELECT	200, RDF

and provide links to example resources in the dataset. The voID vocabulary also tells humans and machines technical features of the datasets, such as the vocabularies used in the dataset and the URI pattern of concepts within the vocabulary. The prefixes *nav:* and *void:* identify concepts from the Navigation and the voID vocabularies respectively.

The root level has its URI space identified by the URI `/resource/dataset`. Note that the implementation should allow the change of the static and dynamic template URIs by configuration. The root level URI is dereferenced to a description of the dataset using voID, that is, a description contains a statement that asserts that the resource `/resource/dataset` is a `void:Dataset`.

The middle level has its URI space identified by the URI template `/resource/structure{?pageParams*}`. The value of the variable *pageParams* is expected to be a map with the keys *offset* and *limit* that should be expanded into the string `?limit=key_value&offset=offset_value`. The

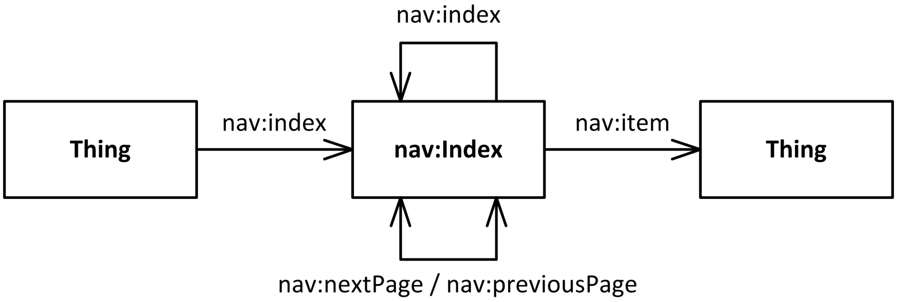


Figure 5.7: Basic vocabulary for Navigation

voID description of the dataset should include a `nav:index` property that relates the URI `/resource/structure`, that is, the dataset, with a resource identified with the URI `/resource/structure`. The property `nav:index` relates a resource with other resource that contains additional information in form of a paginated index list. When the latter is dereferenced, the OWS Linked Data server returns a RDF model that contains a brief description of the different types, that is, RDF classes, found in the dataset. The simplest description is an assertion of being a RDF class (e.g. `http://www.example.com/resource/sdi/Operation a rdfs:Class .`). All the resources pointed by a `nav:index` property are instances of the class `nav:Index`.

If the number of available types is over a threshold, the index is paginated. That is, the representation of `/resource/structure` contains only a number of types below or equal to the threshold, and a statement that relates the URI `/resource/structure` with the property `nav:nextPage` to the resource identified by `/resource/structure?limit={p}&offset={v+1}` where the variable v represents threshold and the variable p represents the page size. The URI naming convention is opaque to the user because it only navigates the `nav:nextPage` links. The property chain `nav:index nav:nextPage` implies that the sibling page is also a part of the index. This is the simplest pagination structure supported by the OWS Linked Data and it is applied when a collection of resources requires to be paginated.

The lower level has its URI space identified by the URI template `/resource/structure/{ns}/{URIstring}{?pageParams*}`. This URI template identifies a collection of instances of the class identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*. When this URI is dereferenced, the OWS Linked Data server returns a RDF model that contains at RDF links to the different instances (e.g. the assertion of being an instance of the class). The RDF links can relate this index with the property `nav:item` to the indexed instances. This lower level is advertised in the description of the class identified by the URI `/resource/{ns}/{URIstring}` in the middle level with a `nav:index` property that relates with `/resource/structure/{ns}/{URIstring}`. The pagination support is identical to the middle level.

5.4.4 Exposing contents of OGC Web services

Chapter 3 identifies different types of interactions available in OGC Web services. Many of the variants of the interactions, *GetDataSubset* and *GetResourceById* give access to a subset of named resources o a named resource in a remote service instance. A Linked Data server may wrap the operations related to these interactions to expose these resources as Linked Data.

This is one of the main functionalities of the OWS Linked Data server: to expose resources and collections of resources accessible through an OGC Web service as Linked Data. The approach is quite similar to the navigational search described above and uses the same conventions (see Table 5.13). All the possible links are described in the representation making the naming structure opaque to the client. The root level composed by resource identified as the parent of a set of the remote items in a service. For example, each of the *feature types* supported by a WFS server is a root candidate. The *record schema* supported by a CSW server is also a root candidate. The middle level is composed by collections of resources of the same kind (e.g. features, metadata records). The lowest level is composed by each of the instances advertised in the middle level.

The resources that can be part of the root level are instances of information types, such as *feature type*, *coverage*, *observation* and *record.* These resources are the information types of the instances returned in *GetResourceById* interactions. Additionally, *Feature types*, *records* and *observations* can give access to collections of resources in *GetDataSubset* interactions. The maintenance processes of the OCG Linked Data server analyse the available information types, and create and maintain `nav:index` links to collections of resources and `nav:item` to instances that do not belong to collections.

The middle or collection level has its URI space identified by the URI template `/resource/collection/{kind}/{ns}/{URIstring}{?pageParams*}`. The variables *ns* and *URIstring* identify the resource in the root level and in the triple store using the semantics described in the previous sections. The collection is identified by the concatenation of the URI prefix mapped to the variable *ns*, the variable *URIstring,* the literal *collection*, the term *kind* and the exploded value of the variable *pageParams* (if any). The variable *kind* plays the role of discriminator. By default, *kind* has the same labels as its information type class (e.g. *record* for record, *featureType* for feature type). The root resource is related with the collection level resources with the property `nav:index.` The pagination is enabled with the variable *pageParams* with the same semantics as above.

When a user agent dereferences the URI that identifies a collection of resources the OWS Linked Data server proceeds as follows (see Figure 5.8):

1. The server looks for resources in the triple store related with the collection resource with the property `nav:item`. The response is limited to an amount of items in the collection. If the limit is surpassed or the server suspects of the existence of a next page, the response should contain a `nav:nextPage` link in the same way as was described for the navigational search.

2. If the collection does not have any description, the server uses the variables *kind*, *ns* and

Table 5.13: Content URI spaces supported; page URI space not included.

URI template	**Semantics**	**SPARQL**	**Resp.**

`/resource/collection/`
`{kind}/{ns}/{URIstring}`
`{?pageParams*}`
	A collection of resources contained in an OGC Web service whose type is identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*. The variable *kind* acts as discriminator. Pagination and other parameters supported (variable *pageParams*).	–	303

`/resource/item/{kind}/`
`{ns}/{URIstring}{?`
`itemParams*}`
	A resource contained in an OGC Web service that is identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*. The variable *kind* acts as discriminator. Additional parameters may be supported (variable *itemParams*).	–	303

`/data/collection/{kind}`
`/{ns}/{URIstring}{?`
`pageParams*}`
	A cached collection of RDF links to content resources. Pagination and other parameter supported. If the cache is empty, the type is accessible through a *Get<X>Subset* interaction, there is a defined mapping from the remote schema to a RDF model the system makes a requests to the OGC service using the parameters provided, retrieve the result, applies the matching, and cache and return the RDF links to the items.	SELECT	200, RDF

`/data/item/{kind}/{ns}`
`/{URIstring}{?itemParams`
`*}`
	A cached description of the content of a resource. If the cache is empty, the content resource is accessible through a *Get<X>ById* interaction, there is a defined mapping from the remote schema to a RDF model the system makes a requests to the OGC service using the parameters provided, retrieve the result, applies the matching, and cache and return the description of the content resource.	DESCRIBE	200, RDF

URIstring to identify the type of resource, the interaction and the operation. The server crafts a request to the remote OGC Web service. If the variable *pageParams* is present, its value is used in the request if it is supported by the standard.

3. The server makes the OGC request and wait for the response. As the response may take too long, the server may return a 202 Accept response to the user agent.

4. The server receives the response and checks if there is available a transformation from the media type of the response to a RDF model. The server uses a registry with transformations from well known OGC and ISO schemas to RDF models for finding the best transformation. This transformation should produce items of the collection. Each new item is locally identified using the URI that identifies the collection concatenated with the parameters required for its identification in a *GetResourceById* request. Each new resource is associated with its collection with the property `nav:item`. Additional Dublin Core metadata terms, such as *source* or *date,* can be automatically added. The operations related with *GetDataSubset* are often paginated. If the server deduces from the response that there is a next page, it adds a `nav:nextPage` link to the next remote page. All the new information is stored in the triple store.

5. The server returns a response following the rules of the first step.

The lower or item level has its URI space identified by the URI template `/resource/item/{kind}/{ns}/{URIstring}{?itemParams*}`. The variables *ns* and *URIstring* identify the resource in the root level as above. The item is identified by the concatenation of the URI prefix mapped to the variable *ns*, the variable *URIstring,* the literal *collection*, the term *kind* and the exploded value of the variable *itemParams* (if any). When a user agent dereferences the URI that identifies an item the OWS Linked Data server proceeds as follows (see Figure 5.9):

1. The server looks for a description of the mapped URI in the triple store. If the description only contains a relation with the property `nav:item` to a root element or to a collection, the server proceeds to the next step. Otherwise, it returns the description found.

2. The server uses the variables *kind*, *ns* and *URIstring* to identify the type of resource, the interaction and the operation. The server craft a request to the remote OGC Web service using the values of the variable *itemParams* to identify the target resource.

3. The server makes the OGC request and wait for the response. As the response may take too long, the server may return a 202 Accept response to the user agent.

4. The server receives the response and checks if there is available a transformation from the media type of the response to a RDF model. The server uses a registry with transformations from well known OGC and ISO schemas to RDF models for finding the best transformation.

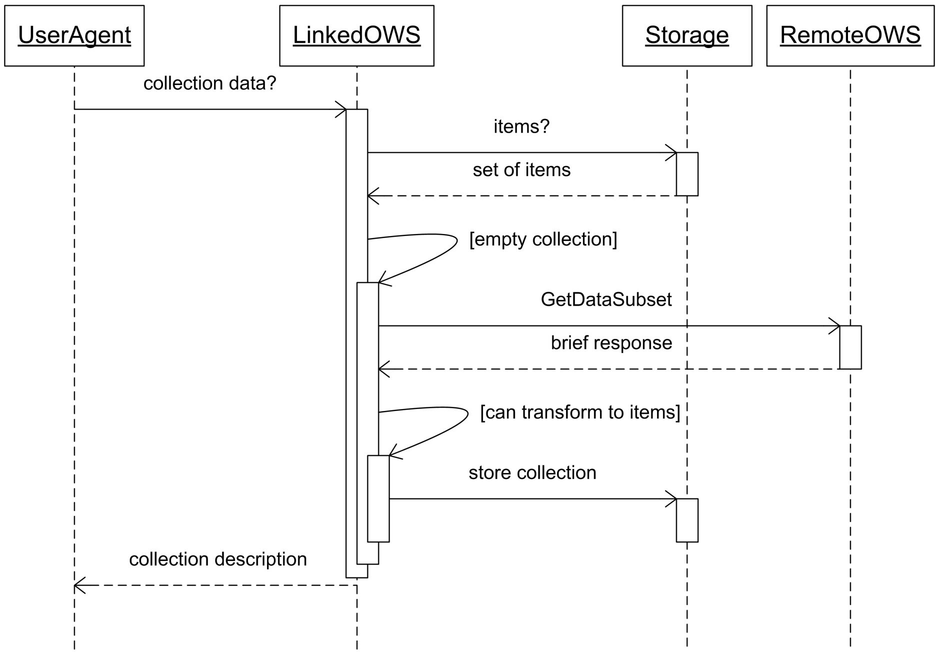


Figure 5.8: Sequence of interactions for obtaining the description of a collection.

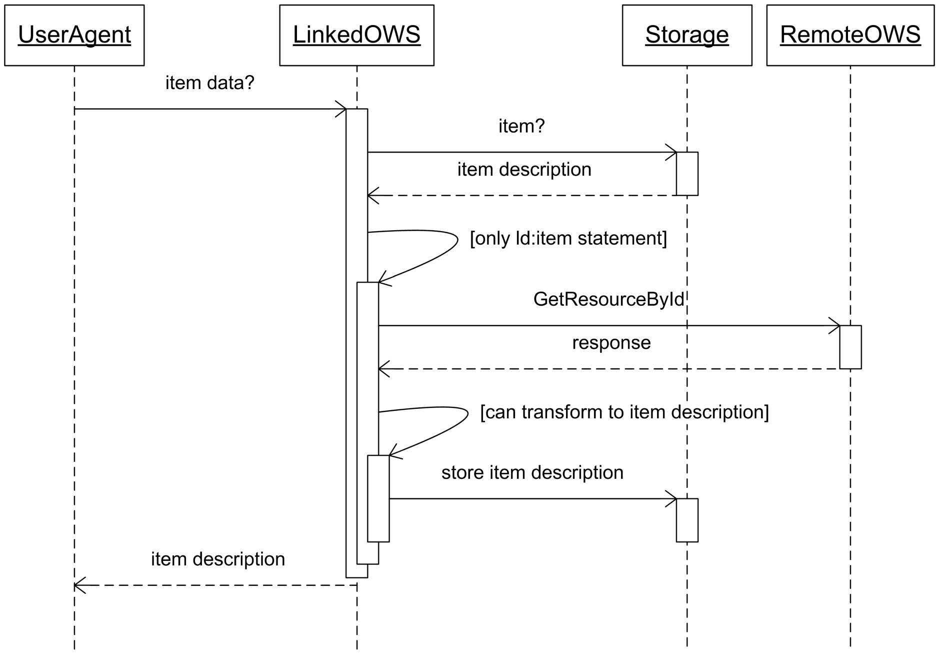


Figure 5.9: Sequence of interactions for obtaining the description of a collection item.

The information obtained is stored in the triple store as properties of the item. Additional Dublin Core metadata such as source, creation date is automatically added. All the new information is stored in the triple store.

5. The server returns the updated description of the mapped URI.

This thesis proposes for the transformations the use of well known Dublin Core vocabulary and the Geo-Net ontology described in Chapter 4. There are available several mappings from well known OGC and ISO schemas to Dublin Core. For example, (WS/MMI-DC, 2003) describes the crosswalk of the geographic metadata ISO 19115 to the Dublin Core vocabulary defined in CWA 14857. In addition, the Dublin Core abstract model (DCAM) has a reference model formalized in terms of the semantics of the RDF abstract model since 2005 (Powell et al., 2007). Geo-Net provides support for basic mapping of GML application schemas. Both are based in the General Feature Model and Geo-Net provides support for capturing a minimum set of relevant properties: the feature, its feature type, feature relationships, place names, spatial location and spatial reference system.

5.4.5 Semantic endpoints for OGC services

A semantic endpoint is a self-describable endpoint. The idea is as follows. Given an OGC Web service, if the Linked OWS engine serves its description as Linked Data, each operation of the OGC Web has a URI managed by the engine. . Hence, it is possible to enable clients to use this URI to ask for the capabilities of the operation and its semantics. If the client knows the OGC standard that controls the behaviour of the operation, the client can craft a valid OGC request targeting that URI. If the client send that request to the OGC Linked Data server and ask for a machine-readable format, the server can return, via redirection, a semantic description of the request. If the client request a format not managed by the OGC Linked Data server, the server redirects the request to the endpoint in the remote OGC server. The semantics of HTTP redirection and the httpRange-14 requires an additional caveat. When the client request a format not managed by the server with a HTTP GET request, the URI in the *resource* URI space redirects via *303 See Other* to the *server* URI space that in turn redirects via *307 Temporary Redirect* to the remote OGC server. If a GET requests redirects via *307 Temporary Redirect*, an identity clash problem arises: a semantic application may thing that the non-information resource identified by the URI in the resource URI space and the information resource identified by the URI request to the remote service are the same. The requirements of this solution are described in Table 5.14.

An OGC client requires an additional element in order to use the semantic endpoints: an OGC service metadata document using these endpoints. This require that copies of the original capabilities were stored in the triple store or available a documental repository. The procedure is as follows (see Figure 5.10):

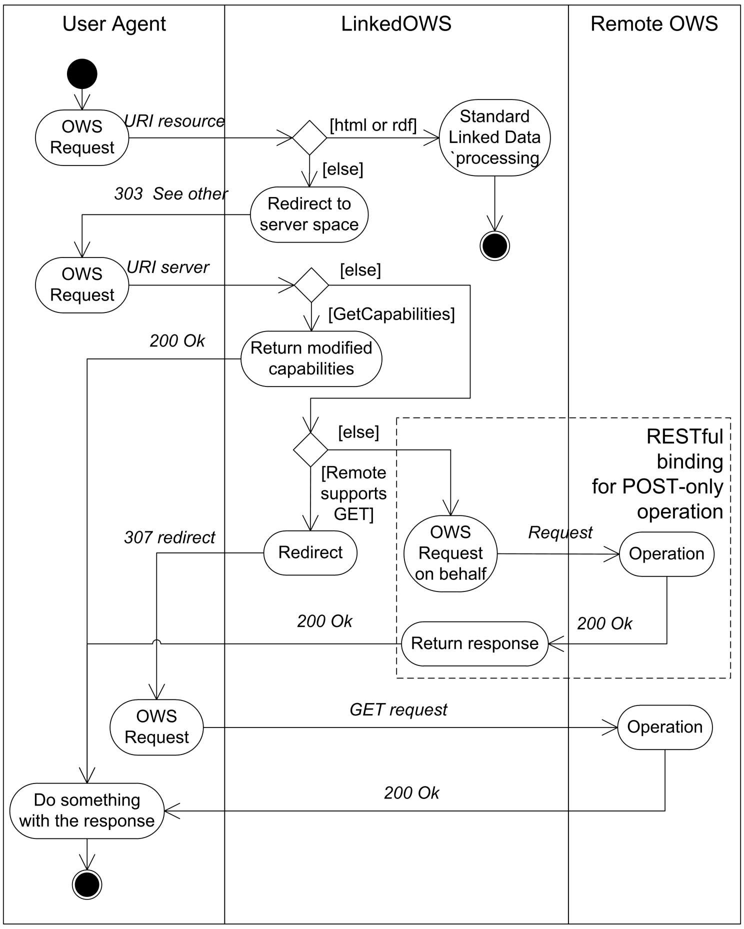


Figure 5.10: Semantic proxy and RESTful extension.

- The Linked Data server detects that the user is requesting neither a semantic description nor a human readable description for an operation. Then, the server redirects to the *server* URI space with a *303 See Other* response.

- The client requests the new URI.

- The Linked Data server manages the new URI as a possible redirect to a remote service. First, the server extracts the mapped URI, and then, obtains the information about the operation.

- If the operation is *GetCapabilities* and the service metadata document is available, the Linked Data server returns a copy of the service metadata document where the managed URIs of the corresponding operations replace the original endpoints of each service operation. Otherwise, the server makes a *307 Temporary Redirect* to the corresponding OGC server.

- The OGC client starts to use the semantic endpoints transparently.

The OGC client is now bound through the Linked Data server to the remote OGC Web service. If the OGC client has a library that can query or navigate the Web of Data for further information, the client can ask this library with the URIs used in its requests for additional information about them.

5.4.6 RESTful binding for OGC services

OGC specification allows using GET, POST or both for operation requests. In Chapter 3 we have identified that the interactions *GetCapabilities*, *GetDataSubset* and *GetResourceById* are read operations. However, it is feasible that an OGC service use only POST for some of these operations. The approach described above for semantic proxies can be used for rewiring the request to these interactions to the GET request even if a GET request is not supported by the remote service. This require two modifications to the behaviour described for semantic proxies (see Figure 5.10):

- The copy return of the service metadata only contains GET endpoints. All the POST endpoints are deleted if a GET endpoint is available. If there is no GET endpoint available, the POST endpoint is substituted by a GET endpoint.

