Ontology Based Portals for Accessing Geospatial Data and Web Services

Aster Denekew Yilma
March, 2006
Ontology Based Portals for Accessing Geospatial Data and Web Services

by

Aster Denekew Yilma

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation, Specialisation: (fill in the name of the specialisation)

Thesis Assessment Board

Chairman Prof. Dr. M.J. Kraak
First Supervisor Ir. Rob Lemmens
Second Supervisor Dr. Andreas Wytzisk
External Examiner Drs. Marian de Vries

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION
ENSCHENDE, THE NETHERLANDS
Disclaimer

This document describes work undertaken as part of a programme of study at the International Institute for Geo-information Science and Earth Observation. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the institute.
Abstract

This research has dealt with one of the geospatial data sharing problems namely, semantic interoperability problem. The motivation for the research is the current initiatives for data sharing such as Spatial Data Infrastructure and Environmental Information Network that are being implemented at national, regional and global levels which require open and interoperable systems.

Geo-information technology is rapidly changing from static and proprietary mapping systems into dynamic, open and interoperable distributed services which allow sharing of geospatial information and other resources over the web. In such open environments, discovering, accessing and integrating information are difficult due to semantic heterogeneity arising from the terminologies and conceptualizations between the different information providers and users. Sharing of geospatial data requires agreement on meanings of concepts, and other standards between information providers and users to have a common understanding of the semantics of the information, in the specific domain.

The current state of the art technology for sharing geospatial information is geospatial web services. Effective use of geo-information requires easy access to metadata that describes the geospatial data and services. Users need metadata in order to understand and choose suitable data and services. The metadata for services are intended to provide sufficient information to allow a client to invoke the services. We have specifically addressed OGC Web Services here. OGC Web Services are self describing and support capabilities representation. The service metadata (capabilities document) returned from the request by the operation called GetCapabilities, specifies the content and supported capabilities of the service. However, current descriptions of services, service metadata, do not provide the semantics or meaning of the data content and therefore are not sufficient to allow semantic interoperability. Overcoming semantic interoperability problem is an issue that requires a solution.

To address this problem, the research deals with developing an ontology, entailing a common vocabulary for concepts in the domain application and the relationships that exist between concepts. This is followed by semantic description of the geospatial web services based on the concepts developed in the ontologies to give users meaningful information. To achieve this, we have chosen an environmental domain application for which we developed an ontology. OGC Web Map Services are implemented for environmental themes. The ontology is then used for semantic annotation in the service metadata to attach meanings for their data content. By enriching the services semantically users will have better understanding of the
information content and the capabilities of the services. Finally, the services along with their service descriptions as well as the ontology are made available for the user in the geospatial portal. A user can find a certain concept based on the domain ontology and have understanding how its meaning is restricted to the application.

Keywords: Interoperability, Knowledge representation, Ontology, Semantics, Semantic Web, Description Logic, Web Services, Geospatial portal
I would like to take this opportunity to extend my gratitude to those individuals who were involved in this research, which would not be possible without their support and ideas. I would like to thank Ir. Rob Lemmens, my first supervisor, for his ideas, assistance and comments throughout the research period. I would also like to extend my sincere gratitude to Dr. Andreas Wytzisk, my second supervisor, for his critical and constructive comments that shapes the thesis.

My appreciation and thanks also goes to Drs. Barend J. Köbben who have been very keen to help, giving technical support in setting up the geospatial server and fixing related problems. I would also like to thank Mr. Ard Blenke, cluster manager, for his continuous help at any time his assistance is needed.

Finally, I would like to thank all ITC staff who have contributed directly or indirectly in the knowledge gained throughout the study period and the research work. And thanks to my GFM2 colleagues whose cooperation, friendship and company that makes my stay comfortable.
3.4 Working Knowledge Base

3.5 Knowledge Representation Languages

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.1</td>
<td>XML, XML Schema and URI</td>
<td>39</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Resource Description Framework (RDF) and RDF-S</td>
<td>41</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Web Ontology Language (OWL)</td>
<td>45</td>
</tr>
<tr>
<td>3.5.2.1</td>
<td>Components of an OWL Ontology</td>
<td>45</td>
</tr>
<tr>
<td>3.5.2.2</td>
<td>Subsumption, Disjointness and Equivalence</td>
<td>46</td>
</tr>
<tr>
<td>3.5.2.3</td>
<td>Properties and Restrictions</td>
<td>48</td>
</tr>
<tr>
<td>3.5.2.4</td>
<td>Namespaces and Namespace Prefixes</td>
<td>52</td>
</tr>
</tbody>
</table>

3.6 Concluding Remarks

4 Implementation

4.1 Use case

4.2 Development of Ontology

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1</td>
<td>Taxonomies</td>
<td>56</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Datasets Used</td>
<td>57</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Tools Used</td>
<td>57</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Concepts of the ontology</td>
<td>58</td>
</tr>
</tbody>
</table>

4.3 Implementing OGC Web Map Services (WMS)

4.4 Description of Web Services

4.5 Development of Geospatial Portal

4.6 Evaluation

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6.1</td>
<td>Evaluation of the ontology</td>
<td>73</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Evaluation of Web Services Implementation and Description</td>
<td>74</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Evaluation of the Geo portal</td>
<td>77</td>
</tr>
</tbody>
</table>

4.7 Concluding Remarks

5 Conclusion and Recommendation

5.1 Conclusion

5.2 Recommendation and Future Work

Bibliography

Appendix A. Use Case

Appendix B. Ontology Development

Appendix C. Web Services Implementation

Appendix D. Web Services Description

Appendix E. Geo portal
List of figures

Figure 1-1 : The Geospatial Portal Reference Architecture ________________________________ 7
Figure 2-1: The three conceptual roles and operations of a Service Oriented Architecture ______ 18
Figure 2-2 : The SOA conceptual architecture with SOAP, WSDL, and UDDI __________________ 20
Figure 3-1: Subsumption (Subclassof) relationship for Flood concept in the Environmental_Domain _______ 47
Figure 3-2: Class Hierarchy of Services showing disjointness ______________________________ 48
Figure 4-1: Sequence Diagram to access geospatial services through Geospatial portal ________ 56
Figure 4-2: Environmental Domain concepts hierarchy _________________________________ 59
Figure 4-3: Service Taxonomy in the Protege ontology editor window ______________________ 60
Figure 4-4: Instances of Map_Access_Service shown in the instance editor window of Protégé __ 61
Figure 4-5: Instances of Map_Access_Service with their properties and corresponding values ___ 62
Figure 4-6: The Service information page for Africa_ProtectedAreas_Web_Map_Service ________ 71
Figure 4-7: Africa_ProtectedAreas_Web_Map_Service displayed in HTML Viewer___________ 72
Figure 4-8: Ontology Browser window generated from Protégé used to refer concepts _________ 73
Figure 4-9: Gaia Client Viewer displaying Africa_Geology_Web_Map_Service________________ 75
Figure 4-10: The Web Map Service integrated with other services in the OGC WMS Catalog Viewer ________________________________ 76
Figure 4-11: The Web Map Services integrated in ArcMap _______________________________ 77
Figure A-1::Top Level Use case diagram for Environmental Information Network ____________ 89
Figure A-2 : Top Level Use case diagram for Environmental Information Network __________ 89
Figure A-3 : Top Level Use case diagram for Environmental Information Network __________ 90
Figure A-4 : Sublevel Use case diagram– Publish Data/Services___________________________ 92
Figure A-5: Sublevel Use case diagram– Search Data/Services ____________________________ 92
Figure A-6: Sublevel Use case diagram– Retrieve Data/Services __________________________ 93
Figure A-7: Example Use Case Diagram – Search and Retrieve Data/Services _______________ 95
Figure B-1: Concepts of the Environmental Domain ___________________________________ 100
Figure B-2: Graphical representation of part of the environmental domain ontology _________ 101
Figure B-3: Service Ontology ______________________________________________________ 102
Figure B-4: The FeatureType concept and its subconcepts (Application Ontology) _____________ 103
Figure C-1 : Africa_Climate_Web_Map_Service ______________________________________ 112
Figure C-2 :Africa_Geology_Web_Map_Service in the Gaia client viewer __________________ 113
Figure E-1 : Portal Home Page _______________________________________________________ 126
Figure E-2 : Service page in the portal _____________________________________________ 127
Figure E-3 : Service access page in the portal ________________________________________ 128
Figure E-4 : Ontology Browser in the Portal __________________________________________ 129
List of tables

Table A-1: Top level use cases ................................................................. 90
Table A-2: Sub level use cases................................................................. 90
Table A-3: Types of Actors ................................................................. 91
Table A-4: The functional roles of Providers and Users.............................. 97
1 About the Research

1.1 Background

One of the major challenges in the millennium is how to manage the world’s environment and its resources. As human population grows the pressure on resources also grows with it. Human activities have also increased impacts on climate change, which become major development challenges for most developing countries. As these problems persist, governments, industry and conservation try to plan and manage resources in a sustainable way [1]. These sustainable development efforts require complex decision making which involves the consideration of environmental, social and economic effects on the use of resources. Such decision making in turn requires ready access by decision makers and stakeholders to up-to-date and reliable information.

In recent years, geo-information has become very important as the means for communicating and describing real world situations. Consequently, vast databases have been created and are being handled by many organizations to provide an inventory of natural resources and to support planning and decision making.

One of the problems in the development of geographic information systems (GIS) is the volume of data that needs to be gathered. Currently, large and valuable datasets are used in some organizations. Others with out knowing the existence of the datasets in these organizations, or with lack of access and data sharing mechanisms duplicate the effort to create the datasets already exist. In addition, with the increasing data acquisition in modern remote sensing and GPS based technologies, the size of GIS databases become more difficult to maintain.

During the last decade, technological developments have facilitated access to geo-information and have made it easier to manipulate, reducing the effort and skills required to exploit it effectively. As a result, the use of geo-information is expanding beyond the traditional users (government organizations), to include new user communities such as industries, and the private sector. As there are more users that require geo-information for different applications and their needs are becoming service oriented, developers and data
providers have to deal with a great diversity of user needs to satisfy, which is a challenging task [2].

In recent years, the problems related with geo-spatial data/information such as duplication of efforts, non-standardized approaches, lack of awareness have been analyzed and understood by most organizations. Geo-information providers have realized that satisfying today’s geo-information needs goes beyond the capacity of ‘single’ organizations. Therefore, these organizations were seeking for mechanisms that enable them to work together to share their data and resources and to provide efficient information services to the users [2]. This brought the emergence of different initiatives. At this time, a number of initiatives are under way to facilitate the sharing of information. The two major initiatives that are being implemented at different level are Spatial Data Infrastructure and Environment Information Network. These are explained in the following section.

1.1.1 Spatial Data Infrastructure (SDI)

Spatial Data Infrastructure (SDI) evolved from the early nineties, where the proposed development of national geospatial data infrastructure in Canada has received acceptance from the public, private sector and the user community. The first international conference on “Emerging Global Geospatial Data Infrastructure” which was held in Bonn in September 1996 brought together diverse communities to share ideas in building a Global geospatial data infrastructure [3]. And subsequent conferences, gave rise to awareness about SDI for a larger information community.

SDI has been defined by different bodies and researchers at different times. The US Federal Geographic Data Committee (FGDC) defines SDI as the technologies, policies, and people necessary to promote sharing of geospatial data throughout all levels of government, the private and non-profit sectors, and the academic community [4].

As [3] defines it Geospatial Data Infrastructure encompasses the networked geospatial databases and data handling facilities, the complex of institutional, organizational, technological, human, and economic resources which interacts with one another and underpin the design, implementation, and maintenance of mechanisms facilitating the sharing, access to, and responsible use of geospatial data at an affordable cost for a specific application domain or enterprise.

SDI have become very important in determining the way in which spatial data are used through out an organization, a state or province, a nation, different regions and the world. As
[5] points out current investigations indicated SDI is multileveled in nature, formed from a hierarchy of inter-connected SDI at corporate, local, state or provincial, national, regional and global levels.

As the information technology is changing fast, the SDI concept is also evolving and adapting to the new technology to meet the changing needs. Originally, the focus of SDI was on the type, development of and access to spatial datasets. As the concept evolved it has expanded to include a focus on people, access, policies and standards in relation to the data as well as adopting a shift in emphasis from “product based” to “process based” approach [5].

Hence, the above type of definitions are static and conceal the evolutionary nature of SDI, [6] argued, and proposed a more comprehensive understanding of SDI, by emphasizing the role of the user and the evolutionary nature of SDI, as:

“multi-levelled, scalable, and adaptable collection of technical and human services, which are interconnected across system, organizational, and administrative boundaries via standardized interfaces. Those services enable users from different application domains to participate in value chains by gaining a seamless access to spatial information and geo-processing resources.”

As [5] emphasized, users are now the dominant drivers in the establishment of spatial databases and the underpinning SDI. Without a use or business applications an infrastructure such as SDI has no justification for its existence. Currently, SDI is being implemented at different levels to facilitate the discovery and access to geospatial data and services to the users.

1.1.2 Environment Information Network

African Environment Information Network initiative (AEIN) is a regional initiative to exchange environmental information with in the nation and across nations in the continent. The idea for this initiative originated from the African Ministerial Conference on the Environment (AMCEN) which proposes that getting information about the conditions of environmental assets in Africa is crucial in monitoring the continent’s environment. AMCEN believed that there is an urgent need to develop an integrated environmental assessment and reporting mechanism within the continent, which will generate and provide relevant environmental information to facilitate efficient monitoring of the environment.
Hence, the AEIN initiative was mandated by AMCEN, which in July 2002 requested UNEP’s assistance to implement it to “build capacity for integrated environmental assessments and reporting” in Africa. Subsequently, it became necessary to establish the National Working Groups to implement the African Environment Information Network initiative at a national level. The implementation of the Africa Environment Information Network (AEIN) is now underway with 13 countries fully involved in the pilot phase and the whole region participating in capacity building activities. The goal of AEIN, which is being coordinated by the UNEP Division of Early Warning and Assessment (DEWA), is to enhance accessibility to more reliable environmental data and information at national level for the environmental assessment and reporting in the region [7].

The geospatial component of environmental information will be significant for environmental monitoring. The implementation of this initiative at national level will also be an input for the national Spatial Data Infrastructure (NSDI).

1.1.3 Distributed Geographic Information (GI) Systems

With the growth of Internet, World Wide Web, and GIS the applications of distributed geographic information systems has resulted in a significant development. Distributed geographic information is the wide spread distribution of geographic information in a variety of forms, including maps, images, datasets, analysis operations and reports [8]. This integration of Internet and GIS opens the research areas referred to as Web based GIS, Internet GIS, Online GIS, and Distributed GI services.

Distributed GIS represents a broader framework including both Internet GIS and Mobile GIS. Distributed GIS allows a variety of client devices such as desktop computers, laptop computers, PDAs or cellular phones, to access geospatial data and processing tools in servers anywhere and at any time. Internet GIS is the framework of network based GIS that utilizes the Internet to access remote geographic information and geoprocessing tools. It has evolved from displaying static maps to interactive maps and expanded to distributed GI services which offers the capabilities to interact with multiple and heterogeneous systems and servers that support more advanced GIS functions [9]. Many enterprises’ GI systems evolve from bottom up approach to build inter-organizational enterprise GIS. At the same time there has been a move to initiate top-down programs that establish infrastructure for all GI users. These spatial data infrastructure (SDI) programs are increasingly viewed as the route to build capacity and encourage GIS participation [10].

The GIS industry has been developing several software packages to provide distributed and on-line mapping functions, such as ArcIMS from ESRI, GeoMedia Web Map from
INTERGRAPH, MapXtreme from MapInfo, and MapGuide from Autodesk. However, like any other desktop GIS software the problems associated with these web mapping softwares are that they have their own proprietary interface designs, data encoding schemes that limit interoperability. From time to time, these vendors update their interfaces, forcing client systems to change and forcing users to upgrade to the new system.

1.1.4 Geospatial Web Services

With the rapid change of the technology, the users’ need also changes requiring almost immediate solutions in which they can define and assemble GIS components [11]. The open and distributed GI web services environment makes it possible to acquire, process and analyze geographic information without the need for installing proprietary GIS software and having GIS expert knowledge [12]. Open systems are considered to be systems that interoperate through open interfaces. An interface is simply a common boundary, a means to make a connection between two software components [13].

A Web Service is defined by [14] as:

“A Web service is any piece of software that makes itself available over the Internet and uses a standardized XML¹ messaging system”.

Geospatial Web Services are web services with geospatial content. Geospatial Web Services provide access to spatial data and GIS functionality through the internet. This makes it possible for users to integrate them with their own systems and applications without the need to develop or host specific GIS tools and data sets by themselves. Geospatial web services make it feasible for multiple organizations that need to access the same data to do so from a server hosting it as a web service, rather than simply duplicating the data in each organization [15].

Currently, vast ranges of geo-information services already exist on the World Wide Web. However, some of these GI services are independent collections of services. Recently researchers are engaged in development of mechanisms to describe, combine and manage these independent collections of services, making them available as independently developed, yet interoperable services.

---

¹ Extensible Mark-up Language
1.1.5 Interoperability

Sharing of spatial data requires agreement among participants on definitions, concepts, and other standards such as data formats and database structures to achieve its objective: “sharing of the large expenses to keep the geographic database current and to reduce duplication of efforts of organizations for database maintenance” [16].

The ability to exchange information at runtime is known as interoperability. The problems that occur when data are exchanged among different users are multiple. Interoperability problems can be divided into syntactic and semantic heterogeneities. In order to achieve interoperability between GIS systems, the data from one system must be integrated into another system. [17], [18]. Two systems are said to be:

*Syntactically interoperable* – if they use the same structure for the information that flows in the systems and is processed by the systems.

*Semantically interoperable* – if they have a common understanding of the semantics of the information that flows in the systems and is processed by the systems [19].

Standardization is one way to overcome interoperability deficiencies between GIS systems [17]. Effective use of GI requires easy access to documentation that describes its properties such as quality, lineage, ownership, and fitness for use. This description is referred to as metadata, data about the data. Therefore, the GIS community needs to develop some sort of standardized system for efficiently sharing and accessing geospatial data and services. The critical elements include a standard for metadata that describes the data source, and a set of interfaces such as geospatial data portals that index data sources [9] and provide access to their metadata. Open Geospatial Consortium Inc. (OGC) has been working in standards and specifications for interfaces, schemas and encodings for open and interoperable geospatial information [20] to address these issues.

1.1.6 Geospatial portals

Portals are web sites that give users organized access, to a collection of information resources, including data sets, services, processing tools and an organized collection of links to many other sites, usually through catalogue services. Geospatial portals are gateways to a collection of online geospatial information resources. Geospatial portals provide capabilities to query metadata records for relevant data and services, and then link directly to the on-line content, services themselves [10, 13]. Therefore, Geospatial portals can be thought of as the “hubs” or “geo-information resource supermarkets” in the Spatial Web. Data providers
register their data and services in catalogue services for access via the portal [13]. Hence, Geospatial portals are a key element of SDI [10].

OGC has put forward an architectural framework that can be used as a guide to the implementation of an operational portal that provides access to geospatial content and their metadata. The Geospatial Portal Reference Architecture is shown in Figure 1-1.

![Figure 1-1: The Geospatial Portal Reference Architecture](Source: [21])

Some examples of geospatial portals are: INSPIRE (Infrastructure for Spatial Information in Europe) – a European initiative aimed at making geospatial content more readily available and useful for sustainable development and increased protection of the environment, currently leading to European SDI (http://eu-geoportal.jrc.it/), Geoconnections – portal for accessing the Canadian Geospatial Data Infrastructure (www.geoconnection.org), Geospatial One Stop – United States National Spatial Data Infrastructure (www.geodata.gov).

1.1.7 Related Works

Over the last few years a great deal of work has been done in standardization for metadata documentation in general and geographic information in particular. Two major organizations that set industry standards for distributed GI Services are ISO/TC 211, the Technical Committee tasked by the ISO (International Standards Organization) and OGC.
ISO/TC 211 is responsible for standardization in the field of digital geographic information amongst others. The international standards for metadata and services, ISO 19115: metadata and ISO 19119: services respectively, are the standardization efforts of this committee.

OGC is a non-profit, international, voluntary consensus standards organization that is involved in the development of specifications for geospatial and location-based services. The main goals of OGC are the full integration of geospatial data and geoprocessing resources into mainstream computing and the widespread use of interoperable geoprocessing software and geodata products throughout the information infrastructure. OGC has been working on specifications for open and interoperable geospatial information [20].

Metadata is a prerequisite for reusing geographic datasets. In a heterogeneous environment, metadata are capable of locating, evaluating, extracting, required information by providing guidance to find the most appropriate dataset for the specific application [22]. Many data access services are associated with descriptions of the available datasets using their metadata. The metadata elements for a service in ISO 19119 are also intended to provide sufficient information to allow a client to invoke the services based on the metadata content [23]. Search for information and retrieval, is usually done using some search engine that accesses a metadata catalogue of the available resources. If multiple information resources are available with no prior knowledge about its content, structure, and semantics, the search may not be successful.

Besides the development of metadata standards, different keywords and thesauri have been developed by different researchers. Ontologies have also been developed and used in different projects to structure and access information in different disciplines. Ontologies are considered to be logical and an adequate methodology to support interoperable distributed systems. They can provide a “common basis” for semantic mapping between information communities [11, 18]. Typically, ontology contains a hierarchical description of important concepts in a domain, and describes crucial properties of each concept.

Currently, there are content standards, standard classification schemes, and keywords, which give taxonomy in the domain applications. Taxonomy is a subject classification used to classify resources [24]. Global Change Master Directory (GCMD) has been developed and maintained by NASA (National Aeronautic Space Administration) to meet the needs of the Earth science community. NASA’s directory for Earth Science data and services, the GCMD, includes approximately 1000 controlled Earth science keywords. These keywords are organized in a hierarchy constructed of TOPICs, TERMS, and VARIABLEs. CORINE (Coordination of Information on the Environment) is also a programme which was initiated
in the European Union and taken over by European Environment Agency (EEA) for an inventory of land cover in 44 classes for most areas of Europe. This classification scheme for land cover is used as standard for inventories, classification of satellite images and cartographic products in the European countries. These classifications and keywords can be used as a framework for development of ontology in the specific domain to ensure consistency among data sets and allowing the sharing of data and integrate multiple sources of information.

1.2 Motivation

The motivation for this research arises from the current initiatives for data sharing such as Spatial Data Infrastructure and Environmental Information Network that are being implemented at national and regional levels which require open and interoperable systems. One of the problems that occur when data are exchanged is semantic interoperability problem. Particularly, solving problems in an environmental domain involves various data from different areas. Usually, all the data are not available from one database but are distributed and have different formats. As environmental science has an interdisciplinary and multi-sectoral nature, environmental information often faces semantic problems. They arise mainly from the use of different terminologies and concepts in different disciplines and sectors. For example, “wetland” can be categorized by Ministry of Water Resources as Surface water, and Environment Agency may categorize it as aquatic habitat or more general concept Biosphere, while these same concepts might refer to the same thing. There must be an agreement on the concepts and definition of terms between the participants in the information sharing. Solving these semantic problems would facilitate information sharing with different disciplines and cross sectors in the environmental domain using Environmental Information Networks. As this information network will share the same or similar infrastructure as the Spatial Data Infrastructures (SDI), this solution could also be applied to SDI at large.

1.3 Problem Definition

Problem of efficient access to geospatial information

Most geo-information providers are willing to share their data and services. But, how do they share it? In principle, they could distribute them online [1]. The question is how will the user find it? As a user one way to find data is to use search engines such as catalogue services [9]. The problem with this kind of search is, too many results are returned which may not be relevant, if the user does not enter the correct search terms. The cause is that the search tools are keyword based and they are unable to extract meaning from terms on the web pages or the
available resources [24]. The users then do not know which one is the best possible link to use. Information providers therefore, need to ensure that their data will be retrieved by a simple, straightforward search.

**Lack of semantic interoperability**

OGC has been working in standards for open and interoperable geospatial information [20]. In this effort, OGC developed a Web mapping approach, called OGC Web Services (OWS), based on open interfaces, encodings and schemas. OGC web Services are self describing and supports capabilities representation by the operation called GetCapabilities, which specifies the content and supported capabilities of the service.

One of the OWS implementations is Web Map Services (WMS). WMS produces maps of spatially referenced data dynamically from geographic information. WMS supports a GetCapabilities operation, to make a request for the service. The GetCapabilities request returns service metadata, also called capabilities document, which specifies the content provided and capabilities supported by the service. This operation provides the capability representation, which can improve interoperability; however, it does not provide the semantics of the data content and therefore not sufficient to allow semantic interoperability.

In such open environment, discovering, access and integrating information are difficult due to the semantic heterogeneity from using different terminology by the information providers and users. Interoperability will be crucial in data sharing and in building dynamic geographic information infrastructures [25]. The OWS standards, particularly the WMS standards address the technical aspects of data exchange. However, the semantic aspects are still significant problems in data sharing [26].

The problem solving approach is:

1. **Enriching the service metadata by describing them with the semantics used in the application domain in order to facilitate the discovery and access of services.**
   
   Overcoming the limitation of the service description by including the semantics for the information content in the capabilities document (service metadata) can improve semantic interoperability. Therefore, this research focuses on semantic description of Web Service for its information content.

2. **Provide a single entry point, geo portal, which gives the possibility to access the services along with their semantics.**
1.4 **Objective**

The aim of this research was to solve semantic problems to facilitate the sharing of geospatial data and services. To achieve this aim the following objectives were identified.

1. To develop an ontology based on the semantics in existing content standards, taxonomies and keywords in the environmental domain.
2. To extend the capabilities of Service description (Service Metadata) using the ontology to give meaning for the data content improving semantic interoperability.
3. To design and implement a prototype ontology-based geo portal for discovery and easy access of available geospatial data and services.

1.5 **Research Questions**

In this research work, we have tried to give answers to several questions related to the above mentioned objectives.

1. What is the state of the art technology to share geospatial data/services for users?
2. What are the problems in using the state of the art technology for sharing geospatial data/services?
3. Are there any content standards and classification schemes that can be used for describing environmental information (the use case for this research)?
4. Is it possible to derive an ontology from the existing content standards and standard classification schemes?
5. How do we extend the existing capabilities of service descriptions (service metadata) by using the semantics in content standards, classification schemes, and keywords to overcome the limitations of service descriptions?

1.6 **Methodology**

This research focused on solving semantic problems in sharing geospatial data and services. The problem solving approach was the development of a prototype ontology for the environmental domain application based on the taxonomy or semantics used in existing content standards and standard classification schemes. This is followed by description of web services using the ontologies developed, in order to facilitate the discovery of services. Finally, design and implementation of prototype ontology-based geo portal to give users access to the available geospatial data and services along with their semantics.

The methodology used for this research is as follows:
1.7 Tools Used

Efficient development tools are a prerequisite for the wide adoption of a new technology. The software tools used in this research are:

- Protégé (Ontology Editor) with “OWL plug-in” – for creating prototype ontology (http://protege.stanford.edu/);
- RACER – Knowledge base reasoner which can be connected to Protégé, used to test the ontology for its consistency (http://www.racer-systems.com/index.phtml);
- ArcIMS – for publishing OGC Web Map Services (WMS) (http://www.esri.com/software/arcgis/arcims/index.html);
- XML Spy – XML Editor, for editing the service metadata (http://www.altova.com/products_ide.html);
- DreamWeaver – for prototype geo portal development (http://www.macromedia.com/software/dreamweaver/).

1.8 Thesis Outline

Chapter two gives definition of Web Services and discusses the current state of the art technologies for Web Services. The common architecture of Web Services (Service Oriented Architecture), OGC Web Services architecture and the OGC Service taxonomy used to classify services based on their functionality are discussed in detail here. Chapter three
focuses on the concepts of Ontologies, Semantic Web and Description Logic as knowledge representation. It provides the knowledge representation languages used to develop ontologies. Then it introduces the syntax and basic constructors used in Description Logic to build knowledge bases. In addition, the knowledge base used in the research is presented as an example in this chapter. Chapter four reports the implementation of the research in four different sections. The first section is the implementation of the Ontology. This section elaborates the domain and application ontologies implemented. The second section discusses the implementation of OGC Web Services and the semantic annotation of these services. The third section presents the prototype design and implementation of the geo portal. The last section of this chapter evaluates the prototype implementations. Finally, chapter five provides the conclusion and recommendation for further work.

In addition, appendices are included to document the research work and to give a quick reference to the implementations. These include sections for Use case (Appendix A), Ontology Development (Appendix B), Web Service Implementation (Appendix C), Web Service Description (Appendix D) and Geo portal (Appendix E).
2 Web Services and Web Service Technologies

2.1 Web Services

The term Web Services has gained a lot of momentum in the last years. With the rapid growth of the technology, the applications of distributed systems have resulted in a significant development, which brought the Web Services arena. Web services made possible for users to integrate independently developed software applications and components with their own to solve their problems.

Although there seems to be convergence towards a common understanding, there is no single definition of Web Services [27]. Several major Web Services infrastructure providers have published their definitions for a Web Service. Web Services Architecture working group defined it as:

“A Web service is a software application or component identified by a URI\(^2\), whose interfaces and binding are capable of being described by standard formats and supports direct interactions with other software applications or components via Internet-based protocols”.

Here are a few more definitions from various industry sources:

Sun defines it as:

“A Web service is any executable code (program) that can be discovered, described, and invoked using XML-based messages via Internet-based protocols. -
http://www.sun.com/software/sunone/faq.xml#q8


“A Web service is programmable application logic accessible using standard Internet protocols.”

\(^2\) Universal Resource Identifier
The above definitions share the same understanding that Web Services are “software components” and they are accessible over “standard Internet protocols”.

A component is a unit of independent deployment, composable with other components and has no observable state. For a component to be independently deployed, it needs to encapsulate its constituent features. For a component to be composable it needs to be sufficiently self contained and encapsulates its implementation and interact with its environment by means of well-defined interfaces. A component should not have any observable state, which means that it is required that the component cannot be distinguished from copies of its own A software component can be defined as a unit of composition with specified interfaces that can be deployed independently and is subject to composition by third party [28].

A Web Service in general is a platform and implementation independent software component that can be:

- Described using a service description language
- Published to a registry of services
- Discovered through a standard mechanism
- Invoked through a declared API
- Composed with other services

A Web Service’s implementation and deployment platform details are not relevant to a program that is invoking the services. A Web Service makes itself available through its declared API and invocation mechanism. The web browser doesn’t care which Server is running the service. Similarly, the Server doesn’t care what kind of client is using it. This shared understanding between components allows Web Services to form a system of loosely coupled components.

2.2 Core Technologies of Web Services

Extensible Mark-up Language (XML) is the heart of Web Services technology. Web Services use XML to represent information. XML’s hierarchical structure, to nest and repeat mark-up, makes it easier to represent complex data structures than the traditional binary formats. The choice of XML also brought another advantage to Web services, the ability to describe data

---

3 Application Program Interface
types and to validate whether the data complies with its specification. This is through the use of XML meta-languages such as XML Schema.

Web Services can communicate because they all speak the same language: XML, to describe their interfaces and to encode their messages. XML-based Web Services communicate over standard Web protocols using XML interfaces and XML messages, that applications can interpret it. However, XML by itself does not ensure effortless communication. The applications need standard formats and protocols that allow them to properly interpret the XML. Hence three XML-based technologies have emerged as the de facto standards for Web Services [27]:

- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description Discovery and Integration (UDDI)

**SOAP** defines a standard XML messaging protocol for Web Services. SOAP provides a simple and consistent mechanism that allows one application to send an XML message to another application.

**WSDL** is an XML vocabulary for describing a Web Service. A WSDL document describes what functionality a Web Service offers, how it communicates, and where it is accessible. WSDL provides a structured mechanism to describe the operations of a Web Service, the formats of the messages, the protocols that it supports, and the access point of an instance of the Web Service.

**UDDI** provides an interface specification to register Web Services that the service provider offers and to locate Web Services that the service consumer would like to consume. UDDI itself is a Web Service; i.e. users communicate with it using SOAP messages.

### 2.2.1 Web Services Architecture (WSA)

A Web services architecture conforms to the concept of a Service Oriented Architecture (SOA). SOA is a pattern recognized in applying Web Service technologies to an application integration problem. SOA is a “publish-find-bind” concept, which makes it applicable to a wide variety of Web Services situations. SOA contains three roles: a Service Provider, a Service Requestor (Consumer), and a Service registry (Broker) [27, 29]. Figure 2-1 depicts the conceptual roles and operations of SOA.
A Service provider is responsible for creating a service description (service contract), publishing it to service registries and receiving Web Service invocation messages from service requestors.

A service requestor (consumer) is responsible for finding a service description published to service registries and is responsible for using service descriptions to bind to or invoke Web services hosted by service providers.

The Service registry is responsible for advertising Web Service descriptions published to it by service providers and for allowing service requestors to search the collection of service descriptions contained within the service registry. The service registry role is a broker (mediator) between service requestor and service provider.

A service provider makes the service available and publishes the contract that describes its interface to a service registry, advertising the service it provides. A service requestor queries the service registry to find a service. The Service registry gives information about where to find the service and its service contract. The service requester then uses the service contract to bind to the service.

A service requestor can be a human user or software agent. A service provider refers to the server or host that makes a Web service available and that publishes its description for its
interface. The term also used to refer to the person or company that hosts the service which made the service available on a network.

A WSA also includes three operations as in SOA: publish, find, and bind.

- The **publish** operation is an act of service registration or advertisement. A service provider publishes its web service description to a service registry, to advertise the details of the web service it provides to the service requestors.
- With the **find** operation the service requestor states search criteria, such as type of service. The service registry then matches the find criteria against its collection of published Web service descriptions. And it returns a list of service descriptions that match the find criteria to the service requestor.
- The **bind** operation is the client-server relationship between the service requestor and the service provider.

Publishing can be simple or complex depending on the implementation of the Service registry. In a simple publish scenario, the service registry role is played by the network itself, with publish being simply transferring the service description into a Web Server’s directory. The other publishing scenario can be to register to a standard interface specification for instance, UDDI registry.