- The server URI space only makes a *307 Temporary Redirect* if the target service supports a GET request. Otherwise, encodes a POST request on behalf of the user agent to the remote OGC service. The response is returned to the client.

Table 5.15 summarizes the changes in the behaviour of the *server* URI space.

5.5 Applications

The OWS Linked Data system is the engine behind two applications of Linked Data in the geospatial domain: the CSW2LD toolkit (Lopez-Pellicer et al., 2010b) and the demonstration of the Geo Linked Data concept (Lopez-Pellicer et al., 2010d).

5.5.1 The CSW2LD toolkit

The CSW2LD toolkit was the precursor of the OWS Linked Data engine and now embeds an OWS Linked Data engine. The CSW2LD toolkit is a system that offers a solution to some drawbacks detected in the OGC Catalogue Service for the Web (CSW) protocol. This implementation uses approach for crosswalking metadata record into RDF model described in the previous chapter.

The context. The CSW is the HTTP protocol binding to the OGC Catalogue Services (CAT) specification (Nebert et al., 2007). This specification defines a set of abstract interfaces for the discovery, access, maintenance and organization of metadata repositories of geospatial information

Table 5.14: Extension of URI spaces for semantic proxies; page URI space not included.

URI template	**Semantics**	**SPARQL**	**Resp.**
`/resource/{ns}/{URIstring} {?params*}`	**Added behaviour:** A GET request neither human nor machine-readable redirects with *303 See Other* to the server URI space. Other requests neither human nor machine-readable redirects with *307 Temporary Redirect* to the server URI space.	–	303, 307
`/data/{ns}/{URIstring} {?params*}`	**Added behaviour:** Describe a resource identified by the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring*. The use of *params* only has sense if the resource is an operation.	DESCRIBE	RDF
`/server/{ns}/{URIstring} {?params*}`	Redirects to a remote resource. It uses as base the information of the resource identified the concatenation of the URI prefix mapped to the variable *ns* and the variable *URIstring* and concatenates the contents of *params*. If the request is a *GetCapabilities* operation and the service metadata document is available, the server returns instead a copy of the service metadata document where URIs in the resource URI space that identifies corresponding operation replace the original endpoints of each operation.	DESCRIBE	XML, 307

Table 5.15: Extension of URI spaces for RESTful behaviour.

URI template	**Semantics**	**SPARQL**	**Resp.**
`/server/{ns}/{URIstring} {?params*}`	**Added behaviour**: POST endpoints in the copy of the service metadata document are replaced by GET endpoints; requests to these GET endpoints are posted to the server on behalf of the user agent and the result returned to it.	DESCRIBE	Any, 307

and related resources in distributed computing scenarios, such as the Web. The management interface supports the ability to administer and organize collections of metadata in the local storage device. The discovery interface allows users to search within a catalogue and provides a minimum query language. Finally, the access interface facilitates access to metadata items previously found with the discovery interface. The CAT specification also defines an abstract information model that includes a core set of shared attributes, a common record format that defines metadata elements and sets, and a minimal query language called CQL. Additionally to the HTTP protocol binding, the CAT specification includes binding implementation guidance for the application protocols Z39.50, a pre-Web protocol widely used in digital libraries, and CORBA/IIOP, a remote procedure call specification in a niche of relative obscurity (see Henning, 2008).

SDIs use CSW as one of the gateways to their geospatial resources. An example of the relevance of CSW is the recommendation issued in the context of INSPIRE by the INSPIRE Network Services Drafting Team (2009) to SDIs in European Union to derive the base functionality of discovery services from the ISO profile of CSW defined in Voges and Senkler (2007).

CSW is quite complex. For example, the operation *GetRecordById* fetches representations of metadata records using the identifier of the metadata in the local metadata repository. The parameter *elementSetName*, if used, establishes the amount of detail of the representation of the source record. Each level of detail specifies a predefined set of record elements that should be present in the representation. The predefined set name full represents all the metadata record elements. By default, the operation *GetRecordById* returns a metadata record representation that validates against the information model of the metadata repository. The parameter *outputSchema* allows user agents to request for a response in a different information model, and the CSW implementations must support at least the representation of the common information schema defined in the CSW standard. Figure 5.11 shows a sample *GetRecordById* request for a metadata record available in IDEE, the SDI of Spain, and the corresponding response. The request URI identifies the location of the CSW server, the operation, the identification of the metadata record (parameter *id*), the amount of detail of the representation (parameter *elementSetName*), and the output schema (parameter *outputSchema*). The XML response consists of a `<GetRecoredByIdResponse>` element that contains a record that conveys the information of the source metadata. When a `<SummaryRecord>` element is the conveyor, the retrieved representation contains a subset of the source metadata record elements. The value of the output schema identifies the subset that conforms to the common information schema defined in the CSW standard.

The CSW standard represents an approach to the discovery of information in the Web. The current trends in the Web show alternative approaches to the discovery of information:

- *Search engines give access to Deep Web databases.* This issue has been analysed in Chapter 2.

- *Navigational Search using Linked Data.*

Request:

```
GET/csw/servlet/cswservlet?request=GetRecordById&id=
ESIGNMAPASRELIEVESERIE200701180000&elementSetName
full&outputSchema=http://www.opengis.net/cat/csw/
2.0.2 HTTP/1.1
Host: www.idee.es
```

Response:

```
HTTP/1.x 200 OK
Content-Type: application/xml;charset=ISO-8859-1
...

<?xml version = '1.0' encoding = 'ISO-8859-1'?>
<GetRecordByIdResponse
 xmlns="http://www.opengis.net/cat/csw/2.0.2"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:dcterms="http://purl.org/dc/terms/"
 ...>
 <SummaryRecord>
  <dc:title>Mapas en Relieve</dc:title>
  <dc:identifier>
   ESIGNMAPASRELIEVESERIE200701180000
  </dc:identifier>
  ...
  <dc:format>PVC</dc:format>
  <dc:subject>elevation</dc:subject>
  <dc:subject>imageryBaseMapsEarthCover</dc:subject>
  <dc:type>dataset</dc:type>
  ...
  <dcterms:spatial>COUNTRIES.SPAIN</dcterms:spatial>
  ...
  <dcterms:spatial>
   northlimit=43.8;
   southlimit=37.83;
   westlimit=-9.32;
   eastlimit=0.72;
  </dcterms:spatial>
 </SummaryRecord>
</GetRecordByIdResponse>
```

Figure 5.11: CSW GetRecordById request and response.

CSW is an OGC Web service standard undoubtedly useful to enable the discovery and access to geographic information resources within the geographic community. However, there are three drawbacks in CSW in relation with the alternative approaches:

- Deep Web crawlers are focused to quite simple interfaces; the CSW is a complex interface.
- Remote Procedure Call style over HTTP is orthogonal to REST approaches, such as Linked Data.
- The CSW does not endorse a uniform practice to relate records; this issue makes difficult to relate metadata records from different sources.

We can identify three drawbacks in the design of CSW that harms its use in the above scenario:

- **Mismatch with operational model of Deep Web crawlers**. CSW is hard to crawl by Deep Web crawlers that follows the standard operational model based on the analysis of forms. This approach is suitable only forms that generate CSW HTTP GET requests without the use of the `FILTER` parameter. CSW HTTP GET request encode constrains in a single named parameter (`FILTER`) as a CQL string (see Nebert et al., 2007), or as a XML Filter (Vretanos, 2005a). This characteristic is incompatible with the query model of the deep Web crawlers.

- **Queries limited to same record properties**. The field based query model of the CAT specification does not define the support for associations in the CQL or Filter syntax. CSW application profiles may describe an ad-hoc support of associations. For example, the ISO application profile (Voges and Senkler, 2007) supports the linkage between services and data instances. Nevertheless, the linkage is based in the equality of literal values of properties, such as `MD_Identifier.code`, and the profile does not extend the CQL and the XML Filter syntax. Hence, association queries require being decomposed in parts. For example, in a metadata repository where metadata records about data and services instances are linked, a query that returns the services that serves data created by a producer requires (1) to query initially about the data created by this producer, (2) to retrieve their identifiers, and then, (3) to query about servers that serve data with these identifiers.

- **RPC approach to access metadata**. Metadata repositories are behind a proprietary RPC from the point of view of other communities. CSW does not define a simple Web API to query and retrieve metadata. Some communities that potentially can use CSW are accustomed to simple APIs and common formats. For example, many geo mashups and related data services (see Turner, 2006) use Web APIs to access and share data follows the REST architectural style instead.

Conceptual solution. The conceptual solution to solve the drawbacks of CSW and accessing the SDI metadata can be decomposed as follows:

- **CSW interface model**. A metadata repository contains metadata about resources. Client applications use CSW requests to query metadata repositories. The CSW requests may generate metadata snapshots that are subsets of metadata at the time of the request. The CSW request determines the amount of information (user defined, brief, summary or full records) and the information schema of the metadata snapshot. The CSW response contains the realization of the metadata snapshot in a supported media format. XML is the only media format that all CSW implementations must support.

- **Harvest model**. The harvest produces a set of metadata snapshots realized in XML representations. The harvest process asks for metadata records whose information model can crosswalk to Dublin Core. The CAT specification defines a common group of metadata elements expressed using the Dublin Core vocabulary. CSW defines a default mapping of the common group of metadata elements to XML that all CSW implementations must support. The harvest process queries for the common representation if available crosswalks cannot be applied to the information model of the catalogue.

- **Semantic access model**. The harvested representation of the metadata snapshot is mapped to the RDF data model, and then, it is published applying the Linked Data principles. The base of the mapping is the DCMI recommendation for expressing Dublin Core using RDF. The result is a semantic description about a resource that is a version of the metadata snapshot that describes the same resource. This semantic description is published according to the best practices to publish Linked Data. The model assumes that a dereferenceable URI, the semantic URI, can identify the resource that the semantic description describes. This semantic URI is owned by the responsible of the semantic publication and redirects to an URI where user agents can get a RDF representation of the semantic description. The semantic description, in turn, has the semantic URI as subject in its assertions. If the mapping process discovers links between the resources, it may replace the original RDF mapping by these semantic URIs. For example, the description of a service may include a brief description of the data. Then, this brief description can be replaced with the URI that identifies the semantic description of the data. The semantic description may contain a link that encodes a CSW HTTP GET re-quest equals to the CSW request done in the harvest. Semantic browsers and search engines, such as Tabulator (Berners-Lee et al., 2006) and Sindice (Tummarello et al., 2007) respectively, can browse and index the semantic descriptions, and use the links to navigate to other resources or to retrieve transparently the original metadata description.

- **Non-semantic access model**. Given the semantic descriptions described in the previous point, the model assumes that an URI identifies the human-readable representation in HTLM format. The semantic URI of a resource may be resolved to this URI if the agent requests a human-readable representation of its semantic description. This representation uses the HTLM element `<link>` to provide information to navigate alternative representations. At least, it includes a link that points to the semantic URI and a link that encodes the CSW HTTP GET request. Web browsers and search engines can browse and index respectively these representations. In addition, they can use the links to navigate to the semantic representations and to retrieve transparently the original metadata description.

Implementation. The CSW2LD Toolkit (initially presented in Lopez-Pellicer et al., 2010b) implements the above conceptual model. The CSW2LD toolkit implements a workflow that can be decomposed in the following steps:

- Analyse the capabilities of the CSW service to discover the information models served and the levels of amount of information.

- Fetch identifiers of new and updated records with the CSW *GetRecords* operation.

- Retrieve new and updated records using the *GetRecordById* operation; request ISO 19115 / ISO 19119 information models if they are available.

Table 5.16: Excerpt of the mapping relations defined in SKOS.

Property	Characteristics	Description
skos:exactMatch	subproperty of *skos:closeMatch* disjoint with *skos:relatedMatch* transitive symmetric	This property links two concepts indicating a high degree of confidence that the concepts can be used interchangeably *across a wide range* of information retrieval applications.
skos:closeMatch	symmetric	This property links two concepts that are sufficiently similar that they can be used interchangeably *in some* information retrieval applications.
skos:relatedMatch	disjoint with *skos:exactMatch*	This property links two concepts with some degree of alignment or association.

- Crosswalk to the Dublin Core vocabulary if the requested information model is not the common information model.
- Map the set of Dublin Core metadata terms to the RDF data model.
- Generate or update the human readable and machine-readable representations from the RDF graphs.

The `GetRecords` operation does a search and returns piggybacked metadata. The harvest process uses the `GetRecords` operation to determine the number of metadata records to retrieve, and to obtain piggybacked unique identifiers for retrieve metadata records. Optionally, along with the identifier, the harvester process can ask for the creation or update date of the record within the catalogue.

The CSW2LD Server is a stand-alone application implemented in Java. In the simplest configuration, the server embeds the Linked OWS engine for publishing metadata records, and the scriptable focused crawler described in chapter 2. A job scheduling service fires periodically a crawling job that performs a deep web crawl of the contents of the service, i.e. metadata records. The crawler is configured to transform them into a RDF model using the configured crosswalk to the Dublin Core vocabulary and the ontology for OGC Web services.

The CSW2LD server can be configured to maintain a collection of mappings between metadata records from different CSW services. This feature is related with the fourth Linked Data rule: include as many interesting links as possible. The mappings discovered are captured by adding the mapping properties defined in the SKOS vocabulary (see the SKOS specification (Miles and Bechhofer, 2009) and table 5.16): `skos:exactMatch`, `skos:closeMatch`, and `skos:relatedMatch`. The mapping is done by comparing the property values of two metadata records retrieved using some of the properties

Table 5.17: Properties of the core catalogue schema (Nebert et al., 2007).

Name	**Card.**	**Data type**	**Description**
Identifier	1..*	Identifier	An unambiguous reference to the resource within a given context. Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system (e.g. URI, DOI, ISBN).
Title	1..*	String	A name given to the resource. Typically, Title will be a name by which the resource is formally known.
Type	0..1	String; code list	The nature or genre of the content of the resource. Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary. To describe the physical or digital manifestation of the resource, use the Format element
Subject	0..1	String; code list	A topic of the content of the resource. Typically, Subject will be expressed as keywords, key phrases, or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
Format	0..*	String; code list	The physical or digital manifestation of the resource. Typically, Format will include the media-type or dimensions of the resource. Format may be used to identify the software, hardware, or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary.
Relation	0..*	String; identifier	A reference to a related resource. Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system
Modified	0..*	Date	The date of a creation or update event of the catalogue record.
Abstract	0..*	String	A summary of the content of the resource
BoundingBox	0..*	Spatial	A bounding box for identifying a geographic area of interest expressed in the WGS 84 coordinate reference system

of the *Core catalogue schema* defined in CAT (see table 5.17) using a combination of measures: string comparison (*identifier, type, subject, format, relation*), string similarities (*title, type, subject, format, relation, abstract)*, date similarities (*date*), and spatial distance measures (*bounding box*). If the identifiers are the same and the normalized combined score is above a configurable level L_{exact}, CSW2LD links both resources with the property `skos:exactMatch` as there is a high level of confidence that both records identifies the same thing and the records can be used indistinctly. If the identifiers are different and the normalized combined score is above a level L_{close} ($L_{close} < L_{exact}$), CSW2LD links both resources with the property `skos:closeMatch`. If the measures about properties, such as the *subject* or the *spatial distance,* are above a third score L_{related}, and the resources are nor exact neither close matches, CSW2LD links both resource with the property `skos :related`, that is, both records are spatially or topically related.

The CSW2LD server supports the standard XML serialization format (RDF/XML) and non-XML serialization formats (N3, Turtle, N-triples) of RDF. As a means to produce human readable content, the CSW2LD server can negotiate human readable content, which is served by means of a template engine that allows the creation of custom made description of resources in XHTML with RDFa annotations. Finally, the CSW2LD server supports the negotiation of non-semantics XML documents (e.g. CSW Capabilities documents, ISO Geographic Metadata documents). This functionality is implemented using the brokerage support provided by the Linked OWS engine.

Lessons learned. The CSW2LD toolkit allows to access as Linked Data metadata stored in repositories accessible by the CSW specification. Applied to SDI metadata catalogues, the CSW2LD toolkit exposes the description of SDI assets as dereferenceable Web resources, and allows search engines to index them. On the other hand, the published RDF description of metadata records and resources is not standard, and can be semantically inaccurate. The main reasons lie on the lack of standards mappings from geographic metadata schemas to the RDF model, and the heterogeneity of communities targeted by CSW. Future versions of the CSW2LD toolkit could include additional technical features, such as additional crosswalks, and functional features, such as the generation of links between the metadata and existing thesauri and ontologies, augment the meta-metadata available about the provenance and quality of the exposed information, and describing the exposed data as aggregations.

5.5.2 Geo Linked Data

The Geo Linked Data (introduced in Lopez-Pellicer et al., 2010d) is a proposal for the inclusion in the content negotiation of Linked Data geospatial representations. The development OWS Linked Data engine was the origin of this proposal and the tool used for the proof of the concept.

The proxy for concept. In the Linked Data approach, when a URI acts as identifier for an entity, which may exist outside the Web, the URI can be dereferenced to a Web resource that provides useful information about the entity (Sauermann and Cyganiak, 2008). The user agent works under the next assumption: when the URI gives access to a Web resource with a different URI at a given time, the Web resource could be interpreted as a *proxy for* an entity, at least in that given time. The concept *proxy for* is part of the Identity of Resources and Entities on the Web (IRE) model proposed by Presutti and Gangemi (2008), a framework for reasoning when a Web URI can be associated to an entity. The *proxy for* association between a Web resource (e.g. a semantic description) and an entity (e.g. Lisbon) means that the representation of the Web resource (e.g. a RDF document) materializes information (e.g. a pair of geographic coordinates) about the entity.

The IRE model classifies the *proxy for* relations as *exact* or *approximate*, and as *formal* or *informal*. An *exact proxy for* relation means that the Web resource only describes one entity, and otherwise is *approximate*. Nevertheless, an *exact proxy* may contain references to related entities. For example, satellite images about Portugal may contain parts of the Atlantic Ocean and Spain. However, satellite images are meant to be *exact proxies for* Portugal. A *formal proxy for* relation means that the representation of the Web resource is in formal language. If not, it is an *informal proxy for* relation.

Geospatial proxies. The *proxy for* concept is independent of the technology and the information about the entity. Hence, the definition of *proxy for* is applicable when the Web resource is a Geospatial Web resource conveying spatial characteristics of other entities. We designate as a *geospatial proxy* any Web resource conveying spatial information about other entities using Geospatial Web standards (e.g. a GML document describing the location of Lisbon, a satellite image in JPEG describing the Earth). A *geospatial proxy* is an *exact geospatial proxy* if it only describes one entity.

Linked Data principles require that the information must be available in RDF and, for humans, should be available in HTML. We can characterize RDF and HTML representations as having *exact formal proxy for* and *exact informal proxy for* relations with an entity, respectively. For humans, an *exact geospatial proxy* and a HTML representation describing the same entity could contain equivalent spatial information. From a machine-processable point of view, an *exact geospatial proxy* could be considered as an alternative representation of the spatial information of an RDF representation when it is possible to map the content of the geospatial proxy to a formal model. Standardized data models (see Kresse and Fadaie, 2004) specify the information contained in Geospatial Web representations facilitate the mapping. It is possible to find in the literature mappings to formal models, such as Lieberman (2006). We can conclude that *exact geospatial proxies* conform to Linked Data rules. That is, when a semantic application requires spatial information, they could be an alternative for HTML and RDF representations.