The key to WSA is the service description. The service description is published by the service provider to the service registry and retrieved by the service requestor using the find operation. It is the service description that tells the service requestor how to bind to or to invoke the Web Service provided by the service provider. It also indicates what information will be returned to the service requestor as a result of the Web Service invocation.

### 2.2.2 Functional Components of Web Services Architecture

The three core functional components *transport*, *description*, and *discovery* in the web services architecture are implemented using **SOAP**, **WSDL**, and **UDDI**, respectively. Figure 2-2 shows the conceptual SOA architecture using these technologies. In this case the UDDI registry plays the role of service broker. The **register** and **find** operations are implemented using the **UDDI Publish** and **UDDI Inquiry** APIs respectively. A WSDL document describes the service contract, and is used to bind the client to the service [29].

The *transport* functional component defines the formats and protocols used to communicate between clients and services. In WSA, SOAP provides a simple messaging framework that
allows one application to send an XML message to another application. All transport functions are performed using SOAP.

For XML messaging Web Services use SOAP in all its data encoding, interaction style and protocol binding variations. SOAP defines an abstract binding framework that allows SOAP messages to be transferred using a variety of underlying transfer protocols. The SOAP specification defines a protocol binding for HTTP. Bindings have also been defined for HTTPS, SMTP, and other protocols.

![Diagram of SOA conceptual architecture with SOAP, WSDL, and UDDI.](image)

**Figure 2-2**: The SOA conceptual architecture with SOAP, WSDL, and UDDI.

The *description* functional component defines the language used to describe a service. The WSA description language is the Web Services Description Language (WSDL). A WSDL document describes what functionality a Web service offers, how it communicates, and where to find it. The service requestor uses the description to bind the client to the service.

Service description tells us how to bind to Web Services, but how did the service requestor get the service description? Clearly, we need some form of a Web Service discovery mechanism.

The *discovery* functional component provides a mechanism to register and find services. WSA discovery mechanism is implemented using a Universal Description, Discovery and Integration (UDDI) registry service.
UDDI is a general-purpose registry, which can be used to register any type of business, not just Web Services. UDDI doesn’t require that the services registered within it support SOAP interfaces or that they be described using WSDL.

Web Service does not necessarily require all of these technologies to be considered as a Web Service. SOAP is not a necessary technology for Web Services. It is possible that a component accessed through a simple HTTP POST can be considered a Web Service. A Web Service doesn’t necessarily has to be described using WSDL. A Web Service doesn’t have to appear in UDDI registry either to be considered as a Web Service. The service description can be put into a Web Server’s directory.

WSDL and UDDI however have limitations. Because UDDI uses XML to describe its data model, it guarantees syntactic interoperability, but it fails to provide a semantic description of its content. Two identical XML descriptions may have different meaning. XML’s lack of explicit semantics becomes an additional barrier to the UDDI’s discovery mechanism. Another limitation is its search mechanism which is limited to their automatic service selection based on a classification scheme used to describe the functionalities of the web services. Its keyword and category based search facilities are not sufficient for selecting suitable services. This classification mechanism does not provide a facility to capture the relationships between entities in its directory and therefore is not capable to make use of the semantic information to infer relationship during search. UDDI also lack matching at service capability level. By using capability based description of web services functionalities can be expressed in terms of the inputs and preconditions they require and their output and effects they produce [30], [31]. The main problems with WSDL and UDDI are that they do not provide semantically enriched description and capability based representations [32]. Currently, ways to address this limitations are being discussed to include semantics in UDDI and a WSDL-S proposal from IBM and University of Georgia aims to provide an extension to WSDL by adding semantics [31].

2.3 Geospatial Web Services

Geospatial Web Services are web services with geospatial content. Geospatial Web Services provide access to spatial data and GIS functionality through the internet. This makes it possible for users to integrate them with their own systems and applications without the need to install special type of GIS software.

One of the most specific advantages of Geospatial Web Services is global access to the geospatial datasets which are created and updated by particular organizations. Because they
are based on simple and non-proprietary standards, Web Services make it possible for computer programs to communicate directly with one another and exchange data regardless of location, processing platforms, operating systems, or languages. Web Services can thus lower the costs of software integration and data-sharing.

2.3.1 Geospatial Services Architecture

OGC and ISO/TC 211 in their standardization effort have jointly developed an international standard for geospatial service architecture. ISO 19119 “Geographic Information – Services” has been adopted as part of the OGC Abstract Specification, Topic 12 “OGC Architecture”. ISO 19119 is based on the Reference Model of Open Distributed Processing (RM-ODP) [ISO/IEC 10746], an international standard for architecting open, distributed processing systems [33], [34], [35].

Architecture is a set of components, connections and topologies defined through a series of viewpoints. OGC Service Architecture follows SOA and ISO 10746 (RM-ODP). Based on the RM-ODP, the following viewpoints are defined [34], [36], [35]:

- **The enterprise viewpoint** is concerned with the purpose, scope and policies of an enterprise or business and how they relate to the specified system or service. An enterprise specification of a service is a model of that service and the environment with which the service interacts. It covers the role of the service in the business and the human user roles and business policies related to the service.
- **The computational viewpoint**: is concerned with the interaction patterns between the components (services) of the system, described through their interfaces.
- **The information viewpoint**: is concerned with the semantics of information and information processing.
- **The engineering viewpoint**: is concerned with the design of the infrastructure required to support distribution.
- **The technology viewpoint**: describes the implementation of the RM-ODP system in terms of a configuration of technology objects representing the hardware and software components of the implementation.

The Computational Viewpoint considers the functional structure of an ODP system. In order to focus on the concept of function, this viewpoint considers ODP to be carried out by interactions between interfaces without referring, if possible, to how such interfaces may physically be implemented. Elements of the computational viewpoint are:
**Service**: distinct part of the functionality that is provided by an entity through interfaces.

**Interface**: named set of operations that characterize the behavior of an entity.

**Operation**: specification of a transformation or query that an object may be called to execute. It has a name and a list of parameters.

The information viewpoint into a service architecture covers the information and data structures which are processed by applications conforming to that architecture. This information includes information which might be characterized as either data or as metadata. The information viewpoint, however, concerns itself mainly with metadata. One of the most important parts of this metadata is the service information which describes the capabilities and content of service components in the architecture.

The information viewpoint is also concerned with the semantics of the information processing. Each particular service will need to define its syntactical interfaces through operations and its semantics through description of the meaning of the operations and their legal sequencing. The focus of this research is in the information view point and computational view point which deals with the semantics of information, which also depends on the functionalities which categorizes the services and description of the operations.

There exist multiple possible taxonomies for services, based on various classification dimensions. The model for the information viewpoint is provided by ISO 19101. ISO 19101 defines the Extended Open Systems Environment (EOSE) model for geographic information. The EOSE defines classes of services based on the semantic type of computation that they provide. EOSE provides the functional decomposition of the services for the geographic domain by extending the more general Open System Environment model [ISO/IEC TR 14252] [35].

Six classes of information technology services are defined in ISO19119 that shall be used to categorize geographic services.

- **Human interaction services** are services for management of user interfaces, graphics, multimedia, and for presentation of compound documents. An example is Catalogue viewer: Client service that allows a user to interact with a catalogue to locate, browse, and manage metadata about geographic data or services.

- **Model/Information management services** are services for management of the development, manipulation, and storage of metadata, conceptual schemas, and datasets. An example of Information management service is Map access service. Service that provides a client access to a geographic map in a form of graphics.
- **Workflow/Task services** are services for support of specific tasks or sequence of work-related activities conducted by different persons.
- **Processing services** are services that perform large-scale computations involving substantial amounts of data, e.g. coordinate transformations.
- **Communication services** are services for encoding and transfer of data across communications networks. An example is portal service.
- **System management services** are services for the management of system components, applications, and networks. These services include management of user accounts and user access privileges.

The OWS service taxonomy implements this classification approach for grouping services that are semantically similar in order to facilitate browsing and discovery. A service catalogue compliant with ISO 19119 categorizes service metadata instances in the categories of this geographic service taxonomy.

### 2.3.2 OGC Web Services (OWS) Framework

OGC Web Services are individual components of dynamic geospatial computing application. The OGC implementation constrains the use of common XML encodings, HTTP protocol for transports, tightly defined interface syntax, specific information models for service descriptions and other metadata. OGC web Services are self describing and supports capabilities representation by the GetCapabilities operation, which specifies the content and supported capabilities of the service [36]. OGC has implemented service description for OWS, using capabilities document, instead of WSDL; OGC catalogue services for service registry, instead of UDDI; and HTTP for transport.

OGC Web Services (OWS) framework elements include Client Services, Registry Services, Processing-Workflow Services, Portrayal Service, and Data Services. Client Services are client applications that interact with human users and server side applications [36], [37]. Client Services are services of type Geospatial Human Interaction Services defined by ISO 19119 and OGC Abstract Specification Topic 12. Registry services provide a mechanism to register, describe, and search the data and services. Portrayal services provide capabilities for visualization of geospatial information. These include Web Map Services (WMS), and Coverage Portrayal Services (CPS). Data Services are services that serve geospatial data. These include Web Feature Services (WFS), Web Coverage Services (WCS) and Sensor Collection Services (SCS).
2.3.2.1 Web Map Services (WMS)

One of the OWS implementations is Web Map Service (WMS). A Web Map Service produces maps of spatially referenced data dynamically from geographic information. OGC’s Web Map Service specification defines “map” as a portrayal of geographic information as a digital image file suitable for display on a computer screen. A map is not the data itself. WMS maps can be presented in a pictorial format such as PNG, GIF or JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats [38].

The OGC WMS specification enables the creation of a network of interoperable map servers from which clients can overlay and build customized maps. This functionality allows organizations to create WMS networks that enable users to combine data from different sources based on their individual needs.

HTTP and SOAP are infrastructure technologies that enable binding to services deployed across the Web. At present the distributed computing platform supported by the OGC Web Services is HTTP. HTTP supports two request methods: GET and POST. One or both of these methods may be defined for a particular OGC Web Service type and offered by a service instance. The use of the Online Resource URL differs in each case.

WMS supports three operations, namely, GetCapabilities which returns service-level metadata, GetMap which returns a map whose geographic and dimensional parameters are well-defined and an optional GetFeatureInfo operation that returns information about particular features shown on a map. A basic WMS shall support the basic service elements, the GetCapabilities operation, and the GetMap operation. These operations can be invoked using a standard web browser by submitting requests in the form of Uniform Resource Locators (URLs). The content of such URLs depends on which operation is requested.

2.3.2.2 GetCapabilities operation

The mandatory GetCapabilities operation of OWS allows any client to retrieve metadata about the capabilities provided by any service that implements an OWS. The response to the GetCapabilities request is a service metadata, called “capabilities XML”. It is an XML file that is returned to the requesting WMS client. This service metadata document, which is a machine readable (and human-understandable), primarily contains description of the WMS service’s information content and acceptable request parameters. This service metadata also makes an OWS server partially self-describing, supporting late binding of clients [37].
The GetCapabilities request returns the WMS server’s service-level metadata. First, however, a WMS client needs to find out what it can request from a particular WMS server. In order to find out what layers a WMS server supplies and what projections it supports, a WMS client makes the GetCapabilities Request. The “Capabilities document” returned enables WMS clients to formulate valid requests and enables the construction of searchable catalogues that can direct WMS clients to particular WMS servers; to bind to the services.

The GetCapabilities operation needs to retrieve a complete listing of which interfaces a map server supports and what map layers it can serve in response to the invocation of the map interface. The GetCapabilities operation provides WMS clients the following functionalities:

- All interfaces a WMS server can support;
- Image formats it can serve (e.g., GIF, JPG, PNG);
- List of spatial reference systems available for delivery of map data from the WMS server;
- List of all exception formats for return of exception that are available from the WMS server. Inclusion of a value for this attribute is optional;
- List of one or more map layers available from a particular WMS server. Inclusion of a value for this attribute is optional;
- Whether a WMS server supports the optional FeatureInfo interface.
- List of all the vendor specific capabilities (or properties) that are available for modifying or controlling actions of a particular WMS server, with the current value of each capability. Inclusion of a value for this attribute is optional.

The response to a GetCapabilities request, a Capabilities XML document, composed of three main sections; Service Metadata, Operation Signature, and Content Metadata. The service Metadata section contains common services metadata that provides human readable description of the services. The Operation Signature (Operation Metadata) section contains a description of the operations that the service supports. Content Metadata section contains data content of the service; any service specific metadata such as feature list for WFS, layer list for WMS, or ISO 19115 metadata element to describe the content of the service.

OGC Web Services provide the self describing GetCapabilities operation, which specifies the capabilities and the data content of the service. This operation provides capability representation, which can improve interoperability; however, it does not provide the semantics of the data content and therefore not sufficient to allow semantic interoperability.
2.3.2.3 GetMap Operation

GetMap operation of OWS returns a map for a specific request from the client. When requesting a map the URL indicates what information is to be shown on the map, the geographic extent of the map, the desired coordinate reference system, and the output image width and height. When two or more maps are produced with the same geographic parameters and output size, the results can be accurately overlaid to produce a composite map. Furthermore, individual maps can be requested from different servers. The response to the GetMap operation is a map with the specified format and parameters. The Getmap request URL includes the host name, Service name and the parameters and their values as shown in this example:


2.4 Concluding Remarks

OGC Web Services provide the self describing Getcapabilities operation, which specifies the capabilities and the data content of the service. This operation provides capability representation, which can improve interoperability; however, it does not provide the semantics of the data content and therefore not sufficient to allow semantic interoperability. Such semantic problems can be solved if the RDF\(^4\) and OWL\(^5\) languages are used to provide semantics or meaning of the data.

\(^4\) Resource Description Framework
\(^5\) Web Ontology Language
3 Semantic Web and Ontology as Knowledge Representation

One of the problems that occur when data are exchanged is the problem of semantic interoperability. Interoperability is recognized for joining heterogeneous computer systems into synergistic units that facilitate a more efficient use of information resources. As [25] points out, interoperability is the basis for facilitating data sharing and helping to resolve duplication. Overcoming semantic interoperability is an issue that requires solution. The first step of interoperability is to use common formats and messages, but the next step is to agree on the meaning of those elements [36]. Data sharing between different information communities needs a common understanding of the semantics of the data, in the specific domain. Semantic interoperability therefore, requires formally defined concepts and terms [18],[39].

3.1 Semantic Web

The semantic web was a vision by Tim Berner Lee, and currently a project undertaken by World Wide Web Consortium (W3C). The Semantic Web is a project aimed to make web pages understandable by computers, to create a universal medium for information exchange by giving meaning (semantics), to the content of documents on the Web [40]. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community.

Tim Berners Lee, the creator of the World Wide Web, says:

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [41].

As [42], the semantic web relies on ontological knowledge in order to achieve semantic interoperability and for representing resources on the ground of concepts and meanings derived from their metadata.

A Semantic Model defines the terms and concepts used to describe and represent an area of knowledge or some part of the world. A semantic model usually includes concepts in the
domain of interest, relationships among them, their properties, and their values. Currently, ontology is the common way to model semantics by expressing meaning and concepts that helps to define a common ground between different information communities. Ontology provides the structured vocabulary and semantics which can be used in the mark-up of web resources in order for the machine/computer to understand it [43]. If different information communities define their domain ontologies, these could be used by others to check whether the data (based on their semantics) are usable, or not [18]. Therefore, Ontologies become a widely accepted state-of-the-art knowledge representation, and have thus been identified as the central enabling technology for the Semantic Web.

3.2 Ontology as Knowledge Representation

People conceptualize the real world in different ways and as a result different specifications of the same “things” exist [18]. An ontology is defined, from computer science perspective, as a formal explicit description of concepts in a domain of discourse. Ontology defines a common vocabulary for people who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations between them. Sharing common understanding of the structure of information among people or software agents is one of the common goals in developing ontologies [44]. Ontologies can enhance the functioning of the Web in many ways. They can be used to improve the accuracy of Web searches, i.e. the search program can look for only those pages that refer to a precise concept instead of the ones using ambiguous keywords [41].

A concept is anything about which something can be said. People have different terms and phrases for the same concept. However, that does not necessarily mean that those people are referring to different concepts. Every concept with in an ontology can be described explicitly by assigning attributes. Attributes allow more complex relations to be modelled using the ontology.

Knowledge acquisition has been done by domain experts and based on relevant documentations to extract this conceptualization in the form of a hierarchy of domain vocabulary. An ontology which is tied to a specific domain is known as domain ontology. Domain ontologies describe a certain part of the world from a certain perspective. A domain ontology relates concepts to each other and restricts their interpretation. An ontology which is not tied to a particular problem domain but attempts to describe general entities is known as a foundation ontology or upper ontology [45]. The task of constructing an application ontology which can be used in specific application in a domain, however, lies in the responsibility of the provider of the information.
3.3 Description Logic

Description logics (DL) are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain (the world) in a formal and structured way. As the name implies description logics, defines the concepts of a domain (its terminology) and describe the properties of objects and individuals occurring in the domain (the world description). One of the characteristics of these languages is they are equipped with logic-based formal semantics. The other important feature is they provide reasoning service which allows one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base [46].

Most knowledge representation languages such as RDF schema and OWL have their foundation on Description Logic. These formalisms are represented by First Order logic. DL and these derived mark-up languages are a sort of defacto standard for representing ontologies based on their well founded semantics and available reasoning services [47].

A knowledge representation based on Description Logics provides facilities to set up knowledge bases, to reason about their content. A DL knowledge base consists of two components, a set of terminological axioms called TBOX and a set of assertional axioms called ABOX. The TBOX introduces the terminology (vocabulary of an application domain), and the ABOX contains assertions about named individuals in terms of this vocabulary. The vocabulary contains concepts, which represent sets of individuals, and roles which indicate binary relationships between individuals [46].

The primary reason for this distinction is that, the separation can be useful when describing and formulating decision-procedures for various DLs. Reasoners might process the TBox and ABox separately, for performance purpose. For example, ‘classification’ is related to the TBox, and ‘instance checking’ to the ABox. The secondary reason is from the knowledge base modeler’s perspective, which distinguish between our conception of terms/concepts in the world (class axioms in the TBox) and particular manifestations of those terms/concepts (instance assertions in the ABox) [47].

In a DL language, elementary descriptions are atomic concepts (unary predicates) and atomic roles (binary predicates). More complex concept descriptions can be built using atomic concepts (A) and and primitive roles (R) by means of constructors formed by the following syntax rules [46]. C, and D are concept names or atomic concepts.
Syntax Name

\[ C, D \rightarrow A \] Atomic Concept, Concept name
\[ \top \] Universal Concept, Top Concept
\[ \bot \] Bottom Concept
\[ R \] Atomic Role, Role name
\[ \neg A \] Atomic Negation
\[ C \cap D \] Concept Conjunction (Intersection)
\[ \forall R.C \] Universal Restriction (Value restriction)
\[ \exists R.T \] Limited Existential Restriction
\[ \exists R.C \] Full Existential Restriction

An interpretation \( \mathcal{I} \) consists of a non-empty set \( \Delta^I \) (the domain of the interpretation) and an interpretation function, which assigns to every atomic concept \( A \) a set of \( A \subseteq \Delta^I \) and to every atomic role \( R \) a binary relation \( R^I \subseteq \Delta^I \times \Delta^I \). The interpretation function is extended to concept descriptions by the following definitions.

\[
\begin{align*}
\top^I &= \Delta^I \\
\bot^I &= \emptyset \\
\neg A^I &= \Delta^I \setminus A^I \\
(C \cap D)^I &= C^I \cap D^I \\
(C \cup D)^I &= C^I \cup D^I \\
(\forall R.C)^I &= \{ a \in \Delta^I \mid \forall b. (a, b) \in R^I \rightarrow b \in C^I \} \\
(\exists R. T)^I &= \{ a \in \Delta^I \mid \exists b. (a, b) \in R^I \} \\
(\exists R. C)^I &= \{ a \in \Delta^I \mid \exists b. (a, b) \in R^I \land b \in C^I \}
\end{align*}
\]

A DL based knowledge representation system not only stores terminologies and assertions, but also offers reasoning services such as satisfiability (whether a description is non-contradictory) and subsumption (whether one description is more general than the other one). Classification of concepts in a form of subconcept-superconcept relationships (called subsumption) allows structuring the terminology in the form of a subsumption hierarchy.

**Definition 1 - Subsumption:** A concept \( C \) is subsumed by a concept \( D \) with respect to a TBOX \( T \), denoted by \( C \sqsubseteq_T D \), if \( C^I \subseteq D^I \) for every interpretation \( I \) of \( T \).
**Definition 2 – Equivalence**: Two concepts C and D are equivalent with respect to T if $C^I = D^I$ for every interpretation $I$ of T.

Hence, Concept C is equivalent with concept D if C is subsumed by D and D is subsumed by C;

$C \equiv D$ means $C \sqsubseteq D$ and $D \sqsubseteq C$.

The second component of the knowledge base is the world description or ABox. In the ABox, one introduces individuals by giving them names, and one asserts properties of these individuals. Individual names are denoted by a, b, c and concept and role names by C and R, and one can make assertions in the ABox: C(a), and R(b, c). The first one is called concept assertion and it means, a belongs to (the interpretation of) C. The second is called role assertion and it is said that c is a filler of role R for b.

For example, in our Knowledge base, *Service* and *Feature_Type* are concepts (TBox). *Service* class has a property *hasFeatureType* which relates its individuals with the individuals of the *Feature_Type* class. This property is restricted to have only individuals of *Feature_Type* class as its allowed values and is described using DL:

$$\forall \text{hasFeatureType.} \text{Feature_Type}$$

The above statement means that any individual of *Service* class has *hasFeatureType* as its property that relates it with the individuals of *Feature_Type* class. The subclasses of *Service* class, e.g. *Map_Access_Service* inherit this property. The following statement shows that *Africa_Hydrology_Web_Map_Service* (ABox), an individual of *Map_Access_Service* class, has a relation with the *DrainageBasin*, an individual of *Feature_Type* class, as its value for the property *hasFeatureType*.

*Map_Access_Service (Africa_Hydrology_Web_Map_Service)*

*Vector_FeatureType (DrainageBasin)*

*hasFeatureType (Africa_Hydrology_Web_Map_Service, DrainageBasin)*

*hasFeatureType* is a role. Both *DrainageBasin* and *Africa_Hydrology_Web_Map_Service* are individuals. The first two statements mean *Africa_Hydrology_Web_Map_Service* is *Map_Access_Service*, and *DrainageBasin* is *Vector_FeatureType* respectively. The last statement means *DrainageBasin* is a *Feature_Type* for *Africa_Hydrology_Web_Map_Service*. 
### 3.4 Working Knowledge Base

An ontology, typically contains a hierarchical description of important concepts in a domain, and describe these concepts through an attribute value using their crucial properties [48]. Ontology development has two main phases. The first phase is defining the concepts or classes, their properties and restrictions on these properties. Once the classes and their corresponding properties are defined, the second phase is to include individuals or instances of the classes along with their values for the corresponding properties. The second phase is a transition from the ontology to knowledge base. A knowledge base is an ontology populated with instances.

The main steps to build an ontology are [44]:

1. To identify concepts, their relations and main attributes;
2. To organize the concepts in an a form of hierarchy;
3. Detailed definition of the concepts using descriptive logic and definition of constraints and restrictions through properties;
4. To populate the ontology with individuals and their corresponding values to the attributes (here it becomes a knowledge base)

This research identifies the concepts and their associated properties for the domains, ‘Environment’, and ‘Geospatial Services”, to be used. The major concepts identified in our knowledge base are:

- **Environmental_Domain** - concept which represent the environmental domain.
- **Service** - concept which represents the abstract geospatial service.
- **Feature_Type** - concept which represents datasets in the environmental domain that will be used in the geospatial services.

Other associated concepts are also identified which are related with these concepts with some properties. These are:

- **Attribute** - concept associated with the **Feature_Type** concept using the property *hasAttribute* (i.e. to define that any Feature type has an attribute).
- **Service_Metadata** - concept associated with the **Service** concept using the property *hasServiceMetadata* (to define any Service has Service Metadata).
- **Binding** - concept which has a relation with the **Service** concept through the property *hasBinding* (to define a Service has a Binding to make itself accessible).
Further the concepts ServiceMetadata, and Binding are related with the associated concepts Operation, and BindingMethod respectively using their properties.

- The ServiceMetadata concept associated with Operations concept using the property hasOperation (used to define that a service metadata has operation metadata which describes the set of operations supported).
- The Binding concept associated with BindingMethod concept using the property hasBindingMethod.
- The Operations concept in addition associated with OperationsParameter concept using the property hasOperationParameter (used to define that the operation has a set of parameters).

The above concepts can be qualified and described using description logic expressions.

Environmental_Domain is subclass of ℰ (Top concept). This can be written as:

\[ \text{Environmental\_Domain} \sqsubseteq \text{ℰ} \]

Service and Feature_Type are also subclass of ℰ

\[ \text{Service} \sqsubseteq \text{ℰ} \]
\[ \text{Feature\_Type} \sqsubseteq \text{ℰ} \]

Feature_Type in addition has the following necessary conditions (properties and restrictions). 

\[ \forall \text{hasDomainCategory}.\text{Environmental\_Domain} \]
\[ \forall \text{hasMetadata.iso19115:MD\_Metadata} \]
\[ \exists \text{hasSpatial\_Reference\_System.j.3:SC\_CoordinateReferenceSystem} \]
\[ \forall \text{hasSpatial\_Reprentaion\_Form.iso19115:MD\_SpatialRepresentationTypeCode} \]

The prefixes iso19115, and j.3 are prefixes attached to concepts in the imported metadata ontology. The prefix “iso19115” indicates elements of ISO19115 Metadata and “j.3” indicates the elements of the “Ontology for Geographic Information - Spatial Referencing by Coordinates (ISO 19111:2003)” accompanied with the imported ontology. The above statement can be written as the following equivalence statement using description logic.

\[ \text{Feature\_Type} \equiv \text{ℰ} \sqcap \forall \text{hasDomainCategory}.\text{Environmental\_Domain} \sqcap \forall \text{hasMetadata.iso19115:MD\_Metadata} \sqcap \exists \]
hasSpatial_Reference_System.j.3:SC_CoordinateReferenceSystem \ \lor
hasSpatial_Representation_Form.iso19115:MD_SpatialRepresentationTypeCode

The above statement and restrictions states that Feature types has domain category from the class of Environmental Domain, and has service metadata in a form of ISO metadata, has spatial reference system, and has spatial representation form, from the spatial representation codes defined by ISO, e.g Vector or Grid.

Feature_Type subsumes two classes: Grid_FeatureType and Vector_FeatureType.

I.e. Grid_FeatureType is subclassof FeatureType, and described using description logic as:

\[
\text{Grid}\_\text{FeatureType} \sqsubseteq \text{FeatureType}
\]

In addition Grid_FeatureType can be qualified by its property hasDimension which indicates the number of columns and rows as a Grid data. Therefore, it can be described as follows.

\[
\text{Grid}\_\text{FeatureType} \equiv \text{FeatureType} \lor \ hasDimension.iso19115:MD\_Dimension
\]

The Grid_FeatureType inherits all the properties and the conditions from Feature_Type. Therefore, it can be written as:

\[
\text{Grid}\_\text{FeatureType} \equiv \top \lor \ has\text{DomainCategory}.\text{Environmental}\_\text{Domain} \lor \ \exists
\ hasMetadata.iso19115:MD\_Metadata
\lor \ hasSpatial\_Reference\_System.j.3:SC\_CoordinateReferenceSystem
\lor \ hasSpatial\_Representation\_Form.iso19115:MD\_SpatialRepresentationTypeCode
\lor \ hasDimension.iso19115:MD\_Dimension
\]

The same for Vector_FeatureType:

\[
\text{Vector}\_\text{FeatureType} \sqsubseteq \text{FeatureType}
\]

In addition it has the properties hasAttribute and hasGeometricType, which describes that vector feature type, has attributes and it has geometric type (e.g., point). And has the following necessary conditions (restrictions on its properties).
\( \forall \) hasAttribute.Attribute
\( \forall \) hasGeometricType.iso19115:MD_GeometricObjectTypeCode

This can be written in the following expression.

\[ \text{Vector	extunderscore FeatureType} \equiv \text{Feature	extunderscore Type} \land \forall \text{hasAttribute.Attribute} \land \forall \text{hasGeometricType.iso19115:MD_GeometricObjectTypeCode} \]

That is the same as the expression:

\[ \text{Vector	extunderscore FeatureType} \equiv \top \land \forall \text{hasDomainCategory.Environmental	extunderscore Domain} \land \forall \text{hasMetadata.iso19115:MD_Metadata} \land \exists \text{hasSpatial	extunderscore Reference	extunderscore System.j.3:SC	extunderscore CoordinateReferenceSystem} \land \forall \text{hasSpatial	extunderscore Representation	extunderscore Form.iso19115:MD	extunderscore Spatial	extunderscore Representation	extunderscore TypeCode} \land \forall \text{hasAttribute.Attribute} \land \forall \text{hasGeometricType.iso19115:MD_GeometricObjectTypeCode} \]

\( Service \) and \( Service\_Metadata \) concepts are also expressed using equivalence statement as follows:

\[ \text{Service} \equiv \top \land \forall \text{hasBinding.Binding} \land \forall \text{hasFeatureType.Feature	extunderscore Type} \land \geq 1\text{hasFeatureType} \land \forall \text{hasProvider.iso19115:CI	extunderscore ResponsibleParty} \land \forall \text{hasProvider} \land \forall \text{hasService	extunderscore Metadata.Service	extunderscore Metadata} \land \forall \text{hasServiceType.Service	extunderscore Type} \]

\[ \text{Service	extunderscore Metadata} \equiv \top \land \forall \text{hasData	extunderscore Identification.Feature	extunderscore Type} \land \forall \text{hasOperation	extunderscore Metadata.Operations} \land \forall \text{hasService	extunderscore Identification.iso19115:MD	extunderscore Service	extunderscore Identification} \]

\[ \text{Operations} \equiv \top \land \forall \text{hasBinding.Binding} \land \forall \text{hasParameters.Operation	extunderscore Parameters} \]

\[ \text{Binding} \equiv \top \land \forall \text{hasBindingMethod.Binding	extunderscore Method} \land \forall \text{hasPorttype.Port	extunderscore Types} \]

The major concepts and properties of our knowledge base are shown in the description logic expressions below. Because the ontology is big, it is difficult to show all the concepts and all the instances of concepts in the ontology. A list of instances is included in Appendix A.
Major concepts

EnvironmentalDomain ⊆ τ
Service ⊆ τ
Feature_Type ⊆ τ

Concepts related to the Feature_Type concept

Feature_Type ≡ τ ∩ ∀

| hasDomainCategory,Environmental_Domain |
| ∩ ∀ | hasMetadata.iso19115:MD_Metadata |
| ∩ ∃ | hasSpatial_Reference_System.iso19115:SC_CoordinateReferenceSystem |
| ∨ | hasSpatial_Representation_Form,iso19115:MD_SpatialRepresentationTypeCode |

Vector_FeatureType ⊆ Feature_Type

Vector_FeatureType ≡ Feature_Type ∩ ∀ hasAttribute.Attribute ∩ ∀

| hasGeometricType.iso19115:MD_GeometricObjectTypeCode |

iso19115:MD_GeometricObjectTypeCode ≡ {Point, Curves, Surface, Solid, Complexes, Composites}

Grid_FeatureType ⊆ Feature_Type

Grid_FeatureType ≡ Feature_Type ∩ ∀

| hasDimension.iso19115:MD_Dimension |

iso19115:MD_Dimension ≡ {iso19115:Column, iso19115:Row}

Concepts related to the Service concept

OperationParameters ⊆ τ
Operations ⊆ τ
GetCapabilities ⊆ Operations
GetMap ⊆ Operations
GetFeatureinfo ⊆ Operations
Ontology Based Portals for Accessing Geospatial Data and Web Services: Solving Semantic Problem in Data Sharing

\[ \text{Service} \equiv \tau \land \forall \text{hasBinding.Binding} \land \forall \text{hasFeatureType.Feature_Type} \geq 1 \text{hasFeatureType} \land \forall \text{hasProvider.iso19115:CI_ResponsibleParty} = 1 \text{hasProvider} \land \forall \text{hasService_Metadata.Service_Metadata} \land \forall \text{hasServiceType.Service_Type} \]

- **Communiction_Services** $\subseteq$ Service
- **Geospatial_Model_Information_Management_Services** $\subseteq$ Service
- **Human_Interaction_Services** $\subseteq$ Service
- **Processing_Services** $\subseteq$ Service
- **System_Management_Services** $\subseteq$ Service
- **Workflow_and_Task_Services** $\subseteq$ Service

- **Catalogue_Service** $\subseteq$ Geospatial_Model_Information_Management_Services
- **Coverage_Access_Service** $\subseteq$ Geospatial_Model_Information_Management_Services
- **Feature_Access_Service** $\subseteq$ Geospatial_Model_Information_Management_Services
- **Gazetteer_Service** $\subseteq$ Geospatial_Model_Information_Management_Services
- **Map_Access_Service** $\subseteq$ Geospatial_Model_Information_Management_Services

- **Catalogue_Viewer** $\subseteq$ Human_Interaction_Services
- **Geospatial_Feature_Editor** $\subseteq$ Human_Interaction_Services
- **Geospatial_Visual** $\subseteq$ Human_Interaction_Services

- **Processing_Services-Metadata** $\subseteq$ Processing_Services
3.5 Knowledge Representation Languages

Currently, the World Wide Web is based primarily on documents written in HTML, a markup language used for presentation of text, and multimedia such as images and interactive forms. HTML has limited ability to classify or represent the blocks of text on a page, apart from the roles they play in documents’ organization and their visual display. The data-centric, customizable markup language, XML, is one step ahead from this limitation, allowing data and resources to be represented using standardized tags. Although, XML is suitable to represent data and resources, it cannot be used to describe them semantically using their metadata [18]. The Semantic Web addresses this shortcoming, using the descriptive technologies RDF and OWL. The machine-readable descriptions allow us to add meaning to the content, thereby facilitating automated information gathering and research by computers [40]. The World Wide Web Consortium (W3C) envisions, Semantic Web, that the future will hold an extensive technological development that is intended to structure the web resources. For this purpose, the consortium has developed the machine readable and descriptive languages, RDF and RDF/Schema, hoping to achieve the vision of the Semantic Web, with which in the future, people will be able to publish and structurally describe their information using terms that are already defined and published in the Internet [18].