We identify four roles that could be useful to understand how to use Geospatial Web descriptions

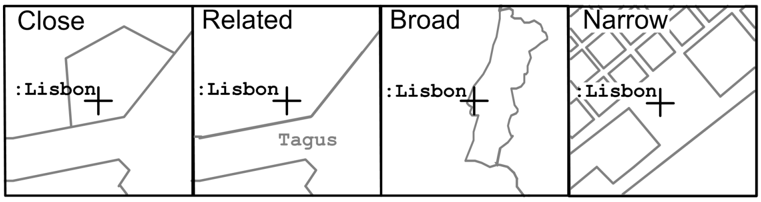


Figure 5.12: Different examples of maps that may act as *geospatial proxy for* the official boundary of Lisbon, capital of Portugal: *close* is a sketch of the boundary, *related* shows the area where the boundary of Lisbon is, *broad* shows the boundary of Portugal, and *narrow* shows the *Praça do Comércio* (Commerce Square), a landmark of Lisbon.

in Semantic Web applications (Figure 5.12):

close A *close proxy* is a proxy that some information systems can use as source for an alternative identifier of the entity. For example, a marker that identifies an entity in a map provided by a GML document is a *close proxy*.

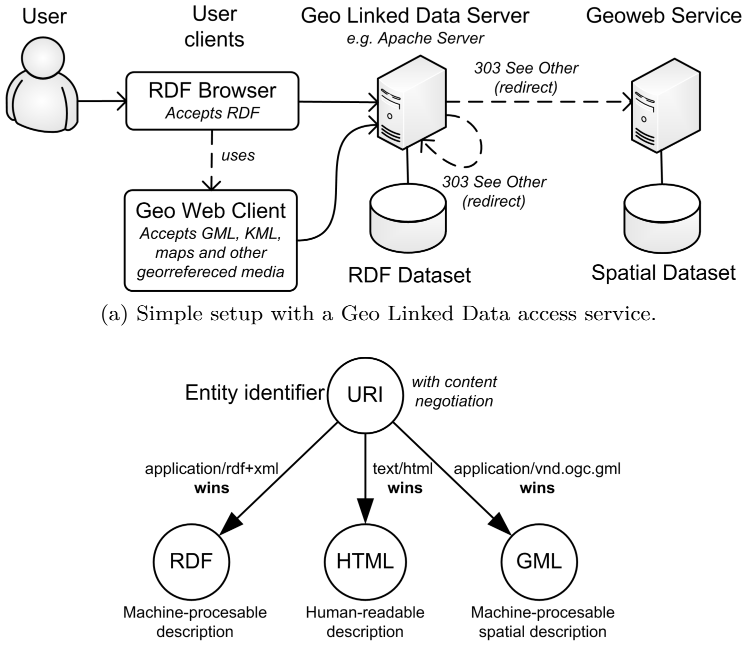
related A *related proxy* provides an indirect description of the resource through the spatial characteristics of the proxy. For example, a satellite image of an area is a *related proxy* of the entities of the area.

broad A *broad proxy* is a kind of related proxy that realizes essential characteristics of an entity that is a parent of the described entity. For example, a large regional map is a broad geospatial proxy for capital cities.

narrow A *narrow proxy* is similar to a broad proxy but realizes essential characteristics of an entity part of the described entity. An example is a map that shows a landmark of a city when the described entity is the city.

The Geo Linked Data recipe. We can summarize the Geo Linked Data recipe as follows (see Figure 5.13b and Figure 5.13a):

1. Each URI is dereferenced following the Linked Data principles.
2. If such a URI is dereferenced accepting a Geospatial Web MIME-type (e.g. *application/ vnd.ogc.gml*), the server must return a geospatial description of the entity that matches the MIME-type (e.g. GML) when available.
3. RDF descriptions may contain properties that advertise the presence and role of Geospatial Web descriptions; clients can use these clues to discover their MIME-type and the best use of these representations.
4. RDF descriptions may contain RDF links to navigate to geospatial descriptions provided by Geospatial Web services.



(a) Simple setup with a Geo Linked Data access service.

(b) The content negotiation model of Linked Data is extended with support to Geospatial Web MIME-types.

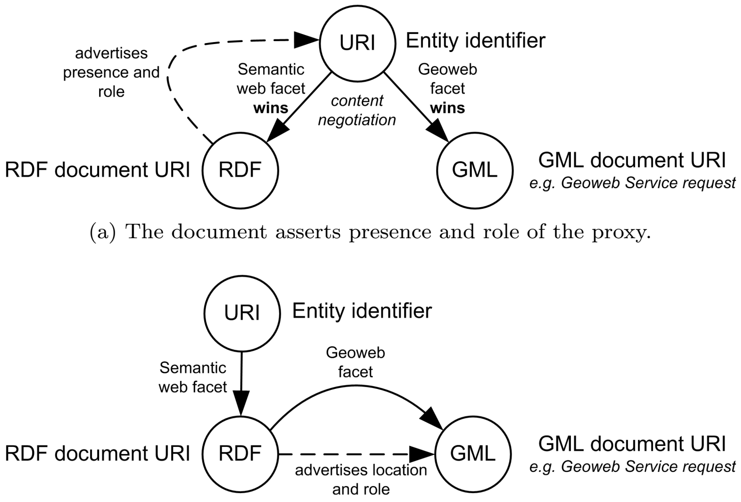
Figure 5.13: Schematic representation of the relations in a simple setup. With a URI that identifies a real world resource (e.g. Lisbon), the user client can access a RDF or HTML representation through the Geo Linked Data Server in the same server (*application/rdf+xml* or *text/html* wins), or a geospatial representation in a Geospatial Web service (*application/vnd.ogc.gml wins*).

We can use RDF to advertise the *presence*, the *role* and the *location* of a geospatial proxy. We can relate a URI to a geospatial proxy adding the next statement to the RDF description of the entity:

```
<entity URI> p <MIME-type> .
```

The semantics of the property `p` should entail that there is a dereferenceable geospatial proxy for the entity identified by the URI. The property `p` could also describe the role of the proxy. Figure 5.14a shows how this assertion should be interpreted. The URI can be dereferenced again, but this time requesting the MIME-type asserted in the statement, to retrieve the geospatial representation. The server that owns the URI is responsible for redirecting the user to the effective location of the geospatial proxy. This way of advertising presence is limited to only one proxy for each content type. If the server cannot be properly configured, or the entity has several geospatial proxies with the same MIME-type, we could make an explicit advertisement of the location of each geospatial proxy. The advertisement requires the assertion of statements like:

```
<entity URI> q <Geospatial Web document URI> .
```



(a) The document asserts presence and role of the proxy.

(b) The document asserts explicit location and role of the proxy.

Figure 5.14: Advertisement of geospatial proxies in RDF documents.

The semantics of the property `q` should assert that the Geospatial Web resource is a geospatial proxy, and, additionally, describe its role. Figure 5.14b shows this approach in detail.

Implementation with the Linked OWS engine. *Geo-Net-PT 02* (Lopez-Pellicer et al., 2010c) is an authoritative RDF dataset about named places of Portugal described in Chapter 4. The administrative features of Portugal are part of this dataset. Geospatial official data about these administrative features can be found in the Official Administrative Boundaries Map (CAOP) of Portugal maintained by the *Instituto Geográfico Português.* The CAOP dataset is available through an OGC Web data access service application (Julião et al., 2009), which makes accessible up-to-date GML documents that are *exact geospatial proxies* of the footprints of the administrative units. Figure 5.15 describes an example of the advertisement of presence, role and location of geospatial proxies in the Geo-Net-PT 02 dataset. The exact link is possible for administrative units. Each administrative unit in Portugal has a unique official identifier as attribute, which exists in the CAOP server and in Geo-Net-PT 02.

Linked OWS engine identifies each resource related with an OGC request to the CAOP server, enabling the redirection using the RESTful binding procedure. It allows clients to dereference a Geo-Net-PT 02 URI, such as `:LisboaFootprint`, and discover that a representation in GML is also available. Then, the client can dereference again but asking for the MIME-type of GML. Then, the linked OWS engine redirects to a URI in the server URI space. As the CAOP supports GET requests for retrieving GML representations, the linked OWS engine redirects the client to the CAOP server. Then the client performs the request to the CAOP server, retrieve the GML representation and then, for example, display the result on a map.

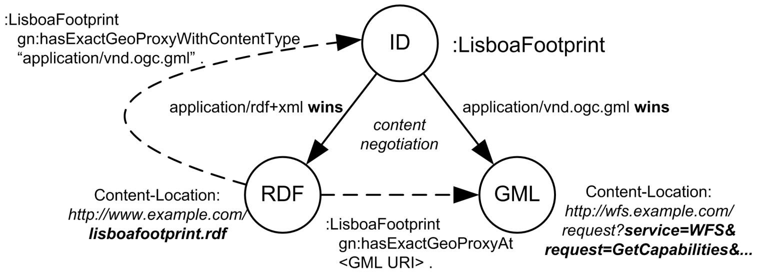


Figure 5.15: Use case scenario: geospatial proxies for Geo-Net-PT 02; the prefix `gn:` identifies the terms added to the vocabulary for advertising Geo Linked Data.

Lessons learned. The research in the geospatial proxies and the Geo Linked Data proposal is derived from a problem not yet solved with the current approach of the Linked OWS engine: the description of parts of geographic content. The request of a part of a continuous geographic feature is the purpose of the interaction *Get<X>Part* with WMS and WCS instances. However, the set of possible partitions of a continuous geographic feature is uncountable. Hence, the approach of collections/items is not applicable for this kind of interactions.

Geospatial proxies offer a possible solution. We can use items, i.e. discrete features, spatially related with a continuous feature and whose footprint might be present in the continuous geographic feature (e.g. a house is a discrete feature, but its shape is part of an image of the area where the house is located), to describe the continuous geographic feature. This relation should be described in terms of the gradation of geospatial proxy roles outlined before.

The proposal for the discovery of geospatial proxies is an alternative to the semantic endpoint of the Linked OWS engine. Instead of returning a wired OGC service metadata description of the CAOP server that the user agent should interpret, the user agent discovers the existence of the content advertised in the returned RDF models. That is, there are interactions that only require to know which content type supports an endpoint. The development of a basic vocabulary able to describe the content types accepted by a resource identified by a URI, and appropriate clients is a line of future research.

5.6 Summary of the Chapter

This chapter has presented and characterized an approach for the diffusion of Geospatial Web content based in the REST architectural style and best practices developed within the Semantic Web. The objective of this approach is to reduce the invisibility of the Geospatial Web. Part of the chapter is devoted to present real scenarios where this approach could help to reduce the social disconnection of the Geospatial Web.

The proposed approach relies on technologies developed in the Semantic Web for the access to descriptions of resources, that is, Linked Data technologies. These technologies can give access to descriptions of Geospatial Web services, such as the descriptions found with the crawler described in Chapter 2, and wrap semantically the accessible Geospatial data. As a natural evolution, the proposed solution can act as a service mediator. This mediator can return service metadata documents where the URIs that identify service endpoints are replaced by URIs managed by the mediator. This enables to use the same URI for querying a service for resources and for dereferencing a machine-processable description of the query. The mediator forwards geospatial queries to remote Geospatial Web service, and manages requests for query descriptions. In addition, the control over the service metadata document allows rewriting all the operations as RESTful. If the remote server does not allow a RESTful request (e.g. a *GetDataSubset* interaction with operations that only allows POST requests), the mediator can interact on behalf the client with the server. Otherwise, the remote server allows a RESTful request and then the mediator forwards the request. From the point of view of the client, the interaction follows the RESTful constraints.

The chapter ends with the presentation of two prototypes that use the proposed solution. The first prototype is focused on the diffusion of the geospatial content. The second prototype is an example of the possible research applications in the field of the Geospatial semantics of the service mediator. Both prototypes allow client applications unaware of the complexities of the Geospatial Web to browse transparently Geospatial content and to access machine processable descriptions of this content. Therefore, a spatial search engine could index this content, and a user can link to it in the Web of Documents and in the Web of Data.

Chapter 6

Conclusions

This chapter summarises the work presented in this thesis, evaluates its contributions, mentions some limitations, and conclude with ideas for further research.

6.1 Summary of Contributions

Chapter 1 presents the concept Geospatial Web. It is identified as the collection of Web services, geospatial data and metadata that supports the use of geospatial data in a range of domain applications. Also introduces OGC, a non-profit organization that leads the development of open and standardized Web service interface specifications for the Geospatial Web. Next, this chapter identifies the problems that this thesis addresses. These problems are related with the invisibility of Geospatial Web services and its consequences, such as the consideration of the Geospatial Content as part of the deep Web and the disconnection of the Geospatial Web from the rest of the Web. The thesis is about the research of a possible solution of the problems derived from the invisibility of Geospatial Web services based in a systematic crawl of the Web for Geospatial Web services combined with the publication as Linked Data of the descriptions and the contents of the discovered Geospatial Web services. This research is scoped to the Web services standardized by OGC.

Chapter 2 provides analyses of the causes of invisibility of the Web. Next, the concepts *invisible Geospatial Web* and *deep Geospatial Web* are introduced. The term *invisible Geospatial Web* highlights that the Geospatial Web content has a priori the same problems of invisibility that the rest of the Web. The term *deep Geospatial Web* is a specialization of the term deep Web that identifies worthwhile Geospatial content invisible to ordinary search engines because it is behind a Geospatial Web service. Henceforth, the study is restricted to OGC Web services, which are endorsed by many public-led geospatial interoperability initiatives. A rationale for crawling of the Web for OGC Web services is provided as justification of the mitigation strategy outlined in Chapter 1. Then, the state-of-the-art of these crawlers is presented. The analysis of these crawlers reveals three main challenges for crawling efficiently the Web for these services: the need of appropriate heuristics for the discovery of disconnected geospatial content, the need of geospatial sensitive crawl ordering policies, and the need of a guarantee of crawl completeness. Next, the chapter details the architecture of an advanced focused crawler with geospatial extension. This architecture is the blueprint of a prototype developed from scratch. The chapter ends with the application of the prototype for discovering OGC Web services and measuring the degree of visibility of these services in general purpose search engines. The prototype seems to improve existing crawlers, and helps to debunk some assumptions found in the literature about the search of geospatial services in search engines.

Chapter 3 and Chapter 4 describe respectively two ontologies named OntoOWS and Geo-Net that have been formalized and implemented. The purpose of these models is to provide to the crawler and to the publication engine a common representation model of the discovered Geospatial Web services and their geospatial content. The OntoOWS is an ontology for describing OGC Web services based on the OGC Web services abstract architecture and the *Reference Model of Open Distributed Processing* (RM-ODP, ISO/IEC 10746-2 Foundations, ISO/IEC 10746-3 Architecture, ISO 15414 Enterprise language) family of standards. The Geo-Net ontology is a place ontology that evolves from the GKB metamodel developed in the project GREASE.

Chapter 3 introduces the approach of this thesis for the modelling OGC Web services. This approach is based in the mapping of concepts found in the OCG Web service metadata documents of each Web service into an ontology derived from concepts of the OGC specifications. The development of the ontology, named OntoOWS, requires a clear delimitation of the scope of the ontology and the use of a well-defined procedure for its development. This thesis proposes the use of an adaptation of the Methontology framework for the construction of the ontology.

The OntoOWS ontology provides a framework that helps to translate in a consistent way OGC Web service metadata encoded into assertions about the service instances, allowing further enrichment. These assertions include not only information about the interfaces, the data types and the platform bindings but also expected behaviour, policies, and technological choices. It is out of the scope of the thesis to detail the implementation of the transformation of OGC Web service metadata documents into knowledge representation models based in the OntoOWS ontology. Nevertheless, part of the chapter is dedicated to provide an overview of the transformation procedure.

Chapter 4 introduces the use of the gazetteer data model for the formalization of a minimum content model. The formalization, named Geo-Net, is an extension of the GKB metamodel. The Geo-Net ontology provides a simple and flexible framework that helps to translate in a consistent way descriptions of features found in WFS servers. The Geo-Net ontology captures the elements that define a feature: its place names, its footprints, its feature types and its relations. It is out of the scope of the thesis to detail the implementation of the transformation of descriptions of features encoded in OGC encoding standards into knowledge representation models based in the Geo-Net ontology. Nonetheless, part of the chapter is dedicated to introduce tools that help to use the Geo-Net ontology and results consequence of transformation procedures. In addition, the chapter highlights the parallelism between the role of Geo-Net as minimum content model for data, and the role of Dublin Core as de-facto minimum content model in the Geospatial Web.

Chapter 5 presents and characterizes an approach for the diffusion of Geospatial Web content based in the REST architectural style and best practices developed within the Semantic Web. The objective of this approach is to reduce the invisibility of the Geospatial Web. Part of the chapter is devoted to present real scenarios where this approach could help to reduce the social disconnection of the Geospatial Web.

The approach is materialized in a framework named Linked OWS. The proposed approach relies on technologies developed in the Semantic Web for the access to descriptions of resources, that is, Linked Data technologies. These technologies can give access to descriptions of Geospatial Web services, such as the descriptions found with the crawler described in Chapter 2, and wrap semantically the accessible Geospatial data. As a natural evolution, the proposed solution can act as a service mediator. This mediator can return service metadata documents where each URI that identifies a service endpoint is also a URI part of the Web of Data. This enables to use the same URI for querying a service for resources and for dereferencing a machine-processable description of the query. The mediator forwards geospatial queries to remote Geospatial Web service, and manages requests for query descriptions. In addition, the control over the service metadata document allows rewriting all the operations as RESTful. If the remote server does not allow a RESTful request (e.g. a *GetDataSubset* interaction with operations that only allows POST requests), the mediator can interact on behalf the client with the server. Otherwise, the remote server allows a RESTful request and then the mediator forwards the request. From the point of view of the client, the interaction follows the RESTful constraints.

The chapter ends with the presentation of two prototypes that use the proposed solution. The first prototype is focused on the diffusion of the geospatial content. The second prototype is an example of the possible research applications in the field of the Geospatial semantics of the service mediator. Both prototypes allow client applications unaware of the complexities of the Geospatial Web to browse transparently Geospatial content and to access machine processable descriptions of this content. Therefore, a spatial search engine could index this content, and a user can link to it in the Web of Documents and in the Web of Data.

6.2 Future Work

The goal of this thesis is to increase the chances for finding Geospatial Web services and part of their contents in the Web. Many open questions and problems remain that require further research. These are the opportunities identified for following they up:

1. **Different Geospatial Web services**. This thesis only considers the OGC Web services. The analysis of the state-of-art showed that some studies have discovered a huge amount of non-OGC Web services. There are other relevant Geospatial Web services, such as those based in the open standard OPeNDAP[1], and in proprietary protocols, mainly the family of ESRI ArcGIS services. Each standard defines a clear set of interfaces and domain models. It is important to extend the support to these services. Therefore, the research should address not only its discovery, but also an appropriate modelling and publication.

2. **Alternative metadata record schemas for Geospatial Web services.** This thesis only considers OGC Web service metadata documents as OGC Web services description providers. Other service metadata schemas in the geospatial domain, such as the ISO/TC 211 Geographic Metadata schema for geospatial services (ISO 19119:2005), can also provide rich descriptions of Geospatial Web services, including OGC Web services. It seems natural to perform further research on the discovery and knowledge extraction from non-OGC metadata schemas.

3. **W3C Web services with Geospatial content**. It is reasonable to perform future research the degree of visibility of W3C Web services with Geospatial content. However, the analysis of W3C Web services is challenging. Any set of system functions can be exposed to the Web with W3C Web services. Thus, the research should address first the discovery of W3C Web services, and then, its classification as services with geospatial content. The model of each W3C geospatial Web service is potentially unique. The investigation of the possible semantic mapping from these models to the models of standard Geospatial Web services is a challenge.