The Semantic Web extends the World Wide Web through the use of standards, markup languages and related processing tools such as XML, XML Schema, RDF, RDF Schema and OWL. The most crucial enablers of the Semantic Web technology along with XML and namespaces are URIs (Uniform Resource Identifiers). The Semantic Web is generally built on syntaxes which use URIs to represent data. These syntaxes are called “Resource Description Framework” syntaxes [49]. These along with description logic form RDF which can be used to express anything. Any resource that can be identified with a URI can be described, so the reasonors can reason about it [40].

3.5.1 XML, XML Schema and URI

The major advantage of XML is its extensibility. The languages RDF and OWL used XML as their foundation. However, the problem with XML is that its semantics are implicit. That is, meaning is derived from the human understandable tags based on common understanding.
and consensus. The disadvantage of these implicit semantics is that they may trigger ambiguity and allow freedom to fragmented representations which are not semantically interoperable [50].

A Document Type Definition (DTD) can be used to ensure that XML documents conform to a common grammar. DTD defines the document structure with a list of legal elements. To validate an XML document, some form of validating rules need to be provided. This can be done by any DTD [51]. A DTD provides syntax for an XML document; however, it doesn’t provide the data types and semantics of the elements. The semantics of a DTD are implicit. That is, the meaning of an element in a DTD is either inferred by a human due to the name assigned to it, is described in a natural-language comment within the DTD, or is described in a document separate from the DTD [50].

The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD. However, it has additional functionalities beyond the DTD. One of the greatest strengths of XML Schemas is the support for data types. It is easier to describe permissible document content and to validate the correctness of data. The other strength is its extensibility, because XML Schema is expressed in XML-based syntax, which can be used to define, describe and catalogue XML vocabularies for classes of XML documents. These syntaxes are called XML schema language also referred as XML Schema Definition (XSD). The purpose of the XML schema language is to provide an inventory of XML mark-up constructs with which to write schemas [51].

The nature of most Semantic Web resources represented by RDF and OWL documents however is significantly different from World Wide Web resources represented by HTML and XML documents. The latter are intended for human consumption and syntactic interoperability. Where as, the former are Knowledge Representation formalisms. I.e. XML, XML Schema and URI are not knowledge representation formalisms, though they contribute to the syntax and the semantics of RDF, RDFS and OWL.

Namespaces are used to discriminate XML elements, attributes, and data types defined in application-specific domains from one another. It is a functionality given by XML and is used for designating a local name in which the set of tags are valid to avoid conflict between other sets of tags. XML namespaces use URI. URIs can be locators, or names. URI locators are known as Uniform Resource Locators (URL). Namespace prefixes are used, usually a few number of characters, to abbreviate the entire namespace.

Here are two examples of namespaces:
the Dublin Core metadata [52] and the ontology developed in this research, respectively. The namespace prefixes, are ‘dc’ and ‘onto’. Any element or resource can be referred using the namespace prefix in front of it, dc:creator for instance to refer to the creator property in the Dublin Core metadata.

3.5.2 Resource Description Framework (RDF) and RDF-S

Resource Description Framework (RDF) is a framework for metadata description developed by W3C with the intention that encoding knowledge on Web pages to make it understandable to electronic agents for searching information [44]. RDF is a machine readable language based on XML, which is used to describe any resource in the World Wide Web. A resource is anything that we can identify. RDF provides a common framework for describing information in order to exchange them between applications with out a loss of their meaning.

RDF is based on the idea that any resource can be described using its properties which have a value [53]. A Statement consists of the combination of a Resource, a Property, and a value. These parts are known as the 'subject', 'predicate' and 'object' of a Statement. Meaning is expressed by RDF, which encodes it in these sets of triples, which can be written using XML tags, to describe a resource or thing using its properties and corresponding values. RDF uses URI, which can be used to uniquely identify resources and as the basis of its mechanism for identifying them using these triples in statements. The triples of RDF form webs of information about related things, thus forms a semantic network. Because RDF uses URIs to encode this information in a document, the URIs ensure that concepts are not just words in a document but are tied to a unique definition that everyone can find on the Web [41].

In RDF, a document makes assertions that particular things have properties with certain values. Subject and object of the triple are each identified by a URI, just as used in a link on a Web page, and can be defined within the current document or refer to another resource on the Web. The predicate can be any (namespace qualified) XML name. This can be shown as in the following example.

The “creator” of the “Environmental Geospatial Services Ontology” is “Aster Denekew”. “Creator” is property which has a value “Aster Denekew”. “Environmental Geospatial Services Ontology” is a resource. It can refer to the online resource and written as: 

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:onto="http://geoserver.itc.nl/aster/Ontology/EnviromentalServicesOntology.owl"
  xmlns:aster="http://aster.itc.nl"
>
  <rdf:Description id="http://aster.itc.nl/aster-ontology/aster" about="http://aster.itc.nl/aster-ontology/aster"
    xmlns:aster="http://aster.itc.nl"
>
    <!-- DC Metadata -->
    <rdf:Description rdf:about="http://aster.itc.nl/aster-ontology/aster"
      xmlns:aster="http://aster.itc.nl"
    >
      <aster:creator>Aster Denekew</aster:creator>
      <aster:contact>Aster Denekew</aster:contact>
      <aster:license>CC BY-NC-SA 3.0</aster:license>
    </rdf:Description>

    <!-- Ontology Metadata -->
    <rdf:Description rdf:about="http://aster.itc.nl/aster-ontology/aster"
      xmlns:aster="http://aster.itc.nl"
    >
      <aster:creator>Aster Denekew</aster:creator>
      <aster:contact>Aster Denekew</aster:contact>
      <aster:license>CC BY-NC-SA 3.0</aster:license>
    </rdf:Description>

  </rdf:Description>
</rdf:RDF>
```
The “creator” of
http://geoserver.itc.nl/aster/Ontology/EnviromentalGeospatialServicesOntology.owl is “Aster Denekew”.

The corresponding RDF code is:

```xml
<rdf:RDF
    xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
    xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#
    xmlns:dc = "http://purl.org/dc/elements/1.1/"
    <rdf:description>
        <dc:creator>Aster Denekew</dc:creator>
        </rdf:description>
</rdf:RDF>
```

The metadata elements such as the creator or author and date of creation of a resource are defined in Dublin Core (DC) metadata [52] in which most common metadata elements are defined. In the example above, `dc:creator` is a property which is defined in the Dublin Core identified using the namespace declared using the prefix ‘dc’. RDF schema does not have to define them for use in RDF but can borrow them from DC. The first three lines inside the `rdf:RDF` tags are declarations for the XML namespaces to be used in this RDF file to identify the elements or syntaxes in the namespaces rdf, rdfs and dc respectively. The statement inside the `rdf:description` tag describe about the resource Environmental Geospatial Services Ontology.

A set of RDF statements uses a particular vocabulary that defines the properties and data types that are meaningful for the application at hand. Such an RDF vocabulary can be defined by using RDF Schema (RDF-S). As part of its schema language, RDF-S also defines some predefined concepts, including primitives to model a class/subclass hierarchy, relationships between classes ("properties"), and domain and range restrictions on such properties [54].

For example, `rdfs:subclassof` property is used to build class hierarchies and `rdfs:subpropertiesof` used to allow properties to be specialized. The following RDF triple statement shows rdfs:subclassof property used to express that `Grid_FeatureType` is subclass of `FeatureType`. 
A class defines a group of individuals that belong together because they share some properties. A class in RDF Schema corresponds to the generic concept of a Type or Category, in other languages. In RDF Schema, a class is any resource having an rdf:type property whose value is the resource rdfs:Class [55]. Classes can be organized in a specialization hierarchy using subClassOf.

Class hierarchies may be created by making one or more statements that a class is a subclass of another class. For example, Vector_FeatureType is subclass of FeatureType. Reasoners can thus infer an instance which is Vector_FeatureType is also a Feature_Type. The corresponding code is shown below.

```xml
<rdfs:Class rdf:about="Vector_FeatureType">
  <rdfs:subClassOf>
    <rdfs:Class rdf:ID="Feature_Type" />
  </rdfs:subClassOf>
</rdfs:Class>
```

Properties can be used to state relationships between individuals or from individuals to data values. For Example: FeatureType class has a property hasAttribute which relates it with the Attribute class; Attribute class has a property hasName, which relates it with an instance of the datatype: string.

rdfs:domain: A domain of a property limits the individuals to which the property can be applied.

rdfs:range: The range of a property limits the individuals that the property may have as its value.
For example, in the following code the RDF syntax is defined:

Statement is first defined as a class (line 13-18), which represents class of statements, and it is a sub class of resource itself. The subject is defined as a property which has as its domain “Statement” and as its range “resource” (line 19-25).

```xml
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:dc="http://purl.org/dc/elements/1.1/">
  <owl:Ontology rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <dc:title>The RDF Vocabulary (RDF)</dc:title>
    <dc:description>This is the RDF Schema for the RDF vocabulary defined in the RDF namespace.</dc:description>
  </owl:Ontology>
  <rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement">
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <rdfs:label>Statement</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment>The class of RDF statements.</rdfs:comment>
  </rdfs:Class>
  <rdfs:Property rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject">
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <rdfs:label>subject</rdfs:label>
    <rdfs:comment>The subject of the subject RDF statement.</rdfs:comment>
    <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdfs:Property>
  ...
</rdf:RDF>
```
3.5.3 Web Ontology Language (OWL)

Web Ontology Language (OWL) is an ontology language for making ontological statements, developed as a follow-on from RDF and RDFS [56]. The Semantic Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for the Semantic Web is an ontology language which can formally describe the meaning of terminology used in Web resources. For the machines to perform reasoning on the basis of these terminologies, the language must be descriptive, which go beyond the semantics of RDF schema. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDFS) by providing additional vocabulary along with a formal semantics. OWL adds more vocabulary for describing properties and classes, among others, relations between classes (e.g. disjointness), cardinality, equivalence, richer typing of properties, characteristics of properties (e.g. functional properties and inverse relations), and enumerated classes [57]. OWL is a further specialization of RDF [24].

OWL ontologies are categorized into three sub-languages: OWL-Lite, OWL-DL and OWL-Full. Defining feature of each sub-language is its expressiveness [58].

**OWL-Lite** is syntactically simple and the least expressive. OWL-DL and OWL-Full are subsequent extensions. **OWL-Full** is the most expressive.

**OWL DL** is the “description logic part” of the OWL language. Description Logics are a decidable fragment of First Order Logics and are therefore subject to automated reasoning. It is therefore possible to automatically compute the classification hierarchy, also called inferred hierarchy, and check for inconsistencies in the ontology that conforms to OWL-DL.

3.5.2.1 Components of an OWL Ontology

An OWL ontology consists of three components called Individuals, Properties, and Classes [58].

**Individuals** also known as instances, represent objects in the domain we are interested in. They can be referred as being instances of classes.

**Properties** are binary relations between individuals. Properties link two individuals of two classes. They are also known as roles in description logic.
**Classes** are a concrete representation of concepts. OWL classes are sets that contain individuals.

### 3.5.2.2 Subsumption, Disjointness and Equivalence

**SubclassOf**

Subclasses are specializations of their super classes. In OWL there is a built-in most general class named *Thing* that is the class of all individuals and is a super class of all OWL classes. There is also a built-in most specific class named *Nothing* that is the class that has no instances and a subclass of all OWL classes. An empty ontology contains the named class *owl:Thing*. All classes created are subclass of *owl:Thing*.

OWL allows general expressions to be used in axioms. “B is a subclass of A” written as $B \subseteq A$ means “B implies A” emphasizes the meaning of subsumption, ‘B is subsumed by A’ written $A \supseteq B$. Subsumption in OWL means necessary implication [59].

In our research example, *Surface_Water* subsumes *Flood*. I.e. *Flood* “is a subclass of” *Surface Water*. 
Figure 3-1: Subsumption (Subclassof) relationship for Flood concept in the Environmental_Domain

The above figure shows that:

Flood is subsumed by Hydrological_Hazards (Flood $\sqsubseteq$ Hydrological_Hazards)
Flood is subsumed by Surface_Water (Flood $\sqsubseteq$ Surface_Water)
Surface_Water is subsumed by Hydrosphere (Surface_Water $\sqsubseteq$ Hydrosphere)
Hydrological_Hazards is subsumed by Natural_Hazards (Hydrological_Hazards $\sqsubseteq$ Natural_Hazards)

Therefore it can be inferred that Flood is an indirect subclass of both Hydrosphere and Natural_Hazard.

DisjointWith

In OWL, classes are assumed to be overlapping until disjointness axioms are entered. In order to separate the classes we must make them disjoint from one another. In this way we ensure
that an individual which has been asserted to be a member of one class can not be a member of another class [58].

For example, an individual in a Vector_FeatureType can not be a member of Grid_FeatureType. And individuals of a Map_Access_Service can not be individuals of Catalogue_Service class. The Taxonomy of the Service is shown in the figure indicating that Map_Access_Service is disjoint with the other classes.

**Figure 3-2: Class Hierarchy of Services showing disjointness**

**EquivalentClass**

Two classes C and D are said to be equivalent classes if all the statement that define C also defines D. Concept C is equivalent with concept D if C is subsumed by D and D is subsumed by C;

i.e. $C \equiv D$ means $C \subseteq D$ and $D \subseteq C$.

**3.5.2.3 Properties and Restrictions**

**OWL Properties**

OWL Properties represent relationships between two individuals of different classes. There are two main types of properties; Object properties and Data type properties. Object properties link individuals of two classes. Data type properties link an individual to an XML Schema Data type value or an rdf literal [58].
Object properties: Object properties link individuals of one class (which is domain to the property) to individuals of another class (which is range of the property).

This can be expressed in DL as:

\[ \forall R(a,b) \quad \text{Where } R \text{ is a role, and } a \text{ and } b \text{ are individuals} \]
\[ \text{rdfs:range } C(b) \quad \text{Where } C \text{ is a class (which is the range of the property)} \]
\[ \text{rdfs:domain } D(a) \quad \text{Where } D \text{ is a class (which is the domain of the property)} \]

For example, the following Object properties are defined in our knowledge base.

has Attribute is an Object property whose domain is Vector_FeatureType, and whose range is Attribute class.

Datatype properties: Data type properties link an individual of one class to a value of an XML Schema Data type or rdf literal

The expression in DL is:

\[ \forall R(a,b) \quad \text{Where } R \text{ is a role, and } a \text{ and } b \text{ are individuals} \]
\[ \text{rdfs:range } C(b) \quad \text{Where } C \text{ is an XML Schema datatypve or rdf literal} \]
\[ \text{rdfs:domain } D(a) \quad \text{Where } D \text{ is a class (which is the domain of the property)} \]

For example, Attribute class has a data type property, called hasName which associates its individuals to the XML data type: string. Both owl:ObjectProperty and owl:DatatypeProperty are subclasses of the RDF class rdf:Property.

OWL Restrictions

The purpose of OWL is not just to create a concept hierarchy but to describe and define concepts. Restrictions on properties define the concepts clearly. OWL allows concepts to be defined by sets of necessary conditions.

In OWL properties are used to create restrictions, to restrict the individuals that belong to a class. There are three main categories of restrictions in OWL; Quantifier Restrictions, Cardinality Restrictions, and hasValue Restrictions [58].
Quantifier Restrictions: use two quantifiers the existential quantifier and the universal quantifier. An existential restriction also known as ‘SomeValueFrom’ denoted by (∃) specifies the existence of at least one relationship along a given property to an individual that is a member of a specific class. A However, existential restrictions do not mandate that the only relationships for the given property that can exist must be to individuals that are members of the specified class. Universal restrictions also known as “AllValueFrom” and denoted by (∀), describes the set of individuals that, for a given property, only have relationships to other individuals that are members of a specific class.

Quantifier restrictions consist of three parts: a quantifier, either ∃ or ∀; a property and a filler, that is a class description.

For example, Service has property hasServiceType, and a restriction is defined for this property as follows.

∀ hasServiceType.Service_Type

This Universal restriction restricts the individuals of a Service class to have a relationship for its property hasServiceType only to the individuals of Service_Type class, if it has this relationship.

Cardinality Restrictions: used to describe the class of individuals that have at least, at most or exactly a specified number of relationships with other individuals or datatype values.

Owl:minCardinality: A minimum cardinality restriction specifies the minimum number of relationships that an individual must participate in. The symbol used for the minimum cardinality restriction is “≥”.

For example: The Owl:mincardinality restriction restricts the individuals of Service class to have at least one individual from the Feature_Type class.

∀ hasFeatureType.Feature_Type

≥ 1 hasFeatureType

The first line restricts that the individuals of Service class to have a relationship along the hasFeatureType property only with the individuals of Feature_Type (if they have one). The
second line imposes the minimum cardinality restriction, which states that Service should have at least one Feature_Type.

**Owl:maxCardinality**: A maximum cardinality restriction specifies the maximum number of relationships that the individual can participate. The symbol used for maximum cardinality restriction is “≤”.

**Owl:Cardinality**: A cardinality restriction specifies the exact number of relationships that an individual participate and uses the equal sign “=”.

For example: *Owl: cardinality* specifies the exact number of relationships that the individuals of Service class can have with the individuals of CI_ResponsibleParty, for its hasprovider property (that means, services should have only one service provider or responsible party).

\[ \forall \text{hasProvider}.\text{iso19115:CI_ResponsibleParty} = 1 \text{hasProvider} \]

**hasValue Restrictions**: A hasValue restriction, denoted by the symbol \( \exists \), describes the set of individuals that have at least one relationship along the specified property to a specific individual.

In OWL, individuals can also be used in class descriptions, in enumerated classes which will have the individuals as values of the class. Here the BindingMethod class is modelled to have the binding methods for instance HTTPGet, HTTPPost, as individuals. Therefore the hasBindingMethod property restricted to have values from the individuals of BindingMethod.

E.g. *hasBindingMethod \( \exists \) HTTPGet* is hasValue restriction, where *hasBindingMethod* is a property and *HTTPGet* is an individual of *BindingMethod* class.

**Domain and Range**

In OWL domain and range are not constraints to be checked, rather they are axioms and can be used in reasoning [58].

For example, in the working knowledge base, the Service concept has the property *hasFeatureType* and the following necessary condition and restriction on the property:
∀ hasFeatureType.Feature_Type

Hence, the ‘hasFeatureType’ property has Service class as its domain and Feature_Type class as its range. I.e. hasFeatureType property links individuals of Service class with individuals of the class FeatureType.

3.5.2.4 Namespaces and Namespace Prefixes

A namespace is a string of characters used as a prefix to represent for the class, property or individual identifiers in an ontology. Every ontology has its own namespace, known as default namespace. An ontology may also use other namespaces. By maintaining different namespaces for different ontologies it is possible for one ontology to reference classes, properties and individuals in another ontology with out ambiguity and naming clashes. To ensure that namespaces are unique, URIs ending with “/” or “#” are used [58].

For instance, the default namespace of our ontology, developed in this research, is http://geoserver.itc.nl/Aster/Ontology/EnvironmentalGeospatialServices.owl#

The ISO 19115 ontology imported to our ontology has namespace

http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.owl# (found in the web page: http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.htm)

Common namespaces that are imported and used in the ontology, i.e. automatically created by the ontology editor when a new ontology file is created are:

- rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns# - The Resource Description Framework namespace
- rdfs:http://www.w3.org/2000/01/rdf-schema# - The RDF-Schema namespace
- owl:http://www.w3.org/2002/07/owl# - The OWL vocabulary namespace
- xsd:http://www.w3.org/2001/XMLSchema# - The XML-Schema namespace

Namespace prefixes are short strings used to represent the full namespace. In order to reference classes, properties or individuals in an ontology, we prefix the identifiers with the namespace prefix and a colon.
E.g. in “iso19115:MD_Metadata”; the prefix “iso19115” indicates the ISO ontology and replaces the full namespace name “http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.owl#”.

3.6 Concluding Remarks

Interoperability is the basis for facilitating data sharing. Data sharing between different information communities needs a common understanding of the semantics of the data, in the specific domain. Therefore, Semantic interoperability requires formally defined concepts and terms. Ontology defines a common vocabulary for people who need to share information in a domain.

RDF, RDFS and OWL are Knowledge Representation formalisms to give machine-interpretable definitions of basic concepts in a domain and relations between them in an ontology. Therefore, the ontology can be used for “semantic annotation” of services, which attach meanings to the resources based on the concepts defined in the ontology, so that human users and machines can better understand and reason with it.
4 Implementation

This research focuses at semantic description of geospatial web services based on the meaning of concepts developed in ontologies. It aims at providing user access to these semantic web services through geospatial portals to facilitate data sharing, solving semantic problems. To achieve our objectives, first we have chosen a domain for which we use the application. We have developed a domain ontology for the concepts in this domain. An application ontology was created and integrated with the domain ontology to serve our purpose, semantic description of services. The domain ontology supplies meaning to the concepts used in the application ontology. OGC Web Map Services are implemented for five environmental themes for the application we are interested in. The ontology is then used for semantic annotation in the service descriptions to attach meanings for their data content. Finally, the services along with their service descriptions as well as the ontology are made available for the user in the geospatial portal. A user can find a certain concept based on the domain ontology and have understanding how its meaning is restricted to the application.

4.1 Use case

The African Environment Information Network is selected as a use case to show solving one of the geospatial data sharing problems in information infrastructures, specifically semantic interoperability problem. The African Environment Information Network initiative is a regional initiative to exchange environmental information within the nation and across nations in the continent. In this research we focus on the geospatial information component which will be one of the core components of the environmental information. The research tries to solve the semantic problems arising from geospatial information sharing among information providers and the users.

As a component of the environmental information, geospatial information will be crucial for environmental monitoring and in the management of environmental assets. In our use case, we considered that Environmental Information Network working groups therefore need to facilitate the sharing of environmental geospatial information within organizations and to disseminate to users. One possible way to share and disseminate information is to use the state of the art technology, Web Services, through a geo portal which gives access for the users to the available geospatial data and services.
One problem in sharing the data and services is the problem of semantic interoperability, i.e. difference in meaning of the concepts in the domain used and understood by the information providers and users. To solve this problem, an ontology is developed to define the common terminology and concepts used in the domain and its application. This ontology will be the basis to describe the data and services being shared. The user can refer to the ontology for meanings and concepts used in the data and services to check if it matches with what he wants to acquire. One of the objectives of this research is to design and implement a prototype ontology based geo portal for discovery and access of semantic geospatial data and services.

The use case is described in detail in Appendix A with example. However, the application varies as the purpose and interest of individual users. We are interested in the technology which facilitates the sharing of geospatial information in any information infrastructures. The result of this research would also be applicable for other information sharing initiatives which will share the same infrastructure such as National Spatial Data Infrastructures.

The main actor in the use case is the user. Imagine a simple scenario that a user, Mr. X, wants to apply for a project and needs to know about the area he is interested in. He wants to know if it is accessible, it is not a protected area and wants to analyze other environmental factors. The information he needs, for example, Infrastructure and protected areas, are physically in different places. These information sources are made available as Web Services hosted by different institutions. The portal however, provides access to the data and services from a single entry point. The Use Case diagram in Figure A-1 (Appendix A) shows the role of data providers and users in the geo portal.

The following sequence diagram shows how the user access services in the portal.
Ontology Based Portals for Accessing Geospatial Data and Web Services: Solving Semantic Problem in Data Sharing

Figure 4-1: Sequence Diagram to access geospatial services through Geospatial portal

Providers can also register their geospatial data and services to catalogue services that give users a possibility to search and access to these information. The use case diagram in Figure A-3 (Appendix A) shows how the user can use catalogue services to access geospatial data and services.

4.2 Development of Ontology

The development of Ontology has different phases. The first phase is conceptualization of the terms and concepts used in the specific application domain. This conceptualization is then converted into knowledge representation formalism, by explicitly defining the relevant concepts of the domain (its terminology) and then using these concepts to specify properties of objects and individuals occurring in the domain.

4.2.1 Taxonomies

Taxonomies are hierarchical classifications of objects. Taxonomy defines classes of objects and their relationship among them in a form of tree like structure. Taxonomies are hierarchical and relationships are inherited down the hierarchy. Ontologies are analogous to taxonomies in a sense that they classify concepts and their relationships in a more structured way and formalism. In this research, the development of ontology was based on existing taxonomies and datasets available at hand for the specific application. The domain ontology was created based on the taxonomy of the Global Change Master Directory (GCMD) Earth Science keywords which is developed and maintained by NASA.
GCMD Earth Science keywords taxonomy includes many topics which are related to Environmental applications. Major Topics in GCMD Earth Science keywords are:

- Agriculture
- Atmosphere
- Biosphere
- Hydrosphere
- Human Dimension
- Land Surface
- Solid Earth

These topics are further classified in terms and the terms in different variables. E.g. Flood is classified under surface water (term) which is classified under the Hydrosphere (topic). Therefore, this taxonomy is selected to define the concepts for the domain ontology, because it includes the concepts which relates to the themes we are interested in for our application.

4.2.2 Datasets Used

Most of the datasets used for this research are from the USGS Global GIS Coverage datasets. Datasets for Africa coverage have been extracted from this global coverage datasets for our research purpose. The associated metadata for these datasets are based on the FGDC (Federal Geographic Data Committee) metadata content standard. However, few datasets are from another source; Lakes and Protected Areas as their metadata indicates in the ontology. The metadata elements in FGDC format are mapped into ISO 19115 elements in the development of the ontology.

4.2.3 Tools Used

An ontology editor, Protégé, is used for the development of the ontology. Protégé is an open platform for ontology modelling and knowledge acquisition. One of the major benefits of using Protégé is its open architecture. The system provides various mechanisms to incorporate custom-tailored extensions into it, so that external components like reasoners can be integrated easily. Another advantage of protégé is existing ontologies could be imported into the project.

The language we used is Web Ontology Language (OWL) in Protégé. Protégé provides an “OWL Plug-in”, a Semantic Web extension of the Protégé ontology development platform. The “OWL Plug-in” can be used to edit ontologies in the OWL language, to access
description logic reasoners, and to acquire instances for semantic mark-up. The “OWL Plug-in” provides direct access to DL reasoners such as RACER. RACER is used here to check the consistency of the ontology. OWL has three options OWL Lite, OWL Full and OWL DL. Here OWL DL is used, because it is more expressive and based on description logic, it is possible to use the reasonors to compute classification hierarchy and check for inconsistencies in the ontology.

4.2.4 Concepts of the ontology

Initially, a domain ontology is developed that represents the real world situation in the specific domain: environmental domain. The concepts of the domain ontology are based on the GCMD Earth Science keywords. The GCMD is selected as a framework, because it includes the concepts which relates to the themes we are interested in for our application in the use case. A domain ontology relates concepts to each other, and thus restricts their interpretation. However, not all users have the same understanding of a certain concept. The concepts used in this ontology will be restricted with respect to environmental applications, though these concepts may have more meanings in different contexts.

Another domain ontology is Service Ontology which refers to the domain of services, based on ISO 19119 service taxonomy. Once the domain concepts are identified and defined, an application ontology can be built for the concepts related to the application in the domain. An application ontology was created for concepts related to the specific application, geospatial web services in an environmental domain, and based on the data at hand which uses the concepts from the domain ontology.

Practical development of ontology includes [44]:

- Defining concepts in the ontology (Concepts are also known as classes, objects and categories in some languages);
- Arranging the classes in a taxonomic hierarchy (called “is-a” hierarchy);
- Defining properties of classes (also referred as roles in DL) and describing allowed values for these properties;
- Create a knowledge base by defining individuals, which are instances of the classes, by filling in the values for their properties and additional restrictions.

The main concept of the environmental domain ontology, *Environmental_Domain* (class) is defined and subclasses for the domain category are defined from topics, terms and variables in the GCMD taxonomy.
Figure 4-2 shows the *Environmental_Domain* class hierarchy. Here we can see the *Flood* class as it is subsumed by (subclassof) *Hydrological_Hazards* and *Surface_Water*. That means Flood is surface water and Flood is also Hydrological Hazard. The purpose of ontology is to model such concepts.

The ontology developed integrates the domain ontology and the application ontology into single ontology. The major concepts (classes) of the ontology are:

- *Environmentenental_Domain*
- *Services*
- *Feature_Type*
Feature_Type concept and its subconcepts, Vector_FeatureType and Grid_FeatureType, are defined based on the datasets available for the application. Feature_Type concept is then associated with Environmental_Domain concept with the property called hasDomainCategory. I.e. Feature Types are associated with the domain ontology according to their category in the domain.

Capabilities of Web services correspond to the functionalities provided by Web services. One approach to express functionalities is to provide an ontology of functions where each class in the ontology corresponds to a class of homogeneous functionalities [32]. The services can be categorized based on their functionality and service type in the application ontology. In this research the ISO 19119 service taxonomy is used to define the Service concept, which represents OGC Web Services, and to classify these services based on their functionalities in the ontology. This research has implemented OGC Web Services – specifically Web Map Services (are Map_Access_Services in the ontology) which are under the categories of Geographic model/information management services in the service taxonomy. However, the ontology implements the full service taxonomy considering for future extendibility. The service taxonomy in our ontology which is expressed in DL is shown in Figure 4-3.

Figure 4-3: Service Taxonomy in the Protege ontology editor window

An ontology is reusable and sharable in its nature. When building an ontology, a portion of another ontology can be incorporated in it and in the semantic web context, it is necessary to
make the metadata interoperable, which requires merging of different ontologies [49]. The ISO Metadata ontology is imported into our ontology to reuse ISO19115 Metadata elements. The Metadata for our datasets are in FGDC format, and they are mapped to ISO 19115 Metadata elements inorder to use these imported elements. Then, the Feature_Type concept is associated with iso19115:MD_Metadata concept in the imported ISO19115 ontology using hasMetadata property inorder to associate feature types with their metadata. The prefix iso19115 is used to indicate concepts of the ISO19115 Ontology. The application ontology also contains other concepts which are related to the Service and Feature_Type concepts. These are Attribute, Binding, BindingMethod, Port, PortType, Operation, and OperationParameters.

After defining concepts and properties, the next step is to add individuals for the classes, and to fill the values for their properties. An OWL ontology allows defining individuals and asserting properties about them. Individuals can also be used in class descriptions, in enumerated classes which will have the individuals as values of the class. For example: *Africa_Climate_Web_Map_Service* is an instance of *Map_Access_Service* class. Figure 4-4 shows the instances of *Map_Access_Service* in our knowledge base.

![Figure 4-4: Instances of Map_Access_Service shown in the instance editor window of Protégé](image)

Figure 4-5 shows the service instances of *Map_Access_Service* in the left side and the properties and corresponding values of the selected instance in the right side of the instance browser window of protégé ontology editor.
The following piece of RDF code from the OWL file, shows Africa_Climate_Web_Map_Service, an instance of Map_Access_Service: its properties and their values.

```
<Map_Access_Service rdf:ID="Africa_Climate_Web_Map_Service">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >
    Africa climate Web Map Service contains features: ecological region, rain, temperature, precipitation, and evapotranspiration. This service is created using ArcIMS 9.0 and compatible with OGC Web Map Service standards. </rdfs:comment>
  <hasProvider rdf:resource="#CI_ResponsibleParty_WMS"/>
  <hasServiceType rdf:resource="#WMS"/>
  <hasFeatureType rdf:resource="#Africa_EcologicalRegions"/>
  <hasFeatureType rdf:resource="#Africa_PotentialEvapotranspiration"/>
  <hasFeatureType rdf:resource="#Africa_MeanTemprature"/>
  <hasFeatureType rdf:resource="#Africa_Boundaries"/>
  <hasFeatureType rdf:resource="#Africa_Precipitation"/>
  <hasFeatureType rdf:resource="#AfricaRain"/>
  <hasFeatureType rdf:resource="#Africa_ActualEvapotranspiration"/>
  <hasBinding rdf:resource="#Africa_Climate_WMS_Access_Point"/>
  <hasService_Metadata rdf:resource="#Service_Metadata_Africa_Climate_WMS"/>
</Map_Access_Service>
```
Africa_Climate_Web_Map_Service has service metadata: Service_Metadata_Africa_Climate_WMS. The following piece of RDF code from the OWL file describes its Service_Metadata property:

```
<Service_Metadata>
  <hasService_Identification>
    <iso19115:MD_ServiceIdentification rdf:ID="MD_ServiceIdentification_Africa_Climate_WMS">
      <iso19115:abstract rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The services are created using environmental datasets to facilitate the sharing of data and services in a distributed environment
      </iso19115:abstract>
      <iso19115:pointOfContact rdf:resource="#CI_ResponsibleParty_WMS"/>
      <iso19115:purpose rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Research: Semantic description of Web Services
      </iso19115:purpose>
    </iso19115:MD_ServiceIdentification>
  </hasService_Identification>
  <hasOperation_Metadata rdf:resource="#GetMap_Africa_Climate_WMS"/>
  <hasOperation_Metadata rdf:resource="#GetCapabilities_Africa_Climate_WMS"/>
  <GetCapabilities rdf:ID="GetCapabilities_Africa_Climate_WMS">
    <hasBinding>
      <Binding rdf:ID="Africa_Climate_Capabilities_Request">
        <hasBindingMethod rdf:resource="#HTTPGet"/>
        <hasLocation rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        </hasLocation>
        <hasPorttype rdf:resource="#HTTP"/>
      </Binding>
    </hasBinding>
    <hasData_Identification>
      <EcologicalRegion rdf:ID="Africa_EcologicalRegions">
        <hasSpatial_Representaion_Form rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-19115#vector"/>
      </EcologicalRegion>
    </hasData_Identification>
  </GetCapabilities>
</Service_Metadata>
```
Line 2-12 is service identification that describes the service using ISO 19115 metadata elements for the abstract, point of contact and purpose. Line 13 and 14 defines the operations metadata, GetMap and GetCapabilities operations. Line 15 to 26 describes the GetCapabilities operations defined in line 14. Line 27-46 is Data Identification part of the metadata. It describes one of the layers of the service EcologicalRegion. Line 29-30 states its spatial representation form is Vector Feature type. Line 31 is associating the concept with its domain category. Line 32-33 states its geometric type is Surface (polygon) feature type. Line 34 gives its spatial reference system: WGS84. The lines 35-42 lists its attributes.
Africa_Climate_Web_Map_Service has a property, hasbinding that indicates the access point for the service instance in the server. This is shown in the following code.