4. **Scripted applications**. Geospatial scripted applications shatter the Web page metaphor on which the Web crawling techniques relies. The combination of scripted languages, the DOM manipulation of Web pages on the client side, along with asynchronous server communication transforms a simple Web document in a complex state machine that relies heavily on user interaction. Geospatial Web applications based in Java Script libraries, such as Google Maps API[2], OpenLayers[3], and Mapstraction[4] are visible examples. Given the success of these applications, there is a practical need to develop techniques able to parse them.

5. **Link analysis**. This thesis has identified several crawl ordering policies that can be applied to the focused crawl of Geospatial Web services. Examples show that a hybrid heuristic that uses a shark search approach with a basic learning mechanism yields good results. However, it would be interesting to evaluate strategies based in an implicit conveyance of authority in the links. For example, the evaluation would analyse if to which extent it is possible to relate the

[1] `http://opendap.org/`
[2] `http://code.google.com/apis/maps/`
[3] `http://openlayers.org/`
[4] `http://www.mapstraction.com/`

fact of having links to visible or invisible Geospatial content with being pointed from pages with visible Geospatial content.

6. **Curated datasets of services**. It could be desirable to distribute periodically curated dataset with the OGC Web services found with the technology developed in this thesis in RDF format and in ISO SRV format. National agencies that maintain geospatial catalogues, and the Linked Open Data initiative are the natural distribution channel of this dataset. Therefore, the agencies can merge this dataset with their service catalogues increasing the visibility of services from other institutions within their organizations, and the practitioners of Linked Data can start to link and explore the services in the dataset.

7. **More Geospatial contents**. The research about the geospatial content is restricted in this thesis to simple features and metadata with crosswalks to Dublin Core. OGC Web services can return imagery, live data, array data, models, routes, and so on. Accessing and exposing any of these contents in a uniform way is a difficult task. However, it would be interesting the research in imagery, live data and array data. Currently, OGC is developing new standards and updating existing standards related with imagery, data generated by sensors, array-oriented scientific data[5].

8. **Improve existing models**. This thesis assumes that features are simple and that metadata records contents can be extracted with crosswalks. These approaches are limited. For example, a WFS server can return features that implements CityGML (Gröger et al., 2008) for the representation, storage and exchange of virtual 3D city and landscape models. CityGML is based on a rich, general-purpose information model in addition to geometry and appearance information. On the other side, an ISO GMD record can have more than 300 elements distributed in sections and subsections whose minimum core is composed by 22 elements. Existing crosswalks often map the minimum core to the 15 elements of Dublin Core (see Nogueras-Iso et al., 2005). An improvement of existing models could be made alongside with the research in new geospatial contents.

9. **Temporal support**. The ontologies developed in this thesis provide an inventory of entities existing at a time. However, services and spatial data are dynamic and change along time. For example, a Web service can change part of its contents between two crawling sessions. Modelling the dynamic of services and their content along time is a complex problem. Future work should be based in the research done in upper spatio/temporal ontologies such as BFO (Grenon and Smith, 2004).

10. **Software product line**. The technologies developed in this thesis can be transferred to agencies responsible of the development of SDIs. The technology transfer can be performed

[5] `http://www.opengeospatial.org/standards/requests`

as-is. It requires the development of production software whose main functionality is a complete integration of the technologies developed in this thesis with the stack of existing technologies. Although, a library may provide support for integration by putting all the components required on it, this approach is probably impracticable for the development of production software because of the high variability among OGC standards, and the future extension towards the inclusion of other standards. The analysis of the interactions of the geospatial Web services has show that they can be separated in different concerns. That is, similar or identical operations with similar or identical signatures with the same concern are shared by OGC Web services. It seems promising the research in software product line approaches to take advance of this characteristic.

11. **Semantic Geospatial Web**. The vision about the Semantic Geospatial Web outlined by Egenhofer (2002) is about better retrieval methods by incorporating the geospatial semantics and exploiting the geospatial semantics during the search process. This thesis gives the possibility to develop a global search engine for geospatial services and data that could use machine processable descriptions of resources to improve the retrieval. This research opportunity requires additional research in spatial semantics (e.g. scale, name-coordinate translation, resource boundary), and in service semantics (e.g. quality of service, service equivalence, service level).

6.3 Conclusions

The central result of the thesis is that large parts of the Geospatial Web remain part of the invisible Web due to non-technical reasons. Geospatial Web services are indexable with an appropriate crawler. Hence, the contents of geospatial databases and datasets that these services give access have a chance to be indexed. It is possible to ease the indexing of these services and content by third party crawlers if they are surfaced as part of the Semantic Web. However, existing Geospatial Web clients might exploit the surfaced information only if the publication is sensitive to them. A SDI worried by the completeness of its catalogues and the visibility of the infrastructure assets among the data consumers following the strategy outlined in this thesis. That is, crawl geospatial metadata instead of waiting their upload, and then, publish the stored metadata as processable data instead of giving access only to the stored metadata. This approach might boost the chances of being indexed by search engines. Once indexed, there is a chance for being found in a resource search, used, commented, linked, and so on. In definitive, there is a chance to reduce the social disconnection of the Geospatial Web.

Appendix A

OntoOWS

This is the encoding of the ontology OntoOWS in OWL 2 using the Manchester Syntax (Horridge and Patel-Schneider, 2009). The ontology is identified with the URI `http://purl.org/iaaa/sw/ontoows`, and its concepts have their names prefixed by the string `ows:` that maps to the string `http://purl.org/iaaa/sw/ontoows#`.

```
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: xml: <http://www.w3.org/XML/1998/namespace>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix: skos: <http://www.w3.org/2004/02/skos/core#>
Prefix: ows: <http://purl.org/iaaa/sw/ontoows#>
Prefix: gn: <http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net#>
Prefix: tm: <http://www.w3.org/2006/time#>
Prefix: vcard: <http://www.w3.org/2006/vcard/ns#>
Ontology: <http://purl.org/iaaa/sw/ontoows>
    Import: <geo-net.owl>
    Import: <http://www.w3.org/2006/time>
    Import: <http://www.w3.org/2006/vcard/ns>

Datatype: rdf:PlainLiteral
Datatype: xsd:anyURI
Datatype: xsd:string
Datatype: xsd:hexBinary
```

```
ObjectProperty: ows:plays
    InverseOf:
        ows:isPlayedBy

ObjectProperty: ows:specifies
    InverseOf:
        ows:isSpecifiedBy

ObjectProperty: ows:subset
    Characteristics:
        Transitive
    InverseOf:
        ows:superset

ObjectProperty: ows:hasPayload

ObjectProperty: ows:isValueOf
    InverseOf:
        ows:hasValue

ObjectProperty: ows:isPlayedBy
    InverseOf:
        ows:plays

ObjectProperty: ows:isReplacedBy
    Characteristics:
        Transitive
    InverseOf:
        ows:replaces

ObjectProperty: ows:isOwnedBy
    Characteristics:
        Functional
    InverseOf:
        ows:owns

ObjectProperty: ows:allowedType
```

```
ObjectProperty: ows:coverage

ObjectProperty: ows:toPlatform

ObjectProperty: ows:owns
    InverseOf:
        ows:isOwnedBy

ObjectProperty: ows:hasBusinessCard

ObjectProperty: ows:isSpecifiedBy
    InverseOf:
        ows:specifies

ObjectProperty: ows:isPartOf
    InverseOf:
        ows:composedOf

ObjectProperty: ows:hasException
    SubPropertyOf:
        ows:hasMessage

ObjectProperty: ows:hasOperation

ObjectProperty: ows:allowedValue

ObjectProperty: ows:hasFirst
    SubPropertyOf:
        ows:hasMessage

ObjectProperty: ows:replaces
    Characteristics:
        Transitive
    InverseOf:
        ows:isReplacedBy
```

```
ObjectProperty: ows:hasNext

ObjectProperty: ows:isRoleOf
    Characteristics:
        Functional
    InverseOf:
        ows:hasRole

ObjectProperty: ows:describes

ObjectProperty: ows:hasRole
    InverseOf:
        ows:isRoleOf

ObjectProperty: ows:defines

ObjectProperty: ows:import
    SubPropertyChain:
        ows:merge o ows:import
    Characteristics:
        Transitive

ObjectProperty: ows:hasBinding
    InverseOf:
        ows:bindingFor

ObjectProperty: ows:composedOf
    InverseOf:
        ows:isPartOf

ObjectProperty: ows:hasValue
    InverseOf:
        ows:isValueOf

ObjectProperty: ows:superset
    Characteristics:
        Transitive
    InverseOf:
        ows:subset

ObjectProperty: ows:withEncoding

ObjectProperty: ows:hasKeywords

ObjectProperty: ows:hasMessage
    SubPropertyChain:
        ows:hasMessage o ows:hasNext

ObjectProperty: ows:hasType

ObjectProperty: ows:usesEndpoint

ObjectProperty: ows:mandatoryPart
    SubPropertyOf:
        ows:composedOf

ObjectProperty: ows:bindingFor
    InverseOf:
        ows:hasBinding

ObjectProperty: ows:hasInput
    SubPropertyOf:
        ows:hasMessage

ObjectProperty: ows:hasOutput
    SubPropertyOf:
        ows:hasMessage

ObjectProperty: ows:profileOf

ObjectProperty: ows:merge
    Characteristics:
        Transitive
```

```
DataProperty: ows:uriTemplate


DataProperty: ows:location


DataProperty: ows:updateSequence


DataProperty: ows:title


DataProperty: ows:serviceType


DataProperty: ows:version


DataProperty: ows:abstract


DataProperty: ows:namespace


DataProperty: ows:verb


DataProperty: ows:serviceTypeVersion


DataProperty: ows:name


DataProperty: ows:literalForm


Class: ows:Object
    SubClassOf:
        owl:Thing


Class: ows:Profile
    EquivalentTo:
        ows:ImplementableStandard
         and (ows:profileOf some owl:Thing)
         and (ows:profileOf only
            (ows:AbstractSpecification
             or ows:ImplementationStandard))


Class: ows:PolicyObject
```

```
    SubClassOf:
        ows:specifies only
            (ows:Community
             or ows:Interaction
             or ows:Object
             or ows:Role)

Class: ows:Payload
    SubClassOf:
        (ows:allowedValue some
            (ows:ExtendedLiteral
             or (ows:name some xsd:string)
             or (ows:namespace some xsd:anyURI)))
         and (ows:allowedType only ows:InformationType)
         and (ows:composedOf only ows:Payload)
         and (ows:name only xsd:string)
         and (ows:name max 1 xsd:string)

Class: ows:Contract
    EquivalentTo:
        (ows:coverage some tm:TemporalEntity)
         and (ows:owns some ows:Interaction)
         and (ows:owns some ows:InterfaceRole)
         and (ows:owns only
            (ows:Interaction
             or ows:PolicyObject
             or ows:Role))
         and (ows:specifies only ows:Community)
         and (ows:owns min 2 ows:Role)
         and (ows:specifies exactly 1 owl:Thing)

Class: ows:Role
    EquivalentTo:
        ows:isRoleOf some ows:Interaction
    SubClassOf:
        ows:isPlayedBy only ows:Object
```

```
Class: ows:Artifact
    SubClassOf:
        ows:Role

Class: ows:Platform
    SubClassOf:
        owl:Thing

Class: ows:Binding
    EquivalentTo:
        (ows:usesEndpoint some owl:Thing)
         and (ows:withEncoding some owl:Thing)
         and (ows:toPlatform only ows:Platform)
         and (ows:usesEndpoint only ows:Endpoint)
         and (ows:withEncoding only ows:Encoding)
         and (ows:toPlatform exactly 1 ows:Platform)

Class: ows:ContactServiceProvider
    EquivalentTo:
        ows:Interaction
         and (ows:hasRole some
            ((ows:Artifact
             and ows:SpatialAsset)
             and (ows:isPlayedBy only
                (ows:Metadata
                 or ows:SpatialDataset
                 or ows:SpatialService))))
         and (ows:hasRole some
            ((ows:Resource
             and ows:SpatialAssetMetadata)
             and (ows:isPlayedBy only ows:OWSMetadata)))
         and (ows:hasRole exactly 1 (ows:Initiator
         and (ows:isPlayedBy only ows:Agent)))
         and (ows:hasRole exactly 1 ((ows:Contributor
         and ows:Responder)
         and (ows:isPlayedBy only ows:Agent)))
```

```
Class: ows:InformationType
    SubClassOf:
        (ows:coverage only
            (gn:GeographicEntity
             or tm:TemporalEntity))
         and (ows:hasKeywords only ows:ExtendedLiteral)
         and (ows:isSpecifiedBy max 1 ows:InformationModel)
         and (ows:abstract only rdf:PlainLiteral)
         and (ows:name only xsd:string)
         and (ows:title only rdf:PlainLiteral)
         and (ows:name exactly 1 xsd:string)

Class: ows:OWSCommunity
    EquivalentTo:
        ows:Community
         and (ows:composedOf some ows:OWSMetadata)
         and (ows:composedOf some ows:SpatialService)
         and (ows:isSpecifiedBy some ows:OWSContract)
    SubClassOf:
        ows:Community
         and (ows:hasKeywords only ows:ExtendedLiteral)
         and (ows:abstract only rdf:PlainLiteral)
         and (ows:title only rdf:PlainLiteral)

Class: ows:Metadata
    EquivalentTo:
        ows:Dataset
         and (ows:describes some owl:Thing)
         and (ows:isSpecifiedBy some owl:Thing)
    DisjointWith:
        ows:SpatialDataset

Class: ows:Agent
    SubClassOf:
        ows:Object
    DisjointWith:
        ows:Dataset, ows:DatasetAggregate
```

```
Class: ows:OWSContract
    EquivalentTo:
        ows:Contract
         and (ows:specifies some ows:OWSCommunity)
         and (ows:owns exactly 1 ows:GetCapabilities)

Class: ows:Operation
    EquivalentTo:
        (ows:hasFirst some ows:Message)
         and (ows:hasBinding only ows:Binding)
         and (ows:hasMessage only ows:Message)
         and (ows:name only xsd:string)
         and (ows:name exactly 1 xsd:string)

Class: ows:Policy
    SubClassOf:
        ows:PolicyObject
         and (ows:name exactly 1 xsd:string)
    DisjointWith:
        ows:PolicyValue

Class: ows:Message
    SubClassOf:
        (ows:hasNext only ows:Message)
         and (ows:hasPayload only ows:Payload)
         and (ows:hasPayload max 1 ows:Payload)

Class: ows:OWSMetadata
    SubClassOf:
        ows:Metadata
         and (ows:describes some ows:OWSContract)
         and (ows:location some xsd:anyURI)
         and (ows:serviceTypeVersion some xsd:string)
         and (ows:location only xsd:anyURI)
         and (ows:serviceType only xsd:string)
         and (ows:serviceTypeVersion only xsd:string)
         and (ows:updateSequence only (xsd:hexBinary or xsd:string))
         and (ows:serviceType exactly 1 xsd:string)
         and (ows:updateSequence max 1 (xsd:hexBinary or xsd:string))
    HasKey:
        ows:location,
        ows:serviceType,
        ows:serviceTypeVersion,
        ows:updateSequence

Class: ows:Contributor
    SubClassOf:
        ows:SpatialRole
         and (ows:hasBusinessCard only vcard:VCard)
    DisjointWith:
        ows:SpatialAsset

Class: ows:ApplicationSchemaStandard
    SubClassOf:
        ows:ImplementationStandard
         and (ows:defines some owl:Thing)
         and (ows:defines only ows:XMLSchema)
         and (ows:specifies only ows:InformationType)

Class: ows:Responder
    SubClassOf:
        ows:Actor
    DisjointWith:
        ows:Initiator

Class: ows:SpatialDataset
    SubClassOf:
        ows:Dataset
    DisjointWith:
        ows:Metadata

Class: ows:DatasetAggregate
    EquivalentTo:
        ows:Object
         and (((ows:composedOf some owl:Thing)
         and (ows:composedOf only ows:Dataset))
         or ((ows:subset some owl:Thing)
         and (ows:subset only ows:DatasetAggregate))
         or ((ows:superset some owl:Thing)
         and (ows:superset only ows:DatasetAggregate)))
    DisjointWith:
        ows:Agent

Class: ows:Community
    EquivalentTo:
        (ows:composedOf some owl:Thing)
         and (ows:isSpecifiedBy some ows:Contract)
         and (ows:composedOf only ows:Object)

Class: ows:FeatureType
    SubClassOf:
        ows:InformationType
         and (ows:coverage min 1 gn:GeographicEntity)

Class: ows:InterfaceRole
    EquivalentTo:
        ows:Role
         and (ows:hasOperation some owl:Thing)
         and (ows:hasOperation only ows:Operation)
    DisjointWith:
        ows:SpatialAssetMetadata

Class: ows:GetDataSubset
    SubClassOf:
        ows:Interaction
         and (ows:hasRole some
            ((ows:Artifact
             and ows:SpatialAssetMetadata)
             and (ows:isPlayedBy only ows:OWSMetadata)))
         and (ows:hasRole some
            ((ows:Resource
             and ows:SpatialAsset)
             and (ows:isPlayedBy only
                (ows:Metadata
                 or ows:SpatialDataset))))
         and (ows:hasRole exactly 1 ows:Responder)
         and (ows:hasRole exactly 1 (ows:Initiator
         and (ows:isPlayedBy only ows:Agent)))
         and (ows:hasRole exactly 1 ((ows:InterfaceRole
         and ows:Responder
         and ows:SpatialAsset)
         and (ows:isPlayedBy only ows:SpatialService)))

Class: ows:Coverage
    SubClassOf:
        ows:InformationType
         and (ows:coverage min 1 gn:GeographicEntity)

Class: ows:PolicyValue
    EquivalentTo:
        ows:isValueOf some ows:Policy
    SubClassOf:
        ows:PolicyObject
         and (ows:isValueOf only ows:Policy)
         and (ows:literalForm only rdf:PlainLiteral)
         and (ows:name only xsd:string)
         and (ows:name max 1 xsd:string)
    DisjointWith:
        ows:Policy

Class: owl:Thing

Class: ows:Interaction
    SubClassOf:
        (ows:hasRole only ows:Role)
         and (ows:hasRole min 2 ows:Role)
```

```
Class: ows:Dataset
    SubClassOf:
        ows:Object
    DisjointWith:
        ows:Agent

Class: ows:InformationModel
    SubClassOf:
        (ows:import only ows:InformationModel)
         and (ows:merge only ows:InformationModel)
         and (ows:replaces only ows:InformationModel)
         and (ows:location some xsd:anyURI)
         and (ows:location only xsd:anyURI)
         and (ows:namespace only xsd:string)
         and (ows:version only xsd:string)
         and (ows:namespace exactly 1 xsd:string)
         and (ows:version max 1 xsd:string)

Class: ows:InformationAggregate
    SubClassOf:
        (((ows:composedOf some owl:Thing)
         and (ows:composedOf only ows:InformationType))
         or ((ows:subset some owl:Thing)
         and (ows:subset only ows:InformationAggregate))
         or ((ows:superset some owl:Thing)
         and (ows:superset only ows:InformationAggregate)))
         and (ows:hasKeywords only ows:ExtendedLiteral)
         and (ows:abstract only rdf:PlainLiteral)
         and (ows:title only rdf:PlainLiteral)

Class: ows:Manage
    SubClassOf:
        ows:Interaction
         and (ows:hasRole some
            ((ows:Artifact
             and ows:SpatialAssetMetadata)
             and (ows:isPlayedBy only ows:OWSMetadata)))
         and (ows:hasRole some
            ((ows:Resource
             and ows:SpatialAsset)
             and (ows:isPlayedBy only
                (ows:Metadata
                 or ows:SpatialDataset))))
         and (ows:hasRole exactly 1 ows:Responder)
         and (ows:hasRole exactly 1 (ows:Initiator
         and (ows:isPlayedBy only ows:Agent)))
         and (ows:hasRole exactly 1 ((ows:InterfaceRole
         and ows:Responder
         and ows:SpatialAsset)
         and (ows:isPlayedBy only ows:SpatialService)))

Class: ows:Actor
    SubClassOf:
        ows:Role
    DisjointWith:
        ows:SpatialAssetMetadata

Class: ows:SpatialAssetMetadata
    SubClassOf:
        ows:SpatialAsset
    DisjointWith:
        ows:Actor, ows:InterfaceRole

Class: ows:EncodingStandard
    SubClassOf:
        ows:ImplementationStandard
         and (ows:specifies only ows:InformationType)

Class: ows:ImplementableStandard
    SubClassOf:
        (ows:replaces only ows:ImplementableStandard)
         and (ows:location some xsd:anyURI)
         and (ows:location only xsd:anyURI)
         and (ows:name only xsd:string)
         and (ows:version only xsd:string)
         and (ows:name exactly 1 xsd:string)
         and (ows:version exactly 1 xsd:string)

Class: ows:Endpoint
    SubClassOf:
        (ows:uriTemplate some xsd:string)
         and (ows:uriTemplate exactly 1 xsd:string)

Class: ows:AbstractSpecification
    SubClassOf:
        ows:ImplementableStandard
    DisjointWith:
        ows:ImplementationStandard

Class: ows:Encoding

Class: ows:Resource
    SubClassOf:
        ows:Role
         and (ows:hasType only ows:InformationType)

Class: ows:GetResourceById
    SubClassOf:
        ows:Interaction
         and (ows:hasRole some
            ((ows:Artifact
             and ows:SpatialAssetMetadata)
             and (ows:isPlayedBy only ows:OWSMetadata)))
         and (ows:hasRole some
            ((ows:Resource
             and ows:SpatialAsset)
             and (ows:isPlayedBy only
                (ows:Metadata
                 or ows:SpatialDataset))))
         and (ows:hasRole exactly 1 ows:Responder)
         and (ows:hasRole exactly 1 (ows:Initiator
         and (ows:isPlayedBy only ows:Agent)))
         and (ows:hasRole exactly 1 ((ows:InterfaceRole
         and ows:Responder
         and ows:SpatialAsset)
         and (ows:isPlayedBy only ows:SpatialService)))

Class: ows:Record
    SubClassOf:
        ows:InformationType

Class: ows:GetCapabilities
    EquivalentTo:
        ows:Interaction
         and (ows:hasRole some
            (ows:Resource
             and ows:SpatialAssetMetadata
             and (ows:isPlayedBy only ows:OWSMetadata)))
         and (ows:hasRole exactly 1 ows:Responder)
         and (ows:hasRole exactly 1 (ows:Initiator
         and (ows:isPlayedBy only ows:Agent)))
         and (ows:hasRole exactly 1 ((ows:Responder
         and ows:SpatialAsset
         and (ows:hasOperation some
            (ows:Operation
             and (ows:name only {"Capabilities" , "GetCapabilities"}))))
         and (ows:isPlayedBy only ows:SpatialService)))

Class: ows:SpatialService
    SubClassOf:
        ows:ServiceInstance

Class: ows:SpatialRole
    SubClassOf:
        ows:Role

Class: ows:HttpBinding
    EquivalentTo:
        ows:Binding
         and (ows:toPlatform only ({ows:PlatformHTTP ,
          ows:PlatformREST}))
         and (ows:verb only {"DELETE" , "GET" , "POST" , "PUT"})
         and (ows:verb exactly 1 xsd:string)

Class: ows:ImplementationStandard
    SubClassOf:
        ows:ImplementableStandard
         and (ows:isSpecifiedBy some owl:Thing)
         and (ows:defines only ows:InformationModel)
         and (ows:isSpecifiedBy only ows:AbstractSpecification)
    DisjointWith:
        ows:AbstractSpecification

Class: ows:Layer
    SubClassOf:
        ows:InformationType
         and (ows:coverage min 1 gn:GeographicEntity)

Class: ows:Observation
    SubClassOf:
        ows:InformationType
         and (ows:coverage min 1 gn:GeographicEntity)

Class: ows:InterfaceStandard
    SubClassOf:
        ows:ImplementationStandard
         and (ows:specifies only
            (ows:OWSContract
             and ows:OWSMetadata))

Class: ows:SpatialAsset
    SubClassOf:
        ows:SpatialRole
    DisjointWith:
        ows:Contributor
```

```
Class: ows:ExtendedLiteral
    SubClassOf:
        (ows:literalForm some rdf:PlainLiteral)
         and (ows:literalForm only rdf:PlainLiteral)

Class: ows:XMLSchema
    SubClassOf:
        ows:InformationModel
         and (ows:import only ows:XMLSchema)
         and (ows:merge only ows:XMLSchema)

Class: ows:Initiator
    SubClassOf:
        ows:Actor
    DisjointWith:
        ows:Responder

Class: ows:ServiceInstance
    SubClassOf:
        ows:Agent

Individual: ows:PlatformHTTP
    Types:
        ows:Platform

Individual: ows:PlatformREST
    Types:
        ows:Platform

Individual: ows:KVP
    Types:
        ows:Encoding

Individual: ows:RESTful
    Types:
        ows:Encoding
```

```
Individual: ows:SOAP
    Types:
        ows:Encoding

Individual: ows:XML
    Types:
        ows:Encoding

Individual: ows:PlatformSOAP
    Types:
        ows:Platform

DisjointClasses:
    ows:ApplicationSchemaStandard,ows:EncodingStandard,
    ows:InterfaceStandard

DisjointClasses:
    ows:Encoding,ows:Endpoint,ows:ExtendedLiteral,
    gn:GeographicEntity,ows:ImplementableStandard,
    ows:InformationAggregate,ows:InformationModel,
    ows:InformationType,ows:Interaction,
    ows:Message,ows:Object,ows:Payload,ows:Platform,
    ows:PolicyObject,tm:TemporalEntity,ows:VCard

DisjointClasses:
    ows:Coverage,ows:FeatureType,ows:Layer,ows:Observation,ows:Record

DisjointClasses:
    ows:Actor,ows:Artifact,ows:Resource

DifferentIndividuals:
    ows:PlatformHTTP,ows:PlatformREST,ows:PlatformSOAP

DifferentIndividuals:
    ows:KVP,ows:PlatformHTTP,ows:PlatformREST,
    ows:PlatformSOAP,ows:RESTful,ows:SOAP,ows:XML
```