```xml
<Binding rdf:ID="Africa_Climate_WMS_Access_Point">
  <hasPorttype rdf:resource="#HTTP"/>
  <hasLocation rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >~@anyURI
  </hasLocation>
  <hasBindingMethod rdf:resource="#HTTPGet"/>
</Binding>
```

### 4.3 Implementing OGC Web Map Services (WMS)

One of the objectives of this research is the semantic description of geospatial web services based on the meaning and concepts developed in ontologies. To achieve this objective, OGC Web Map Service instances are implemented for environmental themes. The OGC Web Map Server Interface Specification (WMS) defines a set of interface specifications that provide uniform access by Web clients to maps rendered by map servers on the Internet. WMS is a service interface specification that:

- Enables the dynamic construction of a map as a picture, as a series of graphical elements, or as a packaged set of geographic feature data.
- Answers basic queries about the content of the map.
- Informs about the maps it can produce and which of those can be queried further.

The software tool used to implement these WMS service instances is ArcIMS. ArcIMS WMS are created for five themes: Hydrology, Geology, Climate, Infrastructure and Protected Areas. This OGC standard Web Map Services (WMS) can be integrated with any other WMS implementation.

The OGC Web Map Services supports two mandatory requests GetCapabilities and GetMap and an optional GetFeatureInfo request.

A Web Map Service shall support the "GET" method of the HTTP protocol (IETF RFC 2616). An Online Resource URL intended for HTTP GET requests is a URL prefix to which additional parameters are appended in order to construct a valid Operation request. A URL prefix is defined in accordance with IETF RFC 2396 as a string including, in order, the
Internet Protocol prefixes ("http" or "https"), the hostname or numeric address, optional port number, path, mandatory question mark '?', and optional string comprising one or more server-specific parameters ending in an ampersand '&'. For example the following URL prefix is used for a GetCapabilities request (to request for the capabilities of the service instance Africa_Climate_Map_Service).


When the HTTP GET method is used, the client-constructed query part is appended to the URL prefix defined by the server (the string before the “?”), and the resulting complete URL is invoked as defined by HTTP. The user request should include the mandatory parameters: SERVICE and REQUEST for example in the GetCapabilities request. The Service then appends these parameters in the above URL prefix. The user request will look like this:


The response to this GetCapabilities request is the Capabilities XML document which describes the service, its content and the supported operations. The capabilities file for the services are shown in Appendix D.

The GetMap operation of WMS returns a map for a specific request from the client. When requesting a map the URL indicates what information is to be shown on the map, the geographic extent of the map, the desired coordinate reference system, and the output image width and height. For example, the GetMap request to the Africa_ProtectedAreas_Web_Map_Service is:


The GetMap response returns the map for the client in the format requested in this GetMap request.

The proprietary implementation of ArcIMS supports client viewers, JAVA and HTML, to give access for the services. The ArcIMS Designer is used to create website in the geoserver
which gives access to the Web Map Services with its HTML client viewer. The user can get access to the services by typing the website address. For example, the Africa_Geology_Web_Map_Service can be accessed using the address:

http://geoserver.itc.nl/website/gfm2_02/Africa_geology_Web_Map_Service/viewer.htm

4.4 Description of Web Services

Semantic Web technologies along with ontology languages such as OWL are being used to add meaning to web content by annotating the data and services, with the concepts in the ontology. This allows users to get an understanding of the web content and helps them for meaningful web searches [60].

Semantic mark-up of Web Services has been proposed by different projects. The proposals include OWL-S from OWL Services Coalition (a group of Semantic Web researchers), and WSDL-S from IBM and University of Georgia. The OWL-S (formerly DAML-S) project defines an ontology for the domain of Web services. This ontology provides tags which can be used for describing Web services. The ontology consists of three sub ontologies, service profile, service grounding and the process model, which are tied together using a service ontology. The profile is used to express “what a service does”, for purposes of advertising, constructing service requests, and matchmaking; Service grounding contains information about invocation. The process model describes the operations of the service [32], [60]. IBM and University of Georgia comes with the WSDL-S proposal aiming to provide an extension to WSDL by adding semantics. WSDL-S refers to the OWL-S profile model (component of OWL-S that describes functionality of Web services) [60].

The same concept as WSDL-S is used here to add semantics to the capabilities document of the OGC web services, extending it by semantic annotation elements. The GetCapabilities request to a WMS returns a capabilities document which describes the service, its capabilities and content. Here the ontology developed (the semantic model) is used for semantic annotation of the OGC Web Map Services, to add meaning to the content of the services, in their capabilities document which in turn increases users understanding for the data and capabilities available in the service. Semantic annotation is additional information in a document that defines the semantics of a part of that document. The semantic annotations are additional XML elements in the service description. They define semantics by referring to a part of a semantic model that describes the semantics of part of the document being annotated [61].
The WMS capabilities document can be edited using any XML editor. We used XMLSpy to edit the capabilities XML files and its associated DTD. OGC uses DTD to define the elements and attributes used in Web Map Services Capabilities documents. The OGC DTD located in “http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd” is modified locally to include the element <Semantics> which is introduced here for semantic reference and saved in the geoserver. Each layer of the services are annotated using the element tag <Semantics> </Semantics> in the capabilities document to describe the data content referencing the semantics used from the ontology.

The semantic annotation is done in the following steps:

1. The <Semantics> element and its attributes are added in the DTD

   <!-- An ontology reference to the resources in the online ontology. -->
   <!ELEMENT Semantics EMPTY>
   <!ATTLIST Semantics
       xmlns:rdf CDATA #FIXED "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs CDATA #FIXED "http://www.w3.org/2000/01/rdf-schema#"
       xmlns:owl CDATA #FIXED "http://www.w3.org/2002/07/owl#"
       xmlns:Sref CDATA #FIXED "http://www.ibm.com/xmlns/WebServices/WSSemantics"
       xmlns:Ontology CDATA #FIXED "http://geoserver.itc.nl/aster/ontology/EnvironmentalServicesOntology.owl#"
       Sref:RefersTo CDATA #REQUIRED >

2. The XML namespaces are declared globally in the capabilities document as follows:

   <WMT_MS_Capabilities version="1.1.1"
       xmlns:xlink="http://www.w3.org/1999/xlink"
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
       xmlns:owl="http://www.w3.org/2002/07/owl#"
       xmlns:Ontology="http://geoserver.itc.nl/aster/ontology/EnvironmentalServicesOntology.owl#"
       xmlns:Sref="http://www.ibm.com/xmlns/WebServices/WSSemantics">
   ...

   </WMT_MS_Capabilities>
We used namespaces to reference classes and properties from the ontology. The XML namespace prefixes are used to represent the namespaces. Sref represents the namespace "http://www.ibm.com/xmlns/WebServices/WSSemantics" and it is used to acquire elements in the WSSemantics namespace for semantic annotation [31, 62]. The namespace prefix Ontology is used to represent our ontology which refers to the concepts we need to use for the semantic annotation.

3. Each layer is annotated using the <Semantics> element.

```xml
<Semantics Sref:RefersTo="Ontology#Africa_ProtectedAreas"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:Sref="http://www.ibm.com/xmlns/WebServices/WSSemantics"
xmlns:Ontology="http://geoserver.itc.nl/aster/ontology/EnvironmentalServicesOntology.owl#" />
```

Here the Semantics element is referring to the concept Africa_ProtectedAreas in the ontology.

As the Web Map Service implementation specification states a Map Server should use one or more <Metadata> elements to offer detailed and standardized metadata about the layers in the service [38]. <MetadataURL> element is used in the DTD to add the metadata of the datasets. This element is used in the capabilities file to indicate the layers’ metadata. The type attribute in the <MetadataURL> element specifies the metadata standard used. Our datasets have FGDC metadata standard which is mapped into ISO19115 in the ontology. The <Semantics> element is used here to give references to the ISO19115 metadata in the ontology. The OnlineResource element indicates the metadata access point.