Appendix B

Geo-Net

This is the encoding of the basic vocabulary for Geo-Net in OWL 2 using the Manchester Syntax (Horridge and Patel-Schneider, 2009). The ontology is identified with the URI `http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net`, and its concepts have their names prefixed by the string `gn:` that maps to the string `http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net#`.

```
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: gn: <http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net#>

Ontology: <http://xldb.di.fc.ul.pt/xldb/publications/2009/10/geo-net>

Datatype: xsd:string

ObjectProperty: gn:footprint
    Characteristics:
         InverseFunctional
    Domain:
         gn:Feature

ObjectProperty: gn:inDomain
    Range:
         gn:InformationDomain

ObjectProperty: gn:relation
    Characteristics:
         Symmetric

ObjectProperty: gn:lineage
    Range:
         gn:Source

ObjectProperty: gn:type

ObjectProperty: gn:referenceSystem
    Characteristics:
         Functional
    Range:
         gn:ReferenceSystem

ObjectProperty: gn:name

DataProperty: gn:representation
    Characteristics:
         Functional

DataProperty: gn:lemma
    Characteristics:
         Functional

DataProperty: gn:languageCode
    Characteristics:
         Functional

DataProperty: gn:geometry
    Characteristics:
         Functional

Class: gn:InformationDomain

Class: gn:Source

Class: gn:GeographicConcept
```

```
Class: gn:Feature
    SubClassOf:
         gn:GeographicEntity
         and (gn:name some gn:PlaceName)
         and (gn:name only gn:PlaceName)
         and (gn:footprint only gn:Footprint)
         and (gn:relation only gn:Feature)
         and (gn:type only gn:FeatureType)
         and (gn:type some gn:FeatureType)

Class: gn:GeographicEntity
    SubClassOf:
         gn:GeographicConcept

Class: gn:Footprint
    SubClassOf:
         gn:GeographicEntity
         and (gn:referenceSystem some gn:ReferenceSystem)
    HasKey:
         gn:referenceSystem,
         gn:geometry

Class: gn:PlaceName
    SubClassOf:
         gn:GeographicEntity
         and (gn:lemma some xsd:string)
         and (gn:languageCode only xsd:string)
    HasKey:
         gn:languageCode,
         gn:lemma

Class: gn:FeatureType
    SubClassOf:
        gn:GeographicConcept
        and (gn:relation only gn:FeatureType)
```

```
Class: gn:ReferenceSystem
    SubClassOf:
        gn:GeographicConcept
    HasKey:
        gn:representation

DisjointClasses:
    gn:GeographicConcept, gn:InformationDomain, gn:Source

DisjointClasses:
    gn:FeatureType, gn:GeographicEntity, gn:ReferenceSystem

DisjointClasses:
    gn:Feature, gn:Footprint, gn:PlaceName
```

Appendix C

Navigation

This is the encoding of the basic vocabulary for Navigation in OWL 2 using the Manchester Syntax (Horridge and Patel-Schneider, 2009). The ontology is identified with the URI `http://purl.org/iaaa/sw/nav`, and its concepts have their names prefixed by the string `ows:` that maps to the string `http://purl.org/iaaa/sw/nav#`.

```
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: xml: <http://www.w3.org/XML/1998/namespace>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix: skos: <http://www.w3.org/2004/02/skos/core#>
Prefix: nav: <http://purl.org/iaaa/sw/nav#>

Ontology: <http://purl.org/iaaa/sw/nav>

ObjectProperty: nav:nextPage

ObjectProperty: nav:isIndexOf
    InverseOf:
        nav:index

ObjectProperty: nav:item

ObjectProperty: nav:previousPage
```

```
ObjectProperty: nav:index
    SubPropertyChain:
        nav:index o nav:nextPage
    SubPropertyChain:
        nav:index o nav:previousPage
    InverseOf:
        nav:isIndexOf

Class: owl:Thing

Class: nav:Index
    EquivalentTo:
        (nav:isIndexOf some owl:Thing) or
        (nav:item some owl:Thing) or
        (nav:nextPage some owl:Thing) or
        (nav:previousPage some owl:Thing)
```

Bibliography

Ontology Definition Metamodel (ODM) v 1.0. OMG Specification formal/2009-05-01, OMG, 2009.

A. I. Abdelmoty, P. Smart, and C. B. Jones. Building place ontologies for the Semantic Web: issues and approaches. In *GIR '07: Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 7–12, New York, NY, USA, 2007. ACM.

L.A. Adamic. The Small World Web. In *Research and advanced technology for digital libraries: third European conference, ECDL'99, Paris, France, September 22-24, 1999*, page 443. Springer Verlag, 1999.

B. Adida, M. Birbeck, S. McCarron, and S. Pemberton. RDFa in XHTML: Syntax and Processing – A collection of attributes and processing rules for extending XHTML to support RDF. W3C Recommendation, W3C, October, 14 2008. URL `http://www.w3.org/TR/rdfa-syntax/`.

ADL. Guide to the ADL Gazetteer Content Standard 3.2. User guide, Alexandria Digital Library (ADL), University of California, Santa Barbara, CA, February 2004.

A. Ager, C. Schrader-Patton, K Bunzel, and B. Colombe. Internet map services: new portal for global ecological monitoring, or geodata junkyard? In *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application*, COM.Geo '10, pages 37:1–37:4, New York, NY, USA, 2010. ACM.

C. C. Aggarwal, F. Al-Garawi, and P.S. Yu. Intelligent crawling on the World Wide Web with arbitrary predicates. In *Proceedings of the 10th international conference on World Wide Web*, pages 96–105. ACM, 2001.

D. Ahlers and S. Boll. Adaptive geospatially focused crawling. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 445–454. ACM, 2009.

AIC and FGDC. Federal enterprise architecture – geospatial profile. Recommendation Version 1.1, Architecture and Infrastructure Committee (AIC), Federal Chief Information Officers Council and Federal Geographic Data Committee (FGDC), 2006.

E. Al-Masri and Q. H. Mahmoud. Discovering Web Services in Search Engines. *IEEE Internet Computing*, 12(3):74–77, 2008a.

E. Al-Masri and Q.H. Mahmoud. Investigating Web services on the World Wide Web. In *Proceeding of the 17th international conference on World Wide Web*, pages 795–804. ACM, 2008b.

Keith Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets. In C. Bizer, T. Heath, T. Berners-Lee, and K. Idehen, editors, *WWW '09: Proceeding of the 18th international conference on World Wide Web, Madrid, Spain, April 20, 2009*, volume 538 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

Alta4. WMS-finder.de, 2010. URL `http://www.wms-finder.de/`.

F. Amardeilh, B. Vatant, N. Gibbins, T. R. Payne, A. Saleh, and H. Wang. Transitioning Applications to Ontologies D1.2 – SWS Bootstrapping Methodology. Deliverable TAO/2008/D1.2/v2.0, 2008.

ANZLIC. The Australian Spatial Data Directory (ASDD) (v2.0). Technical report, ANZLIC, August 2003.

R. Atkinson. Gazetteer service draft candidate implementation specification 0.84. Draft OGC 01-036, Open Geospatial Consortium, March 2001.

S. Auer, J. Lehmann, and S. Hellmann. LinkedGeoData – Adding a spatial Dimension to the Web of Data. In *Proceedings of 8th International Semantic Web Conference (ISWC)*, 2009.

A. Axelrod. On building a high performance gazetteer database. In *Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references*, pages 63–68, Morristown, NJ, USA, 2003. Association for Computacional Linguisitics.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, New York, NY, USA, 2003.

D. Bachlechner, K. Siorpaes, D. Fensel, and I. Toma. Web Service Discovery - A Reality Check. Technical report, DERI Galway, Galway, Ireland, January 2006.

R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez. Crawling a country: better strategies than breadth-first for web page ordering. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 864–872. ACM, 2005.

R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri. The impact of caching on search engines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190. ACM, 2007.

Y. Bai, C. Yang, L. Guo, and Q. Cheng. OpenGIS WMS-based prototype system of spatial information search engine. In *IGARSS'03, Geoscience and Remote Sensing Symposium, 2003*, volume 6, pages 3558–3560. IEEE, 2003.

S. Baillargeon and L. P. Rivest. Rcapture: Loglinear Models for Capture-Recapture in R. Reference manual, Université Laval, Québec, Canada, 2009. URL `http://cran.r-project.org/web/packages/Rcapture/vignettes/RcaptureJSS.pdf`.

J. Bao, S. Hawke, B. Motik, P. F. Patel-Schneider, and A. Polleres. rdf:PlainLiteral: A Datatype for RDF Plain Literals. W3C Recommendation 27 October 2009, W3C, 2009. URL `http://www.w3.org/TR/2009/REC-rdf-plain-literal-20091027/`.

L. Barbosa and J. Freire. Siphoning hidden-web data through keyword-based interfaces. In S. Lifschitz, editor, *Proceedings XIX Simpósio Brasileiro de Bancos de Dados, 18-20 de Outubro, 2004,Brasília, Distrito Federal, Brasil, Anais*, pages 309–321. UnB, 2004.

L. Barbosa and J. Freire. An adaptive crawler for locating hidden-web entry points. In *Proceedings of the 16th international conference on World Wide Web*, pages 441–450. ACM, 2007.

L. Barbosa, H. Nguyen, T. Nguyen, R. Pinnamaneni, and J. Freire. Creating and exploring web form repositories. In *Proceedings of the 2010 international conference on Management of data*, SIGMOD '10, pages 1175–1178, New York, NY, USA, 2010. ACM.

J. Barrasa-Rodriguez and A. Gómez-Pérez. Upgrading relational legacy data to the Semantic Web. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 1069–1070, New York, NY, USA, 2006. ACM.

L.A. Barroso, J. Dean, and U. Holzle. Web search for a planet: The Google cluster architecture. *IEEE micro*, 23(2):22–28, 2003.

J. Bartley. Mapdex: an online index of web mapping services. Slides, Kansas Geological Survey, 2005.

J. K. Batcheller. Automating geospatial metadata generation–an integrated data management and documentation approach. *Computers & Geosciences*, 34(4):387–398, 2008.

S. Batsakis, E. G. M. Petrakis, and E. Milios. Improving the performance of focused web crawlers. *Data & Knowledge Engineering*, 68(10):1001–1013, 2009.

P. Baumann. OGC WCS 2.0 Interface Standard – Core. Interface Standard OGC 09-110r3, Open Geospatial Consortium, October 2010.

D. Beckett and T. Berners-Lee. Turtle – Terse RDF Triple Language. W3C Team Submission, W3C, 14 January 2008. URL `http://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/`.

Dave Beckett. RDF/XML Syntax Specification (Revised). W3C Recommendation, W3C, 2004. URL `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`.

J. Behnke, T.H. Watts, B. Kobler, D. Lowe, S. Fox, and R. Meyer. EOSDIS petabyte archives: tenth anniversary. In *Mass Storage Systems and Technologies, 2005. Proceedings. 22nd IEEE / 13th NASA Goddard Conference on*, pages 81 – 93, 2005.

R. Béjar. *Contributions to the modelling of spatial data infrastructures and their portrayal services.* PhD thesis, Computer Science and System Engineering Department, 2009.

R. Béjar, M. A. Latre, J. Nogueras-Iso, P. R. Muro-Medrano, and F. J. Zarazaga-Soria. An architectural style for spatial data infrastructures. *International Journal of Geographical Information Science*, 23:271–294, March 2009.

K. Belhajjame, S.M. Embury, N. W. Paton, R.D. Stevens, and C.A. Goble. Automatic annotation of web services based on workflow definitions. *ACM Transactions on the Web*, 2(2):1–34, May 2008.

M.K. Bergman. The deep web: Surfacing hidden value. *Journal of Electronic Publishing*, 7(1):07–01, 2001.

M. L. Berman. A Data Model for Historical GIS: The CHGIS Time Series. Technical report, China Historical GIS, Harvard Yenching Institute, 2003.

L. Bernard, I. Kanellopoulos, A. Annoni, and P. Smits. The European geoportal–one step towards the establishment of a European Spatial Data Infrastructure. *Computers, Environment and Urban Systems*, 29(1):15–31, 2005. Geoportals.

T. Berners-Lee. Linked Data – Design Issues, 2006. URL `http://www.w3.org/DesignIssues/LinkedData.html`.

T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). RFC 1738 (Proposed Standard), December 1994. URL `http://www.ietf.org/rfc/rfc1738.txt`. Obsoleted by RFCs 4248, 4266, updated by RFCs 1808, 2368, 2396, 3986.

T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945 (Informational), May 1996. URL `http://www.ietf.org/rfc/rfc1945.txt`.

T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard), January 2005. URL `http://www.ietf.org/rfc/rfc3986.txt`.

T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *Proceedings of the 3rd International Semantic Web User Interaction*, 2006.

T. Berners-Lee, R. Cyganiak, M. Hausenblas, J. Presbrey, O. Seneviratne, and O. E. Ureche. On integration issues of site-specific apis into the web of data. Technical Report 2009-08-14, DERI, August 2009.

D. Berrueta and J. Phipps. Best practice recipes for publishing RDF vocabularies. W3C note, W3C, August 2008. http://www.w3.org/TR/2008/NOTE-swbp-vocab-pub-20080828/.

P. V. Biron and A. Malhotra. XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004, W3C, 2004. URL `http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/`.

C. Bizer, R. Cyganiak, and T. Heath. How to publish linked data on the web. Technical report, Freie Universität Berlin, July 2007. URL `http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/20070727/`.

C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

P. Boldi, M. Santini, and S. Vigna. Paradoxical effects in pagerank incremental computations. *Internet Mathematics*, 2(3), 2005.

D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. Web Services Architecture. W3C Working Group Note 11 February 2004, W3C, 2004. URL `http://www.w3.org/TR/ws-arch/`.

D. Brickley. Basic Geo (WGS84 lat/long) Vocabulary. Technical report, W3C Semantic Web Interest Group, 2004. URL `http://www.w3.org/2003/01/geo/`.

A. Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM, 2002.

J. Brophy and D. Bawden. Is Google enough? Comparison of an internet search engine with academic library resources. In *Aslib Proceedings*, volume 57, pages 498–512. Emerald Group Publishing Limited, 2005.

C. Burns. Retooling for the Digital Data Revolution Geospatial and GIS Technologies. *GEO Informatics*, 12(5):28–30, July/August 2009.

M. Cafarella and D. Cutting. Building nutch: Open source search. *Queue*, 2:54–61, April 2004.

F. Can, R. Nuray, and A. B. Sevdik. Automatic performance evaluation of web search engines. *Information Processing & Management*, 40(3):495–514, 2004.

J. Cardoso. The Semantic Web Vision: Where Are We? *Intelligent Systems, IEEE*, 22(5):84–88, 2007.

N. Cardoso, D. Batista, F. J. Lopez-Pellicer, and M. J. Silva. Where in the wikipedia is that answer? the xldb at the gikiclef 2009 task. In *Working Notes of CLEF 2009*, Corfu, Greece, September 2009.

J. J. Carroll and G. Klyne. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

S. Chakrabarti, M. Van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11-16):1623–1640, 1999.

S. Chakrabarti, B.E. Dom, S.R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the web's link structure. *Computer*, 32(8):60–67, 2002.

A. Chao. An overview of closed capture-recapture models. *Journal of Agricultural, Biological, and Environmental Statistics*, 6(2):158–175, 2001.

A. Chao, P. K. Tsay, S. H. Lin, W. Y. Shau, and D. Y. Chao. The applications of capture-recapture models to epidemiological data. *Statistics in Medicine*, 20:3123–3157, 2001.

M. Chaves. Criação e expansão de geo-ontologias, dimensionamento de informação geográfica e reconhecimento de locais e seus relacionamentos em textos. In *Encontro Linguateca: 10 anos*, Aveiro, Portugal, Sep 2008. In Portuguese.

M. Chaves, M. J. Silva, and B. Martins. A Geographic Knowledge Base for Semantic Web Applications. In C. A. Heuser, editor, *Proceedings of the 20th Brazilian Symposium on Databases*, pages 40–54, Uberlândia, Minas Gerais, Brazil, October, 3–7 2005.

N. Chen, J. Gong, and Z. Chen. A high precision ogc web map service retrieval based on capability aware spatial search engine. In L. Kang, Y. Liu, and S. Y. Zeng, editors, *Advances in Computation and Intelligence, Second International Symposium, ISICA 2007, Wuhan, China, September 21-23, 2007, Proceedings*, pages 558–567, 2007.

N. Chen, L. Di, G. Yu, Z. Chen, and J. He. Geospatial sensor web data discovery and retrieval service based on middleware. In *XXIst ISPRS Congress, 3-11 Jul 2008 Beijing, China*, 2008.

R. Chinnici, S. Weerawarana, J. J. Moreau, and A. Ryman. Web services description language (WSDL) version 2.0 part 1: Core language. W3C recommendation, W3C, June 2007. http://www.w3.org/TR/2007/REC-wsdl20-20070626.

J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1-7):161–172, 1998.

A. Chowdhury and I. Soboroff. Automatic evaluation of world wide web search services. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 421–422. ACM, 2002.

J. T. Coppock and D. W. Rhind. The History of GIS. In D. J. Maguire, M. F. Goodchild, and D. W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, volume 1, pages 21–43. Longmans Publishers, London, 1991.

F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet computing*, 6(2):86–93, 2002.

R. Cyganiak and C. Bizer. Pubby – A Linked Data Frontend for SPARQL Endpoints, 2007. URL `http://www4.wiwiss.fu-berlin.de/pubby/`.

R. Cyganiak and L. Sauermann. Cool URIs for the Semantic Web. W3C note, W3C, December 2008. http://www.w3.org/TR/2008/NOTE-cooluris-20081203/.

R. Cyganiak, R. Delbru, and G. Tummarello. Semantic Web Crawling: A Sitemap Extension. Technical report, DERI Galway, 2007. URL `http://sw.deri.org/2007/07/sitemapextension/`.

R. Cyganiak, F. Maali, and V. Peristeras. Self-service linked government data with dcat and gridworks. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10, pages 37:1–37:3, New York, NY, USA, 2010. ACM.

M. Daly. Coordinate Transformation Services. Implementation Specification OGC 01-009, OpenGIS Consortium Inc., January 2001. Version 1.0.

R. G. Darman. Coordination of Surveying, Mapping, and Related Spatial Data Activities. Circular No. A-16 Revised, Office of Management and Budget, US Government, October l9, 1990. URL `http://www.whitehouse.gov/omb/circulars_a016`.

A. Das Sarma, X. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 861–874. ACM, 2008.

A.R. Dasgupta. SDI Cookbook – Outreach and Capacity Building, January 2009. URL `http://www.gsdidocs.org/GSDIWiki/index.php/Chapter_9`.

DCTG. Darwin core. Tdwg current standard, Darwin Core Task Group (DCTG), Taxonomic Databases Working Group (TDWG), 2010. URL `http://www.tdwg.org/standards/450/`.

P.M.E. De Bra and R. D. J. Post. Information retrieval in the world-wide web: Making client-based searching feasible. *Computer Networks and ISDN Systems*, 27(2):183 – 192, 1994.

J. de la Beaujardiere. Web Map Service – version 1.1.0. OpenGIS Implementation Specification OGC 01-047r2, Open Geospatial Consortium, Inc., Wayland, MA, USA, March 2001.

J. de la Beaujardiere. Web Map Server – version 1.3.0. OpenGIS Implementation Specification OGC 06-042, Open Geospatial Consortium, Inc., Wayland, MA, USA, March 2006.

B. de Longueville. Community-based geoportals: The next generation? concepts and methods for the geospatial web 2.0. *Computers, Environment and Urban Systems*, 34(4):299 – 308, 2010.

E. Della Valle, D. Cerizza, I. Celino, J. Estublier, G. Vega, M. Kerrigan, J. Ramírez, B. Villazón-Terrazas, P. Guarrera, G. Zhao, and G. Monteleone. SEEMP: An Semantic Interoperability Infrastructure for e-Government Services in the Employment Sector. In E. Franconi, M. Kifer, and W. May, editors, *The Semantic Web: Research and Applications*, volume 4519 of *Lecture Notes in Computer Science*, pages 220–234. Springer Berlin / Heidelberg, 2007.

M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused crawling using context graphs. In A. E. Abbadi, M. L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K. Y. Whang, editors, *VLDB 2000, Proceedings of the 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egipt*, pages 527–534. Morgan Kaufmann, 2000.

DLESE. ADN Metadata framework. Medadata framework 0.6.50, Digital Library for Earth System Education (DLESE), 2004. URL `http://www.dlese.org/Metadata/adn-item/0.6.50/index.php`.

H. Dong, F. Hussain, and E. Chang. State of the art in metadata abstraction crawlers. In H. Wo and H. Xie, editors, *IEEE International Conference on Industrial Technologies, Apr 21 2008*, volume 1-6, Chengdu, China, 2008. Institute of Electrical and Electronics Engineers (IEEE).

H. Dong, F. Hussain, and E. Chang. State of the art in semantic focused crawlers. In O. Gervasi, D. Taniar, B. Murgante, A. Laganà, Y. Mun, and M. Gavrilova, editors, *Computational Science and Its Applications – ICCSA 2009*, volume 5593 of *Lecture Notes in Computer Science*, pages 910–924. Springer Berlin / Heidelberg, 2009.

P. Doran, V. Tamma, and L. Iannone. Ontology module extraction for ontology reuse: an ontology engineering perspective. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 61–70. ACM, 2007.

A. Doyle. OpenGIS Web Map Server Interface Implementation Specification. OGC Standard OGC 00-028, Open Geospatial Consortium Inc., 2000. Version 1.0.0.

A. Doyle, C. Reed, J. Harrison, and M. Reichardt. Introduction to OGC Web Services. OGC White paper, Open GIS Consortium, Inc., Wayland, MA, USA, 2001.

M. Duerst and M. Suignard. Internationalized Resource Identifiers (IRIs). RFC 3987 (Proposed Standard), January 2005. URL `http://www.ietf.org/rfc/rfc3987.txt`.

J. Duhl. Rich internet applications. White Paper 3906, IDC, 2003.

M. J. Egenhofer. Toward the semantic geospatial web. In *GIS '02: Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 1–4, McLean, Virginia, USA, 2002. ACM.

M. Ehrig and A. Maedche. Ontology-focused crawling of web documents. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 1174–1178. ACM, 2003.

J. Elson, J. Howell, and J. R. Douceur. Mapcruncher: integrating the world's geographic information. *ACM SIGOPS Operating Systems Review*, 41(2):50–59, 2007.

T. Erl. *Service-Oriented Architecture: Concepts, Technology, and Design.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.

ESRI. ESRI Shapefile Technical Description. White paper, Environmental Systems Research Institute (ESRI), Inc., March 1998. URL `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`.

ESRI. The ArcIMS OGC WMs Connector. Technical report, ESRI, 2003. URL `http://portal.opengeospatial.org/files/files/?artifact_id=5901`.

European Parliament and Council of European Union. Directive of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Directive 2007/2/EC, European Union, 2007.

J. D. Evans. Web Coverage Service (WCS), Version 1.0.0. Interface Standard OGC 03-065r6, Open Geospatial Consortium, October 2003.

M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, 1997.

C. Ferris and J. Farrell. What are Web services? *Communications of the ACM*, 46(6):31, 2003.

FGDC. Content Standard for Digital Geospatial Metadata Workbook. Technical report, Federal Geographic Data Committee (FGDC)), Washington, D.C., 2000. URL `http://www.fgdc.gov/metadata/documents/workbook_0501_bmk.pdf`.

FGDC. The Federal Geographic Data Committee: Historical Reflections – Future Directions. White paper, Federal Geographic Data Committee (FGDC), USA, January 2004. URL `http://www.fgdc.gov/library/whitepapers-reports/white-papers/fgdc-history`.

FGDC. The national geospatial data clearinghouse – factsheet. Technical report, Federal Geographic Data Committee, February 2005.

R. Fielding. *REST: Architectural Styles and the Design of Network-based Software Architectures.* Doctoral dissertation, University of California, Irvine, 2000. URL `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`.

R. Fielding. W3C Technical Architecture Group's resolution on the range of HTTP dereferencing, June 2005. URL `http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html`. Message to the www-tag@w3.org mailing list, 18 Jun 2005.

R. Fielding. REST APIs must be hypertext-driven, October 2008a. URL `http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven`.

R. Fielding. A little REST and Relaxation. Presentation at ApacheCon Europe. Slides available at http://roy.gbiv.com/talks/200804_REST_ApacheCon.pdf, 2008b.

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068 (Proposed Standard), January 1997. URL `http://www.ietf.org/rfc/rfc2068.txt`. Obsoleted by RFC 2616.

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. HTTP/1.1, part 2: Message Semantics. draft-ietf-httpbis-p2-semantics-12, HTTPbis Working Group, Internet Engineering Task Force, 2010. URL `http://tools.ietf.org/html/draft-ietf-httpbis-p2-semantics-12`.

J. Fitzke and R. Atkinson. OGC Best Practices Document: Gazetteer Service - Application Profile of the Web Feature Service Implementation Specification. OpenGIS Publicy Available Standard OGC 05-035r2, Open GIS Consortium Inc., June 2006. Version 0.9.3.

C. Fritz, C. Kirschner, D. Reker, A. Wisplinghoff, and H. Paulheim. Geospatial Web Mining for Emergency Management. In *Sixth international conference on Geographic Information Science, GIScience 2010, Zurich, Switzerland, September 14-17*, 2010.

G. Fu, C. B. Jones, and A. I. Abdelmoty. Building a geographical ontology for intelligent spatial search on the web. In M. H. Hamza, editor, *Databases and Applications, IASTED International Conference on Databases and Applications, part of the 23rd Multi-Conference on Applied Informatics*, pages 167–172, Innsbruck, Austria, 14–16 Feb 2005 2005. IASTED/ACTA Press.

GeoConnections. A Developers' Guide to the CGDI: Developing and publishing geographic information, data and associated services. Technical report, GeoConnections, Ottawa, ON, Canada, November 2007.

Geops. Mapmatters.org, 2010. URL `http://www.mapmatters.org/`.

F.R. Gibotti, G. Câmara, and R.A. Nogueira. Geodiscover–a specialized search engine to discover geospatial data in the web. In *VI Brazilian Symposium in Geoinformatics, GeoInfo*. Citeseer, 2005.

D. Gomes and M. J. Silva. The Viúva Negra crawler: an experience report. *Software: Practice and Experience*, 38(2):161–188, 2008.

GOMWG. *Government of Canada Metadata Implementation Guide for Web Resources, 4th edition*. The Training Sub-group of the Government On-Line Metadata Working Group (GOMWG), Treasury Board of Canada Secretariat, Ottawa, Ontario, Canada, October 2005.

M.F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

J. Goodwin, C. Dolbear, and G. Hart. Geographical linked data: The administrative geography of great britain on the semantic web. *Transactions in GIS*, 12(s1):19–30, 2009.

Google, Inc. Creating Google-friendly sites – Rich snippets (microdata, microformats, RDFa). Technical report, 2010. URL `http://www.google.com/support/webmasters/bin/topic.py?hl=en&topic=21997`.

Google, Inc., Yahoo, Inc., and Microsoft Corporation. Sitemaps protocol. Best practice, 2008. URL `http://www.sitemaps.org/terms.php`.

J. Gosling, B. Joy, G. Steele, and G. Bracha. *The Java (TM) Language Specification*. Addison-Wesley Professional, 2005.

K. Gottschalk, S. Graham, H. Kreger, and J. Snell. Introduction to Web services architecture. *IBM Systems Journal*, 41(2):170–177, 2002.

L. Gravano, P.G. Ipeirotis, and M. Sahami. Qprober: A system for automatic classification of hidden-web databases. *ACM Transactions on Information Systems (TOIS)*, 21(1):1–41, 2003.

J. Greenberg, M.C. Pattuelli, B. Parsia, and W.D. Robertson. Author-generated Dublin Core metadata for web resources: a baseline study in an organization. *Journal of Digital Information*, 2(2):38–46, 2001.

J. Gregorio, R. Fielding, M. Hadley, and M. Nottingham. URI Template. Internet-draft, Internet Engineering Task Force, 2010. URL `http://tools.ietf.org/html/draft-gregorio-uritemplate-04`.

P. Grenon and B. Smith. SNAP and SPAN: Towards dynamic spatial ontology. *Spatial Cognition & Computation*, 4(1):69–104, 2004.

G. Gröger, T. H. Kolbe, A. Czerwinski, and C. Nagel. OpenGIS City Geography Markup Language (CityGML). Encoding Standard OGC 08-007r1, Open Geospatial Consortium, October 2008.

B.N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM, 2003.

M. Gruninger and M.S. Fox. Methodology for the design and evaluation of ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI*, volume 95, 1995.

R. V. Guha and D. Brickley. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903. ACM, 2005.

C. Gutierrez, C. A. Hurtado, and A. Vaisman. Introducing time into rdf. *IEEE Transactions on Knowledge and Data Engineering*, 19:207–218, 2007.

GWG. Guide to Geospatial Intelligence (GEOINT) Standards. Guide, Geospatial Intelligence Standards Working Group (GWG), 2009.

H. Halpin and V. Presutti. An ontology of resources: Solving the identity crisis. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 521–534. Springer Berlin / Heidelberg, 2009.

H. Halpin, R. Iannella, B. Suda, and N. Walsh. Representing vCard Objects in RDF. Member submission, W3C, January 2010. Latest version: http://www.w3.org/Submission/vcard-rdf.

B. Haslhofer and B. Schandl. The oai2lod server: Exposing oai–pmh metadata as linked data. In *Proceedings of the Linked Data on the Web Workshop, Beijing, China, April 22, 2008*. CEUR Workshop Proceedings, 2008. URL `http://CEUR-WS.org/Vol-369/paper03.pdf`. ISSN 1613-0073.

M. Hausenblas. Linked Data Applications. Technical Report 2009-07-26, DERI Galway, 2009a.

M. Hausenblas. Exploiting linked data to build web applications. *Internet Computing, IEEE*, 13(4): 68 –73, 2009b.

P. Hayes. RDF Semantics. W3C Recommendation 10 February 2004, W3C, 2004. URL `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`.

B. He and K.C.C. Chang. Statistical schema matching across web query interfaces. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 217–228. ACM, 2003.

B. He and K.C.C. Chang. Automatic complex schema matching across web query interfaces: A correlation mining approach. *ACM Transactions on Database Systems (TODS)*, 31(1):346–395, 2006.

B. He, C. Li, D. Killian, M. Patel, Y. Tseng, and K.C.C. Chang. A structure-driven yield-aware web form crawler: Building a database of online databases. Technical Report UIUCDCS-R-2006-2752, UILU-ENG-2006-1792, University of Illinois at Urbana–Champaign, July 2006.

B. He, M. Patel, Z. Zhang, and K.C.C. Chang. Accessing the deep web. *Communications of the ACM*, 50(5):94–101, 2007.

J. Hendler and J. Golbeck. Metcalfe's law, Web 2.0, and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):14–20, 2008.

M. Henning. The rise and fall of CORBA. *Commun. ACM*, 51(8):52–57, 2008.

J. R. Herring. Opengis implementation specification for geographic information - simple feature access - part 1: Common architecture. OpenGIS Implementation Specification OGC 06-103r3, Open GIS Consortium Inc., October 2006. Version 1.2.0.

M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur. The shark-search algorithm. an application: tailored web site mapping. *Computer Networks and ISDN Systems*, 30 (1-7):317–326, 1998.

A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4): 219–229, 1999.

L. Hill. Core elements of digital gazetteers: Placenames, categories, and footprints. In *ECDL '00: Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 280–290, London, UK, 2000. Springer-Verlag.

L. Hill, J. Frew, and Q. Zheng. Geographic names: The implementation of a gazetteer in a georeferenced digital library. *D-Lib Magazine*, 5(1):03–11, 1999.

J. R. Hobbs and F. Pan. Time Ontology in OWL. W3C Working Draft 27 September 2006, W3C, December 2006. http://www.w3.org/TR/2006/WD-owl-time-20060927/.

M. Horridge and P. F. Patel-Schneider. OWL 2 Web Ontology Language – Manchester Syntax. W3C Working Group Note, W3C, 27 October 2009. URL `http://www.w3.org/TR/2009/NOTE-owl2-manchester-syntax-20091027/`.

I. Horrocks. Using an expressive description logic: Fact or fiction? In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998*, pages 636–649. Morgan Kaufmann, 1998.

I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible sroiq. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 57–67. AAAI Press, 2006.

IEEE SC. Ieee standard for software test documentation. IEEE Std 829-1998, IEEE Compute Society, 1998.

INSPIRE DT NS. INSPIRE – Network Services Architecture. Technical Report Version 3.0, INSPIRE Network Services Drafting Team Network Services (INSPIRE DT NS), 2008.

INSPIRE DT NS. Technical guidance for inspire discovery services. Technical Guidance v2.0, INSPIRE Drafting Team Network Services, 2009.

ISO/TC 211. ISO 19112: Geographic information – Spatial referencing by geographic identifiers. Published standard ISO 19112:2003, International Organization for Standardization, October 2003a.

ISO/TC 211. ISO 19115: Geographic information – Metadata. Published standard ISO 19115:2003, International Organization for Standardization, October 2003b.

ISO/TC 211. ISO 19119: Geographic information – Services. Published standard ISO 19119:2005, International Organization for Standardization, October 2005.

J. Paul Getty Trust. The Getty Thesaurus of Geographic Names (TGN). http://www.getty.edu/research/conducting_research/vocabularies/tgn/ about.html (last access: 19 February 2007), 2007.

I. Jacobs and N. Walsh. Architecture of the World Wide Web, Volume One. W3C recommendation, W3C, December 2004. http://www.w3.org/TR/2004/REC-webarch-20041215/.

K. Janowicz, S. Schade, A. Bröring, C. Keßler, P. Maué, and C. Stasch. Semantic Enablement for SDIs. *Transactions in GIS*, 14(2):111–129, April 2010.

A. Jhingran. Enterprise information mashups: integrating information, simply. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 3–4. VLDB Endowment, 2006.

J. Jiang, C. J. Yang, and Y. C. Ren. A spatial information crawler for OpenGIS WFS. In L. Liu, X. Li, K. Liu, X. Zhang, and A. Chen, editors, *Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Geo-Simulation and Virtual GIS Environments*, volume 7143, 2008.

C. B. Jones, H. Alani, and D. Tudhope. Geographical information retrieval with ontologies of place. In D. Montello, editor, *Spatial Information Theory*, volume 2205 of *Lecture Notes in Computer Science*, pages 322–335. Springer Berlin / Heidelberg, 2001.

C. B. Jones, R. S. Purves, P. D. Clough, and H. Joho. Modelling vague places with knowledge from the web. *Int. J. Geogr. Inf. Sci.*, 22(10):1045–1065, 2008.

JTC 1/SC 7. ISO/IEC 15414:2069 Information Technology — Open Distributed Processing — Reference Model — Enterprise language. International standard, ISO/IEC, 2006.

JTC 1/SC 7. ITU-T X.902 | ISO/IEC 10746-2:2009 Information Technology — Open Distributed Processing — Reference Model — Foundations. International standard, ITU | ISO/IEC, 2009.

R. P. Julião, S. Mas, A. F. Rodríguez-Pascual, and D. Furtado. Portugal and Spain twin SDI's from national projects to an Iberian SDI. In *GSDI-11:Spatial Data Infrastructure Convergence, Rotterdam, The Netherlands 15-19 June 2009*, 2009.

P. Kalberer. Geopole: Performance and statistical analysis of WMS. In *FOSS4G Barcelona 2010*, 2010.

N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, and C. Barreto. Web Services Choreography Description Language Version 1.0. W3C Candidate Recommendation 9 November 2005, W3C, 2005. URL `http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/`.

A. Kiryakov, D. Ognyanov, and D. Manov. OWLIM — A Pragmatic Semantic Repository for OWL. In M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, and Q. Sheng, editors, *Web Information Systems Engineering – WISE 2005 Workshops*, volume 3807 of *Lecture Notes in Computer Science*, pages 182–192. Springer Berlin Heidelberg, 2005.

E. Klien, D.I. Fitzner, and P. Maué. Baseline for registering and annotating geodata in a semantic web service framework. In *Proceedings of 10th International Conference on Geographic Information Science (AGILE 2007)*, 2007.

D. Koenig, A. Glover, P. King, G. Laforge, and J. Skeet. *Groovy in action.* Manning Publications Co. Greenwich, CT, USA, 2007.

M. Koster. A standard for robot exclusion. Best practice, robotstxt.org, 2010. URL `http://www.robotstxt.org/wc/robots.html`.

C. Kottman and C. Reed. Topic 5: Features. OpenGIS Abstract Specification Topic Volume OGC 08-126, Open Geospatial Consortium, Wayland, MA, USA, January 2009.

W. Kresse and K. Fadaie. *ISO Standards for Geographic Information.* Springer, Berlin, 2004.

U. Kuster and B. Konig-Ries. Towards standard test collections for the empirical evaluation of semantic web service approaches. *International Journal of Semantic Computing*, 2(3):381–402, 2008.

C. Lagoze and H. Van de Sompel. The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) version 2.0. Specification, Open Archives Initiative, 2002.

A. Langegger, W. Wöß, and M. Blöchl. A semantic web middleware for virtual data integration on the web. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic Web: Research and Applications*, volume 5021 of *Lecture Notes in Computer Science*, pages 493–507. Springer Berlin / Heidelberg, 2008.

C. Larman and V.R. Basili. Iterative and incremental developments. a brief history. *Computer*, 36 (6):47 – 56, 2003.

J. Larson, M. A. Olmos, M. Silva, E. Klien, and S. Schade. Are geospatial catalogues reaching their goals? In *AGILE Conference on Geographic Information Science*, Visegrád, Hungary, 2006.

R. R. Larson. Geographic information retrieval and spatial browsing. In L. C. Smith and M. Gluck, editors, *Geographic information systems and libraries: patrons, maps, and spatial information*, pages 81–124, 1996.

H. Lausen and J Farrell. Semantic annotations for WSDL and XML schema. W3C recommendation, W3C, August 2007. http://www.w3.org/TR/2007/REC-sawsdl-20070828/.

S. Lawrence and C.L. Giles. Searching the world wide web. *Science*, 280(5360):98, 1998.

D. Lewandowski. Mit welchen Kennzahlen lässt sich die Qualität von Suchmaschinen messen. In *Die Macht der Suchmaschinen*, pages 243–258. Halem, Köln, Germany, 2007.

D. Lewandowski. Google Scholar as a tool for discovering journal articles in library and information science. *Online Information Review*, 34(2):250–262, 2010.

J. Li, K. Furuse, and K. Yamaguchi. Focused crawling by exploiting anchor text using decision tree. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1190–1191. ACM, 2005.

W. Li, C. Yanga, and C. Yangb. An active crawler for discovering geospatial web services and their distribution pattern–a case study of ogc web map service. *International Journal of Geographical Information Science*, 24(8):1127–1147, 2010.

Y. Li, Y. Liu, L. Zhang, G. Li, B. Xie, and J. Sun. An exploratory study of web services on the internet. In *IEEE International Conference on Web Services, 2007. ICWS 2007.*, pages 380–387. IEEE, 2007.

J. Lieberman. OpenGIS Web Services Architecture. Technical Report OGC 03-025, Open Geospatial Consortium, Wayland, MA, USA, January 2003.

J. Lieberman. Geospatial Semantic Web interoperability experiment report. OpenGIS Discussion Paper OGC 06–002r1, OGC, Inc., 2006.

T. Lindholm and F. Yellin. *Java virtual machine specification.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1999.

J. Lloyd. *Foundations of Logic Programming.* Berlin: Springer-Verlag, 1987.

P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind. *Geographic information systems and science.* John Wiley & Sons Inc, Chichester, West Sussex, England, 2005.

J. M. López-Cobo, S. Losada, L. Cicurel, J. Bas, S. Bellido, and R. Benjamins. Ontology management in e-banking applications. In R. Jain, A. Sheth, M. Hepp, P. Leenheer, A. Moor, and Y. Sure, editors, *Ontology Management*, volume 7 of *Semantic Web and Beyond*, pages 229–244. Springer US, 2008.

F. J. Lopez-Pellicer, F. J. Zarazaga-Soria, A. Mogollón-Diaz, J. Nogueras-Iso, and P. R. Muro-Medrano. The gazetteer content model issue: Could spatial data infrastructures provide it? In S. I. Fabrikant and M. Wachowicz, editors, *The European Information Society: Leading the Way with Geo-information, Proceedings of the 10th AGILE Conference, Aalborg, Denmark, 8-11 May 2007*, Lecture Notes in Geoinformation and Cartography, pages 187–200. Springer Berlin Heidelberg, 2007.

F. J. Lopez-Pellicer, A. Florczyk, Javier Lacasta, F. J. Zarazaga-Soria, and P. R. Muro-Medrano. Administrative units, an ontological perspective. In I. Y. Song, M. Piattini, Y. P. Chen, S. Hartmann, F. Grandi, J. Trujillo, A. Opdahl, F. Ferri, P. Grifoni, M. Caschera, C. Rolland, C. Woo,

C. Salinesi, E. Zimányi, C. Claramunt, F. Frasincar, G. J. Houben, and P. Thiran, editors, *Advances in Conceptual Modeling – Challenges and Opportunities*, volume 5232 of *Lecture Notes in Computer Science*, pages 354–363. Springer Berlin / Heidelberg, 2008.

F. J. Lopez-Pellicer, M. Chaves, C. Rodrigues, and M. J. Silva. Geographic Ontologies Production in GREASE-II. Technical Report TR 09-18, University of Lisbon, Faculty of Sciences, LaSIGE, November 2009. URL `http://hdl.handle.net/10455/3256`.

F. J. Lopez-Pellicer, R. Béjar, A. Florczyk, P. R. Muro-Medrano, and F. J. Zarazaga-Soria. State of Play of OGC Web Services across the Web. In *INSPIRE 2010 conference, 22-25 June, Kraków, Poland*, 2010a.

F. J. Lopez-Pellicer, A. Florczyk, J. Nogueras-Iso, P. R. Muro-Medrano, and F. J. Zarazaga-Soria. Exposing CSW Catalogues as Linked Data. In W. Cartwright, G. Gartner, L. Meng, M. P. Peterson, M. Painho, M. Y. Santos, and H. Pundt, editors, *Geospatial Thinking*, Lecture Notes in Geoinformation and Cartography, pages 183–200. Springer Berlin Heidelberg, 2010b.

F. J. Lopez-Pellicer, M. J. Silva, and M. Chaves. Linkable geographic ontologies. In *GIR '10: Proceedings of the 6th Workshop on Geographic Information Retrieval*, pages 1–8, New York, NY, USA, 2010c. ACM.

F. J. Lopez-Pellicer, M. J. Silva, M. Chaves, F. J. Zarazaga-Soria, and P. R. Muro-Medrano. Geo linked data. In P. Bringas, A. Hameurlain, and G. Quirchmayr, editors, *Database and Expert Systems Applications*, volume 6261 of *Lecture Notes in Computer Science*, pages 495–502. Springer Berlin / Heidelberg, 2010d.

F. J. Lopez-Pellicer, A. J. Florczyk, R. Béjar, P. R. Muro-Medrano, and F. J. Zarazaga-Soria. Discovering geographic web services in search engines. *Online Information Review*, 35(6), 2011. Accepted for publication.

F.J. Lopez-Pellicer, R. Béjar, F.J. Zarazaga-Soria, and P.R. Muro-Medrano. Aspectos de modelos e infraestructura de servicios pare el soporte de un servicio nacional estándar de nomenclátor en web. In C. Granell, editor, *Avances en las infraestructuras de datos espaciales*, volume 26 of *Treballs d'Informàtica i Tecnologia.* Universitat Jaume I, Servei de Comunicació i Publicacions, 2006.

J. Lu and D. Li. Estimating deep web data source size by capture–recapture method. *Information Retrieval*, 13:70–95, 2010.

R. Lucchi and C. Elfers. Resource Oriented Architecture and REST: Assesment on impact and advantages on INSPIRE. Technical Report EUR 23397 EN - 2008, JRC, European Communities, 2008.

J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy. Web-scale data integration: You can only afford to pay as you go. In *Proceedings of CIDR*, pages 342–350, 2007.

J. Madhavan, D. Ko, L Kot, V. Ganapathy, A. Rasmussen, and A. Y. Halevy. Google's Deep Web crawl. *Proceedings of the VLDB Endowment*, 1(2):1241–1252, 2008.

J. Madhavan, L. Afanasiev, L. Antova, and A. Y. Halevy. Harnessing the deep web: Present and future. In *CIDR 2009, Fourth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2009, Online Proceedings*, 2009.

D. J. Maguire and P. A. Longley. The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems*, 29(1):3–14, 2005.

D. Manov, A. Kiryakov, B. Popov, K. Bontcheva, D. Maynard, and H. Cunningham. Experiments with geographic knowledge for information extraction. In *Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references, May 31 2003, Edmonton, Alberta*, pages 1–9, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

D. M. Mark. *Human Factors in Geographical Information Systems*, chapter Human Spatial Cognition, pages 51–60. Halsted Press, New York, NY, USA, 1993.

K. Markert and M. Nissim. Towards a corpus annotated for metonymies: the case of location names. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'2002), Las Palmas, Spain*, 2002.

D. Martin, M. Burstein, J. R. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. R. Payne, E. Sirin, N. Srinivasan, and K. Sycara. OWL-S: Semantic Markup for Web Services. W3C Member Submission 22 November 2004, W3C, 2004. URL `http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/`.

B. Martins, J. Borbinha, G. Pedrosa, J. Gil, and N. Freire. Geographically-aware information retrieval for collections of digitized historical maps. In *GIR '07: Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 39–42, New York, NY, USA, 2007. ACM.

J. Masó, K. Pomakis, and N. Julià. OpenGIS Web Map Tile Service Implementation Standard. Implementation Standard OGC 07-057r7, Open Geospatial Consortium, April 2010.

I. Masser. The future of spatial data infrastructures. In *ISPRS Workshop on Service and Application of Spatial Data Infrastructure, XXXVI (4/W6), Oct.14-16, Hangzhou, China*, 2005.

P. Maué. Semantic annotations in OGC standards. Discussion Paper OGC 08-167r1, Open Geospatial Consortium, July 2009.

F. McCown and M.L. Nelson. Agreeing to disagree: search engines and their public interfaces. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 309–318. ACM, 2007.

D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. W3C recommendation, W3C, February 2004. URL `http://www.w3.org/TR/2004/REC-owl-features-20040210/`.

S.A. McIlraith, T.C. Son, and H. Zeng. Semantic web services. *Intelligent Systems, IEEE*, 16(2): 46–53, 2001.

L. McKee and C. Kottman. Inside the OpenGIS Specification. White paper, Open Geospatial Consortium, Inc., Wayland, MA, USA, October 1999.

MDWG. Content standard for digital geospatial metadata (revised june 1998). Standard FGDC-STD-001-1998, Metadata Ad Hoc Working Group (MDWG), Federal Geographic Data Committee, Washington, D.C., 1998. URL `http://www.fgdc.gov/metadata/csdgm/`.

G. Meditskos and N. Bassiliades. Combining a DL Reasoner and a Rule Engine for Improving Entailment-Based OWL Reasoning. In A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 277–292. Springer Berlin Heidelberg, 2008.

F. Menczer. Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology*, 55(14):1261–1269, 2004.

F. Menczer, G. Pant, and P. Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology (TOIT)*, 4(4):378–419, 2004.

A. Mesbah, E. Bozdag, and A. van Deursen. Crawling Ajax by inferring user interface state changes. In *Eighth International Conference on Web Engineering*, pages 122–134. IEEE, 2008.

Microimages. MicroImages' WMS and ArcIMS Catalogs. Technical guide, Microimages, Inc, 2008. URL `http://www.microimages.com/documentation/TechGuides/74MIwebCatalogs.pdf`.

A. Miles and S. Bechhofer. SKOS Simple Knowledge Organization System Reference. W3C Proposed Recommendation, W3C, June 2009. URL `http://www.w3.org/TR/2009/PR-skos-reference-20090615/`.

M. Minsky. A framework for representing knowledge. Technical Report AIM-306, Massachusetts Institute of Technology, Cambridge, MA, USA, 1974.

G. Mohr, M. Stack, I. Rnitovic, D. Avery, and M. Kimpton. Introduction to Heritrix. In *4th International Web Archiving Workshop*, 2004.

A. Moshchuk, T. Bragin, S.D. Gribble, and H. M. Levy. A crawler-based study of spyware on the web. In *Proceedings of the 2006 Network and Distributed System Security Symposium*, pages 17–33. Citeseer, 2006.

B. Motik, B. Parsia, and P. F. Patel-Schneider. OWL 2 Web Ontology Language – Structural Specification and Functional-Style Syntax. W3C Recommendation, W3C, 27 October 2009a. URL `http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/`.

B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009b.

U. Müller and N. Mandery. Leistungsfähigkeit aktueller WMS-Dienste. In *INTERGEO 2009*, 2009.

M. Muñoz-Nieto, J. Finat, P. Martín-López, M. Martínez, B. Valverde, and F. Delgado. Ontology-based web services for accessing to cultural heritage environments. In *iiWAS'2010 - The 12th International Conference on Information Integration and Web-based Applications & Services, 8-10 November 2010, Paris, France*, 2010.

A. Na and M. Priest. Sensor Observation Service. Implementation Standard OGC 06-009r6, Open Geospatial Consortium, November 2007.

M. Najork and J.L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th international conference on World Wide Web*, pages 114–118. ACM, 2001.

NASA. NASA's Earth System Science Data Resources: tapping into a wealth of data, information, and services. Brochure, National Aeronautics and Space Administration, 2007. URL `http://outreach.eos.nasa.gov/outreach/broch/daac/ESSDR112007.pdf`.

NDMSO. MARC 21 Standards. Standard, Network Development and MARC Standards Office (NDMSO), U.S. Library of Congress, 1999. URL `http://www.loc.gov/marc/`.

NDMSO. MODS: Metadata Object Description Schema. Standard, Network Development and MARC Standards Office (NDMSO), U.S. Library of Congress, 2005. URL `http://www.loc.gov/standards/mods/`.

D. Nebert. Developing Spatial Data Infrastructures: The SDI Cookbook. Technical report, Global Spatial Data Infrastructure, 2004. URL `http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf`. Version 2.0.

D. Nebert, A. Whiteside, and P. Vretanos. Open GIS Catalogue Services Specification. OpenGIS Publicy Available Standard OGC-07-006r1, Open GIS Consortium Inc., February 2007. Version 2.0.2.

M. Nilsson, A. Powell, P. Johnston, and A. Naeve. Expressing Dublin Core metadata using the Resource Description Framework (RDF). Dcmi recommendation, Dublin Core Metadata Initiative, 2008. URL `http://dublincore.org/documents/dc-rdf/`.

NIMA. Department of Defense – World Geodetic System 1984: Its Definition and Relationships With Local Geodetic Systems. NIMA Technical Report TR8350.2, National Imagery and Mapping Agency (NIMA), 1997.

J. Nogueras-Iso, F. J. Zarazaga-Soria, J. Lacasta, R. Béjar, and P. R. Muro-Medrano. Metadata standard interoperability: application in the geographic information domain. *Computers, Environment and Urban Systemsis*, 28(6):611– 634, 2004.

J. Nogueras-Iso, F. J. Zarazaga-Soria, and P. R. Muro-Medrano. *Geographic Information Metadata for Spatial Data Infrastructures: Resources, Interoperability and Information Retrieval.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

J. Nogueras-Iso, F. J. Lopez-Pellicer, J. Lacasta, F. J. Zarazaga-Soria, and P. R. Muro-Medrano. Building an address gazetteer on top of an urban network ontology. In J. Teller, C. Roussey, and J. Lee, editors, *Ontologies for Urban Development*, volume 61 of *Studies in Computational Intelligence*, pages 157–167. Springer, 2007.

A. Ntoulas, P. Zerfos, and J. Cho. Downloading textual hidden web content through keyword queries. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 100–109, New York, NY, USA, 2005. ACM.

OGC. OGC Reference Model version 2.0. Technical Report OGC 08-064r4, Open GIS Consortium Inc., Wayland, MA, USA, November 2008.

OGC. OGC – Implementation statistics, November 2010. URL `http://www.opengeospatial.org/resource/products/stats`.

C. Olston and M. Najork. Web crawling. *Information Retrieval*, 4(3):175–246, 2010.

D. Orth and R. Payne. Domestic names: Principles, policies, and procedures. Policies, United States Board on Geographic Names and Domestic Geographic Names, 2003.

OWL WG. OWL 2 Web Ontology Language Document Overview. W3C Recommendation, W3C OWL Working Group (OWL WG), W3C, 27 October 2009. URL `http://www.w3.org/TR/owl2-overview/`.

G. Pant and P. Srinivasan. Learning to crawl: Comparing classification schemes. *ACM Transactions on Information Systems (TOIS)*, 23(4):430–462, 2005.

G. Pant, P. Srinivasan, and F. Menczer. Exploration versus exploitation in topic driven crawlers. In *WWW02 Workshop on Web Dynamics*, 2002.

M. P. Papazoglou. Service-Oriented Computing: Concepts, Characteristics and Directions. *Web Information Systems Engineering, International Conference on*, 0:3, 2003.

M.P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-oriented computing: State of the art and research challenges. *Computer*, 40(11):38–45, 2007.

B. Parsia and E. Sirin. Pellet: An OWL DL Reasoner. In *Third International Semantic Web Conference-Poster*. Citeseer, 2004.

A. Paschke, D. Reynolds, G. Hallmark, H. Boley, M. Kifer, and A. Polleres. RIF core dialect. Candidate recommendation, W3C, October 2009. http://www.w3.org/TR/2009/CR-rif-core-20091001/.

A. A. Patil, S.A. Oundhakar, A.P. Sheth, and K. Verma. METEOR-S web service annotation framework. In *Proceedings of the 13th international conference on World Wide Web*, pages 553–562. ACM, 2004.

C. Pautasso, O. Zimmermann, and F. Leymann. RESTful Web services vs. "big" Web services: making the right architectural decision. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 805–814, New York, NY, USA, 2008. ACM.

S. Payette and C. Lagoze. Flexible and extensible digital object and repository architecture (FEDORA). *Research and Advanced Technology for Digital Libraries*, pages 517–517, 2009.

C. Pedrinaci and J. Domingue. Toward the next wave of services: Linked services for the web of data. *J. UCS*, 16(13):1694–1719, 2010.

C. Pedrinaci, D. Liu, M. Maleshkova, D. Lambert, J. Kopecky, and J. Domingue. iServe: a linked services publishing platform. In *CEUR Workshop Proceedings*, volume 596, 2010.

G. Percivall. Topic 12: Opengis service architecture. OpenGIS Abstract Specification Topic Volume OGC 02-116, Open Geospatial Consortium, Wayland, MA, USA, January 2002.

C. Portele. OpenGIS Geography Markup Language (GML) Encoding Standard. OpenGIS Standard OGC 07-036, Open Geospatial Consortium Inc., July 2007. Version 3.2.1.

A. Powell, M. Nilsson, A. Naeve, P. Johnston, and T. Baker. DCMI Abstract Model. Dcmi recommendation, Dublin Core Metadata Initiative, 2007. Available from: http://dublincore.org/documents/abstract-model/.

V. Presutti and A. Gangemi. Identity of Resources and Entities on the Web. *International Journal on Semantic Web and Information Systems*, 4(2):49–72, 2008.

E. Pultar, M. Raubal, and M.F. Goodchild. GEDMWA: geospatial exploratory data mining web agent. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–4. ACM, 2008.

M.R. Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12(5):410–430, 1967.

S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 129–138, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

A. Rajabifard and I.P. Williamson. Spatial data infrastructures: concept, SDI hierarchy and future directions. In *Proceedings, of GEOMATICS'80 Conference, Tehran, Iran*, 2001.

C. Reed. An introduction to GeoRSS: A standards based approach for geo-enabling RSS feeds. OpenGIS White Paper OGC 06-050r3, Open GIS Consortium Inc., July 2006. Version 1.0.0.

C. Reed. A Uniform Resource Name (URN) Namespace for the Open Geospatial Consortium (OGC). RFC 5165 (Informational), April 2008. URL `http://www.ietf.org/rfc/rfc5165.txt`.

C. Reed. The OGC and REST. white paper, Open Geospatial Consortium Inc., 2009.

Refractions Research. OGC Services Survey. white paper, Refractions Research, Victoria, Canada,, October 2006.

M. Reichardt. GSDI Depends on Widespread Adoption of OGC Standards. In *From Pharaohs to Geoinformatics: FIG Working Week 2005 and GSDI-8, Cairo, Egypt April 16-21, 2005*, 2005.

G. Reynolds. GO-1 Application Objects. Implementation Standard OGC 03-064r10, Open Geospatial Consortium, May 2005.

L. Richardson and S. Ruby. *RESTful web services.* O'Reilly Media, Inc., 2007.

C. Rodrigues. An Ontology of the Physical Geography of Portugal. Master's thesis, University of Lisbon, Faculty of Sciences, June 2009.

A. F. Rodríguez-Pascual, E. López-Romero, P. Abad-Power, and A. Sánchez-Maganto. Modelo de Nomenclátor de España 1.2. Technical Report SGTMNE200507, GT IDEE, Consejo Superior Geográfico, Mayo 2006.

D. Roman and E. Klien. Swing – a semantic framework for geospatial services. In A. Scharl and K. Tochtermann, editors, *The Geospatial Web*, Advanced Information and Knowledge Processing, pages 229–234. Springer London, 2007.

D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 01 2005.

D.E. Rose and D. Levinson. Understanding user goals in Web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM, 2004.

L. Rose. Geospatial Portal Reference Architecture: A community guide to implementing standards-based geospatial portals - 0.2. OGC Discussion Paper OGC 04-039, Open Geospatial Consortium Inc., July 2004.

Y. Ru and E. Horowitz. Indexing the invisible web: a survey. *Online Information Review*, 29(3): 249–265, 2005.

M. Sabbouh, S. Jolly, D. Allen, P. Silvey, and P. Denning. Interoperability. In *W3C Web Services Workshop*, pages 11–12, 2001.

T. Sammes and B. Jenkinson. *Forensic Computing: a pratitioner's guide*. Springer, 2nd edition, 2007.

J. T. Sample, R. Ladner, L. Shulman, E. Ioup, F. Petry, E. Warner, K. Shaw, and F. P. McCreedy. Enhancing the US Navy's GIDB Portal with Web Services. *Internet Computing, IEEE*, 10(5): 53–60, Sept.-Oct. 2006.

J.T. Sample, L.I. Shulman, and F.P. Mccreedy. System and method for automated discovery, binding, and integration of non-registered geospatial web services, May 24 2007. US Patent App. 11/753,389.

L. Sauermann and R. Cyganiak. Cool URIs for the Semantic Web. W3C note, W3C, March 2008. URL `http://www.w3.org/TR/2008/NOTE-cooluris-20080331/`.

S. Schade, C. Granell, and L. Diaz. Augmenting SDI with Linked Data. In *Workshop On Linked Spatiotemporal Data 2010 in conjunction with the 6th International Conference on Geographic Information Science (GIScience 2010) Zurich, 14-17th September, 2010*, 2010.

P. Schut. OpenGIS Web Processing Service. Standard OGC 05-007r7, Open Geospatial Consortium, June 2007.

P. Schut, X. Geng, C. Higgins, F. J. Lopez-Pellicer, B. Low, J. Masó, A. Turner, and C. West. OpenGIS Georeferenced Table Joining Service (TJS). Implementation Standard OGC 10-070r2, Open Geospatial Consortium, November 2010.

A. Schutzberg. Skylab Mobilesystems crawls the Web for Web Map Services. *OGC User*, 4(1), August 2006. URL `http://ogcuser.opengeospatial.org/node/7`.

A. Seaborne and E. Prud'hommeaux. SPARQL Query Language for RDF. W3C recommendation, W3C, January 2008. URL `http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/`.

F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

N. Shadbolt, W. Hall, and T. Berners-Lee. The Semantic Web Revisited. *Intelligent Systems, IEEE*, 21(3):96 –101, 2006.

Y. Shang and L. Li. Precision evaluation of search engines. *World Wide Web*, 5:159–173, 2002.

S. Shankland. Google mapping spec now an industry standard, April 2008. URL `http://news.cnet.com/8301-10784_3-9917421-7.html`.

C. Sherman and G. Price. *The invisible Web: Uncovering information sources search engines can't see.* Information Today, Inc., 2001.

M. J. Silva. Searching and archiving the web with Tumba. In *CAPSI 2003-4a. Conferência da Associaçao Portuguesa de Sistemas de Informaçao*, 2003.

M. J. Silva, B. Martins, M. Chaves, N. Cardoso, and A. P. Afonso. Adding Geographic Scopes to Web Resources. *Computers, Environment and Urban Systems*, 30(4):378–399, July 2006.

G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A metadata catalog service for data intensive applications. In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 33. IEEE Computer Society, 2003.

Skylab. OGC WMS Server List, 2005. URL `http://www.skylab-mobilesystems.com/en/wms_serverlist.html`.

R.M. Smullyan. *First-order logic.* Dover Publications, 1995.

S. Staab, R. Studer, H.P. Schnurr, and Y. Sure. Knowledge processes and ontologies. *Intelligent Systems, IEEE*, 16(1):26–34, 2001.

S. Steiniger and E. Bocher. An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science*, 23(10):1345 – 1370, 2009.

N. Steinmetz, H. Lausen, and M. Kammerlander. Crawling research report. Service-Finder project Deliverable 2.1, seekda OG, Innsbruck, Austria, 2008.

N. Steinmetz, H. Lausen, and M. Brunner. Web service search on large scale. In L. Baresi, C. H. Chi, and J. Suzuki, editors, *Service-Oriented Computing*, volume 5900 of *Lecture Notes in Computer Science*, pages 437–444. Springer Berlin / Heidelberg, 2009.

R.D. Stevens, A.J. Robinson, and C.A. Goble. myGrid: personalised bioinformatics on the information grid. *Bioinformatics*, 19(Suppl 1):i302, 2003.

Mari Suárez-Figueroa, A. Gómez-Pérez, and B. Villazón-Terrazas. How to write and use the ontology requirements specification document. In R. Meersman, T. Dillon, and P. Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2009*, volume 5871 of *Lecture Notes in Computer Science*, pages 966–982. Springer Berlin Heidelberg, 2009.

A. Tamayo, C. Granell, and J. Huerta. On the complexity of OGC Web Services' XML Schemas. Technical Report DLSI-01/06/2010, Centro de Visualización Interactiva, Universitat Jaume I, 2010.

TC 211. ISO/TS 19136: Geographic information – Geography Markup Language (GML). Standard, TC 211 – Geographic Information/Geomatics, International Organization for Standardization, Geneva, Switzerland, 2007a.

TC 211. ISO/TS 19139: Geographic information – Metadata – XML schema implementation. Standard, TC 211 – Geographic Information/Geomatics, International Organization for Standardization, Geneva, Switzerland, 2007b.

TC 37/SC 2. ISO 639-3:2007 – Codes for the representation of names of languages – Part 3: Alpha-3 code for comprehensive coverage of languages. ISO International Standard, International Organization for Standardization (ISO), 2007.

TC 46/SC 4. ISO 28500:2009 Information and documentation – WARC file format. ISO International Standard, International Organization for Standardization (ISO), 2009.

M. Thelwall. Evidence for the existence of geographic trends in university web site interlinking. *Journal of Documentation*, 58(5):563–574, 2002.

W.R. Tobler. A computer movie simulating urban growth in the Detroit region. *Economic geography*, 46:234–240, 1970.

G. Tummarello, E. Oren, and R. Delbru. Sindice.com: Weaving the open linked data. In K. Aberer, K. S. Choi, N. Noy, D. Allemang, K. I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, G. Schreiber, and P. Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, volume 4825 of *LNCS*, pages 547–560, Berlin, Heidelberg, November 2007. Springer Verlag.

A. Turner. *Introduction to neogeography.* O'Reilly Media, Inc., 2006.

UDDI/TC. UDDI Version 3.0.2. Technical report, OASIS UDDI Specification Technical Committee (UDDI/TC), 2004.

UNCHS. Nairobi spatial information for sustainable development, 2 – 5 october 2001, recommendations. Technical report, Nairobi Conference on Spatial Information for Sustainable Development secretariat, UNCHS/FIG/ISK, October 2001.

UNGEGN. Manual for the national standardization of geographical names. Manual ST/ESA/STAT/SER.M/88, United Nations Group of Experts on Geographical Names (UNGEGN) ,Department of Economic and Social Affairs, Statistics Division, United Nations, New York, NY, USA, 2006.

UNGIWG. Strategy for developing a United Nations Spatial Data Infrastructure in support of human response, economic development, environmental protection, peace and safety. Technical report, United Nations Geopatial Information Working Group (UNGIWG), feb 2007. URL `http://www.ungiwg.org/docs/unsdi/UNSDI_Strategy_Implementation_Paper.pdf`.

M. Uschold. Building ontologies: Towards a unified methodology. In I. Watson, editor, *16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, Cambridge, United Kingdom, 1996.

B. Vatant and M. Wick. GeoNames Ontology. Available from: http://www.geonames.org/ontology/, 2007.

L. M. Vilches-Blázquez, B. Villazón-Terrazas, V. Saquicela, A. de León, Oscar Corcho, and A. Gómez-Pérez. GeoLinked data and INSPIRE through an application case. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 446–449, New York, NY, USA, 2010. ACM.

S. Vinoski. Serendipitous reuse. *Internet Computing, IEEE*, 12(1):84–87, 2008.

T. Vitvar, J. Kopecky, M. Zaremba, and D. Fensel. WSMO-lite: Lightweight semantic descriptions for services on the web. In *Fifth European Conference on Web Services, 2007 (ECOWS'07)*, pages 77–86. IEEE, 2007.

U. Voges and K. Senkler. Open GIS Catalogue Services Specification 2.0.2: ISO Metadata Application profile. OpenGIS Publicy Available Standard OGC 07-045, Open GIS Consortium Inc., February 2007. Version 2.0.2.

J. Volz, C. Bizer, , M. Gaedke, and G. Kobilarov. Silk – a link discovery framework for the web of data. In *18th International World Wide Web Conference*, April 2009. URL `http://www2009.eprints.org/227/`.

L. von Ahn, M. Blum, N. Hopper, and J. Langford. Captcha: Using hard ai problems for security. In E. Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 646–646. Springer Berlin / Heidelberg, 2003.

P. Vretanos. Web Feature Service Implementation Specification. Implementation Standard OGC 02-058, Open Geospatial Consortium, May 2002.

P. Vretanos. OpenGIS Filter Encoding Implementation Specification. OpenGIS Publicly Available Implementation Specification OGC 04-095, Open Geospatial Consortium Inc., May 2005a. Version 1.1.0.

P. Vretanos. Web Feature Service Implementation Specification - Version 1.1.0. Implementation Standard OGC 04-094, Open Geospatial Consortium, May 2005b.

P. Vretanos. OpenGIS Web Feature Service 2.0 Interface Standard. Implementation Standard OGC 09-025r1 and ISO/DIS 19142, Open Geospatial Consortium, November 2010.

Q. Wang and J. Wang. Intelligent web map service aggregation. In *Computational Intelligence and Natural Computing, 2009. CINC'09. International Conference on*, volume 2, pages 229–231. IEEE, 2009.

A. Waxman. ArcIMS 3.0 – an application developer's perspective, 2000. URL `http://spatialnews.geocomm.com/newsletter/2000/22/arcims.html`.

C. Welty, D. L. McGuinness, and M. K. Smith. OWL web ontology language guide. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

WG ISS. Interoperability handbook. Technical Report Issue 1.1, Working Group on Information Systems and Services (WG ISS), Committee on Earth Observation Satellites, February 2008.

A. Whiteside. OpenGIS Web services architecture description. OpenGIS Best Practices Paper OGC 05-042r2, Open GIS Consortium Inc., February 2005. Version 2.0.2.

A. Whiteside. OGC Web Services common specification. Discussion Paper OGC 07-095r2, Open Geospatial Consortium, Wayland, MA, USA, September 2007.

A. Whiteside and J. D. Evans. Web Coverage Service (WCS) Implementation Specification - Version 1.1.0. Interface Standard OGC 06-083r8, Open Geospatial Consortium, October 2006.

T. Wilson. OGC KML. OGC Standard OGC 07-147r2, Open GIS Consortium Inc., April 2008. Version 2.2.0.

WS/MMI-DC. CWA 14857:2003 – Mapping between Dublin Core and ISO 19115, Geographic Information – Metadata. CEN Workshop Agreement 14857, CEN/ISSS Workshop on Metadata for Multimedia Information - Dublin Core (WS/MMI-DC), November 2003.

P. Wu, J.R. Wen, H. Liu, and W.Y. Ma. Query selection techniques for efficient crawling of structured web sources. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, page 47. IEEE, 2006.

Q. Xu and W. Zuo. First-order focused crawling. In *Proceedings of the 16th international conference on World Wide Web*, pages 1159–1160. ACM, 2007.

C.A. Yaguinuma, G. F. Afonso, V. Ferraz, S. Borges, and M.T.P. Santos. A fuzzy ontology-based semantic data integration system. In *Information Reuse and Integration (IRI), 2010 IEEE International Conference on*, pages 207 –212, 2010.

C. Yang, R. Raskin, M. F. Goodchild, and M. Gahegan. Geospatial Cyberinfrastructure: Past, present and future. *Computers, Environment and Urban Systems*, 34(4):264 – 277, 2010.

P. Yang, J. Evans, M. Cole, N. Alameh, S. Marley, and M. Bambacus. The emerging concepts and applications of the spatial Web portal. *Photogrammetric Engineering and Remote Sensing*, 73(6): 691, 2007.

P. Yendluri, M. Hondo, A. S. Vedamuthu, F. Hirsch, D. Orchard, Ü. Yalçinalp, and T. Boubez. Web Services Policy 1.5 – Framework. W3C Recommendation, W3C, September 2007. URL `http://www.w3.org/TR/2007/REC-ws-policy-20070904`.

P. Yue, L. Di, W. Yang, G. Yu, and P. Zhao. Semantics-based automatic composition of geospatial Web service chains. *Computers & Geosciences*, 33(5):649 – 665, 2007.

P. Zhao, L. Di, G. Yu, P. Yue, Y. Wei, and W. Yang. Semantic Web-based geospatial knowledge transformation. *Computers & Geosciences*, 35(4):798–808, 2009.

Y. Zhou, E. Reid, J. Qin, H. Chen, and G. Lai. US domestic extremist groups on the Web: link and content analysis. *IEEE intelligent systems*, 20(5):44–51, 2005.