```xml
<MetadataURL type="FGDC">
  <Format>FGDC</Format>
  <OnlineResource xlink:href="http://geoserver.itc.nl/Portal/Metadata/africa_ProtectedAreas.htm"
    xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" />
  <Semantics Sref:RefersTo="Ontology#MD_Metadata_Africa_ProtectedAreas"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```xml
```

69
This code is included to indicate the metadata of the *Africa_ProtectedAreas* layer and it is referring to the metadata of *Africa_ProtectedAreas* in the ontology.

The full Service Description and DTD are included in Appendix D.

### 4.5 Development of Geospatial Portal

Data and services become distributed across multiple servers in different organizations. These distributed services can be searched and accessed using different search engines and catalog services. However, currently these kind of search do not provide capability based search and do not provide the meaning of the data and services. Portals bring these distributed data and services together and make them available from a single entry point. In this research a geospatial portal is developed to give users access to distributed data and services along with the semantic reference.

The Geospatial portal implemented makes available the Web Map Services and the layers in each service, the service description (service metadata) to understand the service and its capabilities, and their semantics using the ontology.

Graphical representation is generated from protégé visualization plug-in, “OntoViz”, for the instances in the ontology. This graphical representation of the Ontology is used in the portal development to give meaningful information for the user and the graphics is used to link to the services and their service description as well as dataset metadata. Here an example in Figure 4-6 shows the graphical representation for the *Africa_ProtectedAreas/Web_Map_Service* instance used in the portal.
In this example, the *Africa_ProtectedAreas_WMS_Service* has a binding to access the service, and links to the Service to be displayed in the HTML client viewer. *HasServiceType* indicates its service type is *WMS*. *HasService_Metadata* links to the service description, which tells the capabilities of the services and their content which has semantic annotation to give reference to the ontology for concepts used. *HasFeatureType* indicates the layers of the service and links to the dataset metadata. These layers are linked to their metadata (FGDC format) as HTML files.

The HTML viewers used to display the services are customized to show their service metadata. The lower frame shows the service metadata to give reference to the semantics of the data content and their metadata. The purpose of the service description is not only meant to display it for the user. It can also be used in client viewers and catalogue services, for instance Gaia client viewer, an open geospatial client viewer developed by Carbon project, to extract the information from the service metadata and display it in its graphical interface.
Ontology Based Portals for Accessing Geospatial Data and Web Services: Solving Semantic Problem in Data Sharing

Figure 4-7: Africa_ProtectedAreas_Web_Map_Service displayed in HTML Viewer

Here the `<Semantics>` element refers to the `Africa_ProtectedAreas` in the ontology. The semantics used in the service description can be referred from the Ontology browser and OWL files accessed from the ontology page in the portal. Protégé generates ontology documents in the form of HTML pages (OWL Doc) from the ontology which provides the links between the elements of the ontology: concepts (classes), properties and instances. Because Protégé is still in its development stage, it has encountered some limitations. The documents for the instances could not be generated and are not included in the ontology browser. However, the concepts and their descriptions are rich in the ontology, which can be accessed in the ontology editor, Protégé itself. The user can download and browse through the ontology if needed. Figure 4-8 indicates the HTML ontology browser window for the classes and properties.
4.6 Evaluation

4.6.1 Evaluation of the ontology

As [44] explained, there is no one best way or methodology for developing ontologies. The best solution to model a domain depends on the application that you have in mind and the extension that you anticipate. However, it identified an iterative approach to ontology development as the best way and a methodology to evaluate ontology. What it suggests as the fundamental rules in ontology design is concepts in the ontology should be close to objects and relationships in your domain of interest. An ontology is a model of reality of the world and the concepts must reflect this reality.
Evaluating ontology therefore involves checking the applicability and usefulness by using it in the application or problem solving methods or by discussing it with experts in the field or both. We assess the quality of our ontology only by using it in the application for which we designed for. Tests with external expert users would have been a sound method to evaluate, but the time constraints prevented us from doing so. Therefore, we used the ontology to describe the services and checked if it is sufficient to describe the services. In the development process we have redesigned it iteratively to include all the concepts that are used for our application. And it was sufficient to describe the services and their data content by annotating them using the “Capabilities document”.

4.6.2 Evaluation of Web Services Implementation and Description

The OGC Web Map Services are implemented in ArcIMS. ArcIMS designer is used to create a website which displays the services in one of its client viewers, an HTML viewer. These services can be integrated with services in other client viewers or catalogue services and in specific systems like ArcMap and ArcExplorer.

The evaluation of the services is done by trying to access it from client viewers, and different systems. The service description can be used to give users semantically enriched services as well as software clients to extract meaningful information. We use Gaia client viewer to check if this is possible with current viewers. The Gaia client viewer can display the services with other services in any server registered to the client. However, it doesn’t have the element and capability to extract the semantic annotations used in the services.
Figure 4-9: Gaia Client Viewer displaying Africa_Geology_Web_Map_Service

Figure 4-10 also shows that *Africa_Geology_Web_Map_Service* integrated with other services in the Intergraph OGC WMS Viewer, developed by Open Geospatial Network, an open network which is aiming to give users access to global spatial data and services towards achieving Global Spatial Data Infrastructure. Our services can be integrated with other services in Intergraph OGC WMS Viewer; however, it doesn’t have a feature to extract the semantics from their service description.
The Web Map Services can be integrated in the users own system such as ArcMap or ArcExplorer with other local data. And it is proved that these semantic services are compatible with and can be integrated with other services.
4.6.3 Evaluation of the Geo portal

The Geo portal was checked by browsing through it to access the services and their description as well as the ontology to see if they help the user to get relevant information about the available services and data. It is clear that providing the semantics in the portal will give the users an understanding of the data and services and therefore, better understanding and access to them. The service metadata and the ontology pages gives the capabilities of the service and the reference to the semantics /meaning used, for the users to have the same understanding of the data and services with the provider. The services are accessed in HTML client viewer customized to show their service metadata which gives reference to the ontology for their usage. However, the semantic reference used in the service metadata is in XML format. A user for example a geoscientist, who is not experienced in XML, may have difficulty to read the capabilities file. However, help file, “how to use the ontology”, is included in the portal as a guide to allow the users how to read and understand these files. With the time limit working further on extracting these semantics in a graphical user interface was not achieved. It is proposed for the client viewers and catalogues to consider extracting this information to give the semantics along with other elements already available.
4.7 Concluding Remarks

The service description of OGC Web Services, capabilities document, can be edited to describe the OGC Web Map Services semantically. These capabilities document can be used to give users access to semantically enriched services as well as software clients to extract meaningful information.

This research aims at semantic description of geospatial web services, aiming to provide users access to these semantically described web services through geospatial portals. This is achieved by semantic annotation of the content of the services, in the capabilities document to refer to the concepts used in the ontology (our semantic model). Providing the semantics in the portal also gives users an understanding of the data and services and therefore, better understanding and access to them. However, the portal view for the capabilities document is in XML format. This semantic information can be extracted and used to give the information in a graphical user interface.
5 Conclusion and Recommendation

5.1 Conclusion

During the research, as learning process many concepts are gained about Knowledge representation. These theories and concepts are discussed in the previous chapters. RDF, RDFS and OWL are knowledge representation formalisms to give machine-interpretable definitions of basic concepts in a domain and relations between them. A knowledge representation based on Description Logics provides facilities to set up knowledge bases, to reason about their content. OWL DL is the “description logic part” of the OWL language. This research uses OWL DL for the ontology development, as it has expressive power and is subject to automated reasoning.

Ontology development process requires acquiring knowledge of the domain of application. The ontology developed in this research covers the environmental domain and its application in the web services domain. A domain ontology relates concepts to each other, and restricts their interpretation. The concepts used in our ontology will be restricted with respect to environmental applications, though these concepts may have more meanings in different contexts. However this can easily be extended to any other domain. The Ontology developed is used for semantic description of web services. Here, the semantic annotation used in the capabilities document specifically addresses the OGC Web Map Services semantic problem.

Currently, the geospatial standards developed by OGC gives abstract and implementation specifications for online mapping such as OGC Web Services, metadata and catalogue services. These are used as a basis to build distributed interoperable geospatial services accessible through the web by means of portals and catalogue services. However, these standards do not address the semantic aspects of distributed systems. OGC Web Map Services specifically, provide the self describing GetCapabilities operation, which specifies the capabilities and the data content of the services. This operation provides capability representation, which can improve interoperability; however, it does not provide the semantics of the data content and therefore is not sufficient to allow semantic interoperability.
Semantic interoperability needs a well established semantic reference. This research tries to achieve this by defining concepts in the domain of application. Such semantic problems can be addressed using the RDF and OWL, knowledge representation languages, which provide machine-interpretable definitions of concepts in a semantic model, an ontology.

OGC Web Map Services can be semantically enriched by referencing to the concepts used in the ontology giving meaning to the services. This research focused on semantic description of OGC Web Map Services based on the meaning and concepts in ontologies, and to provide users access to these semantic web services through geospatial portals. This is achieved by developing ontologies; and semantic annotation of the content of the services, in the capabilities document to refer to the concepts used in the ontology. The capabilities document can be used to give users access to semantically described services as well as software clients to extract meaningful information. A user can find a certain concept based on a domain ontology and will have better understanding how its meaning is restricted in the application. Providing the semantic reference in the portal gives users an understanding of the data and services and therefore, better understanding and access to them.

### 5.2 Recommendation and Future Work

Based on the results of the research the following recommendations are forwarded.

- The methodology used in this research to describe geospatial web services semantically is by means of ontologies. This semantically described web services and their semantic reference are also made available for human users through geo portal. Similar data sharing initiatives and information infrastructures such as National Spatial Data Infrastructure can benefit from this methodology, for solving semantic problems in data sharing.

- The OGC WMS capabilities document is associated with the standard WMS capabilities DTD which defines the elements and attributes used in WMS service descriptions. We recommended that a semantic tag to be included in the standard OGC WMS capabilities DTD or future schema, which can be used for semantic service descriptions.

- An interesting next step will be to search and access services in a semantically enriched way. Current client viewers and catalogue services do not benefit from such semantically described services. Therefore, we recommended that future work focuses
on customizing and developing clients and catalogue services which can extract information from these semantic annotations.
Bibliography


Appendix A. Use Case

A. 1. USE CASE: ENVIRONMENTAL INFORMATION NETWORK

The Use case used here is the case where geo portal (client interface) is used for facilitating access to geo data and Web services for environmental applications. The use case will have two levels, top level and sub levels.

Top Level Use Case:

Description of Use case

The use case selected here is African Environmental Information Network (AEIN). AEIN is a regional initiative to exchange environmental information with in the nation and across nations in the continent. The African Ministerial Conference on the Environment (AMCEN) has realized that getting information about the conditions of environmental assets in Africa is indispensable in monitoring the continent’s environment. Hence AMCEN believed that there is an urgent need to develop an integrated environmental assessment and reporting mechanism within the continent, which will generate and provide relevant environmental information to facilitate efficient monitoring of the environment. Hence, AEIN initiative was mandated by AMCEN, currently being coordinated by UNEP [7].

Subsequently, it became necessary to establish the National Working Groups to implement the African Environment Information Network initiative at a national level in the thirteen selected pilot countries of the region. One of these is Ethiopia. To mention an example for the national implementation of AEIN, Ethiopian Environment Information Network (Ethio-EIN), formed by coordinating 12 core institutions that are major producers and users of environmental data/information.

As a component of the environmental information, geospatial information will be crucial for environmental monitoring and in the management of environmental assets. The Environmental Information Network working groups therefore need to facilitate the sharing of environmental geospatial information with in the stakeholder organizations and to disseminate to external users.
One problem that will arise in sharing the geospatial information is the problem of semantic interoperability, i.e. difference in meaning of the concepts in the domain used and understood by different sectors or partner organizations and users. This research tries to show how to solve semantic problems arising from geospatial information sharing among information providers and the user community. Ontology is used in this research to define the common terminology and concepts used in the environmental domain. This ontology will be the basis to describe the datasets and the services.

A user from the stakeholder organization or external community may need to perform environmental impact assessment to analyze the environmental issues and impacts that may arise in a development project. This is a multi step process which also requires the actual analysis of the environmental information. The user then needs to know which data and services are available and needs to access the data and services provided.

The best possible way to share and disseminate information is to use the state of the art technology, creating a geo portal to give users access to the available geo data and Web services. The geo portal should give access to the geo data and services provided by the stakeholder organizations both to the partner organizations and external users. It also provides the description of the data and services, the metadata, so that the user should know if they are usable or not. The user can also refer to the ontology provided for the semantics (meaning and concepts) used in the data and services to check if it matches with what he wants to acquire.

The main actor in the use case is the user. For example: Imagine a user, Mr. X, wants to apply for a project and needs to know about the area he is interested. He needs to know if it is accessible, it is not protected area and analyze for other environmental factors. The information he needs for example, Infrastructure and protected areas are physically in different places, in different organizations. These information are made available as web services hosted by different organizations. The portal however, provides access to the data from a single point.

The following is a use case diagram for the scenario that providers make available their data and services in Geospatial portals.
Figure A-1: Top Level Use case diagram for Environmental Information Network (Scenario1 - providers make available their data and services in Geospatial portals)

The following sequence diagram shows how the user makes use of the portal to find and access services in the portal.

Figure A-2: Sequence Diagram to access geospatial services through Geospatial portal

Providers can also register their geospatial data and services to catalogue services that also give users a possibility to search and access to these information.
Figure A-3: Top Level Use case diagram for Environmental Information Network (Scenario 2 - providers make available their data and services in Catalog Services)

Table A-1: Top level use cases

<table>
<thead>
<tr>
<th>Actor</th>
<th>Top level use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>User and Provider</td>
<td>Environment Information Network</td>
</tr>
<tr>
<td></td>
<td>National Spatial Data Infrastructure (NSDI)</td>
</tr>
</tbody>
</table>

The above use case can be used in the same way for the National Spatial Data Infrastructure (NSDI).

Table A-2: Sub level use cases

<table>
<thead>
<tr>
<th>Actor</th>
<th>Sub level use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider</td>
<td><strong>Publish</strong> Data/Services, Data/Services Descriptions (metadata), and Ontology</td>
</tr>
<tr>
<td>User/Provider</td>
<td><strong>Search</strong> Data/Services, refer to semantics (ontology)</td>
</tr>
<tr>
<td>User/Provider</td>
<td><strong>Retrieve</strong>/Analyze Data and services</td>
</tr>
</tbody>
</table>

Provider can act as a user when accessing information from partner organizations.
Table A-3 : Types of Actors

<table>
<thead>
<tr>
<th>Actor</th>
<th>Provider</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholder Organizations</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Public Sector</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Private Sector</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NGO</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Researchers/Academia</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Citizens</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Used sub cases:

1. Publish Data

Description

The stakeholder organizations want to share their geospatial information among themselves and to make it available to the external user community. To share their data and services they have to decide which data should be shared with in the partner organizations and which data should be made available for the external user community. If they decide to share or disseminate the data and services, they have to create their description, the metadata, as well as the ontology that gives the meanings/concepts used and publish them in a catalogue service or any server to give information to the user. If the data or services are available, the user then search and access them using the portal or catalogue services in which they are registered.
2. Search Data

Description

An information user has to know which data he needs to solve his problem or to analyze the situation. The user can search for the available geospatial data and services using the searching mechanism available, such as catalogue services.
3. **Retrieve data**

*Description*

Once the user finds the data/service he wants to use, he can retrieve them. He then:
- Select/access the data or service
- Visualize and analyze the data

![Figure A-6: Sublevel Use case diagram– Retrieve Data/Services](image)

**Use Case Example: Flood Risk Analysis**

A researcher (Mr. Y) wants to carry out flood risk assessment and related human vulnerability to environmental change for a certain area (e.g. river basin). He then needs to access the relevant datasets to analyze his questions.

Flood risk assessment involves different factors that cause vulnerability to flood disaster. Deforestation has an effect on reducing forest cover, as a result, soil erosion to be intensified. Rainfall intensity is another factor that causes river overflow, and removal of water-retaining vegetation which made riverine areas more susceptible to flooding. The severity of the floods, intern effect to the destruction of forest, wetlands, and dams on the main rivers and damage to human life. Another factor might be the change in land use. The change of Wetlands around the lake into cultivated land may cause to lose their function as natural filters for silt and nutrients. Run-off carries soil and excess nutrients from the cultivated areas...
into the lake. The resulting algal growth covers the surface water and reduces oxygen availability, seriously affecting the habitat of endemic fish species, which prefer clear waters. This further aggravates food insecurity in lakeside communities [63].

This analysis needs many datasets to be used. Forest cover, soil moisture, Rainfall, precipitation, vegetation cover, rivers, streams, lakes, wetlands, dams, settlements, population are possible datasets to use for this analysis.

He can access these datasets from different distributed systems through the geospatial portal or catalogue services without even knowing they are physically in different locations. And he can integrate it into his system to analyze his problems. All these datasets are not used in this research, however, would be included in the themes of environmental datasets. Some of the themes that can be included are:

1. Environmental: Conservation
2. Hydrology
3. Climate
4. Geology
5. Infrastructure
6. Topography
The following *Example Use case diagram* shows how the user search and retrieve the information. It is a combination of search (figure ) and retrieve (figure ) operations.

![Figure A-7 : Example Use Case Diagram – Search and Retrieve Data/Services](image-url)
A. 2. REQUIREMENT ANALYSIS - ENVIRONMENTAL INFORMATION NETWORK

i. Introduction

This requirement analysis is based on the user perspective in facilitating access to geospatial data and services. The use case indicates the information providers and information users as shown in the top level use case diagrams for Environmental Information Network (Figure A-1 and Figure A-3).

The use case describes how the different actors (the geo-information providers and users) interact in sharing the available data/services by means of the geo portal. The data/services are made available along with their description and semantics by the stakeholder organizations to be accessed through the geo portal or published in catalogue services.

The elements of the framework in developing a mechanism to facilitate sharing of geospatial data and services are listed below.

A. Interface for searching and accessing the geospatial data/services (Geospatial portal)

B. Services and Functionalities
   a. Geospatial data/services (provided/published by the providers)
   b. Geospatial data/service descriptions, metadata, published by the providers
   c. Ontology which gives a common understanding of the concepts used between the provider and the user

C. Guidelines to be used to publish data/services and their metadata by each stakeholder organizations.

A. Interface (Geo Portal)

The Geo portal (interface) should provide functionalities to browse and access geospatial data/services. The Geospatial portal would be developed and maintained by geo-information providers (the technical working group members drawn from each stakeholder organizations). The prototype geospatial portal is created in this research.

B. Services and Functionalities

- Geospatial data and Services
  Geospatial data and Services and their metadata are published by the providers. Users search and access the data and services by means of interface, the Geospatial portal.

- The ontology which gives a common understanding of the concepts used between the provider and the user should be provided by the provider.

96
The functionalities provided are shown in the table below.

<table>
<thead>
<tr>
<th>Publish data/services</th>
<th>Publish Data/Services and their metadata (Server and Catalogue Service)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Publish Ontology</td>
</tr>
<tr>
<td>Search data/services</td>
<td>Search Geospatial data and services from Catalogue Services or by means of geospatial portal</td>
</tr>
<tr>
<td>Retrieve data/services</td>
<td>Access Geospatial data and Services</td>
</tr>
</tbody>
</table>

### C. Guidelines

Guidelines should be prepared and used for sharing and disseminating information. Metadata should be created based on the metadata standards. And all data/service providers should use the guideline documents for publishing data/services and their metadata.

#### ii. Usage

Stakeholder Organizations should collaborate to facilitate the sharing of geospatial data and services. They have to create and maintain data/services and metadata according to the standards and guidelines. The data/service providers are stakeholder institutions (here an example is taken from Ethio_EIN):

1. Environmental Protection Authority (EPA) (coordinator);
2. Ministry of Agriculture and Rural Development (MoARD);
3. Ministry of Water Resources (MoWR);
4. Ministry of Health (MoH);
5. Disaster Prevention and Preparedness Commission (DPPC);
6. Population and Housing Census Commission Office (PHCCo);
7. Central Statistical Authority (CSA);
8. Ethiopian Mapping Authority (EMA);
9. Geological Survey of Ethiopia (GSE);
10. Institute for Bio-Diversity Conservation (IBC);
11. National Meteorological Service Agency (NMSA), and
12. Christian Relief and Development Association (CRDA), representative for non government organizations (NGO).

The Users Communities are:

1. Public Sector
2. Private Sector
3. NGOs
4. Researchers/Academia
5. Citizens
Appendix B. Ontology Development

B.1. In this research a domain ontology is developed that represents the real world situation in the specific domain: environmental domain. The environmental domain ontology is created based on the GCMD Earth Science keywords. The concepts in this domain are shown in the hierarchical representation below.

- Environmental_Domain
  - Agriculture
    - Soil
  - Atmosphere
    - Atmospheric_Temperature
      - Mean_Temperature
      - Minimum_Maximum_Temperature
      - Potential_Temperature
    - Atmospheric_Water_Vapor
      - Evapotranspiration
        - Actual_Evapotranspiration
        - Potential_Evapotranspiration
    - Precipitation
      - Precipitation_Amount
      - Rain
  - Eiosphere
    - Vegetation
    - Wetlands
  - Human_Dimensions
    - Boundaries
      - Administrative_Boundaries
      - Political_Boundaries
    - Economic_Resources
      - Mines
    - Environmental_Impacts
      - Conservation
    - Infrastructure
      - Buildings
      - Builtup_Area
    - Transportation
      - Airfields
      - Rail_Roads
      - Roads
  - Natural_Hazards
    - Geological_Hazards
      - Earthquake_Occurrence
      - Volcanic_Eruption
    - Hydrological_Hazards
      - Drought
      - Flood
Figure B-1: Concepts of the Environmental Domain
B.2. Graphical representation of part of the Domain Ontology – Environmental Domain category based on the GCMD hierarchy.

Figure B-2: Graphical representation of part of the environmental domain ontology
B.3. Service Ontology refers to the domain of services, based on ISO 19119 service taxonomy. Figure B-3 shows the graphical representation of the service taxonomy.

```

▼ Service
  ● Communication_Services
  ▼ Geospatial_Model_information_Management_Services
     ● Catalogue_Service
     ● Coverage_Access_Service
     ● Feature_Access_Service
     ● Gazetteer_Service
     ● Map_Access_Service
  ▼ Human_Interaction_Services
     ● Catalogue_Viewer
     ● Geospatial_Feature_Editor
     ● Geospatial_Viewer
  ▼ Processing_Services
     ● Processing_Services-Metadata
     ▼ Processing_Services-Spatial
        ● Coordinate_Conversion_service
        ● Rectification_Service
        ● Route_Determination_Service
        ● Subsetting_Service
        ● Processing_Services-Temporal
        ● Processing_Services-Thematic
        ● System_Management_Services
        ● Workflow_and_Task_Services
```

Figure B-3: Service Ontology
B.4. An application Ontology was created for concepts related to the specific application, geospatial web services in an environmental domain, and based on the data at hand which uses the concepts from the domain ontology. The application ontology contains the Feature_Type concepts as shown in Figure B-4.

![Figure B-4: The FeatureType concept and its subconcepts (Application Ontology)](image)

B.5. Ontology File (Environmental Geospatial Services Ontology)

The OWL file describing the concepts of the ontology in the OWL DL language is shown below. These are an example of part of the ontology. The whole ontology file is accessible from in the following address.

http://geoserver.itc.nl/ Aster/Ontology/EnvironmentalGeospatialServices.owl#
Example: Environmental Domain Ontology

```xml
<?xml version="1.0" ?>
_< rdf:RDF
xmlns="http://geoserver.itc.nl/aster/Ontology/DomainOntology.owl#"
xmlns:j.1="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19107#
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:iso19110="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19110#"
xmlns:iso19108="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19108#"
xmlns:j.2="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19103#
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:j.3="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19111#"
xmlns:iso19115="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19115#
xmlns:daml="http://www.daml.org/2001/03/daml+owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:j.4="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19109#"
xml:base="http://geoserver.itc.nl/aster/Ontology/DomainOntology.owl">
_ <owl:Ontology rdf:about="">
  <owl:imports
rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19115" />
  <owl:imports
rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19103" />
  <owl:imports
rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19107" />
  <owl:imports
rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19109" />
  <owl:imports
rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-
19112" />
</owl:Ontology>
</rdf:RDF>
```
</owl:Ontology>
<owl:Class rdf:ID="Conservation">
<rdfs:subClassOf>
<owl:Class rdf:ID="Environmental_Impacts" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Topographical_Relief">
<rdfs:subClassOf>
<owl:Class rdf:ID="Topography" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Mines">
<rdfs:subClassOf>
<owl:Class rdf:ID="Economic_Resources" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Topography">
<rdfs:subClassOf>
<owl:Class rdf:ID="Land_Surface" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Human_Dimensions">
<rdfs:subClassOf>
<owl:Class rdf:ID="Environmental_Domain" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Landscape_Ecology">
<rdfs:subClassOf>
<owl:Class rdf:ID="Landscape" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Seismology">
<rdfs:subClassOf>
<owl:Class rdf:ID="Solid_Earth" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Natural_Hazards">
  <rdfs:subClassOf rdf:resource="#Human_Dimensions"/>
</owl:Class>

<owl:Class rdf:ID="Geologic_Age">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Geology"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Operations">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Rain"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Lakes">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Surface_Water"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Administrative_Boundaries">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Boundaries"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Landscape">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Land_Surface"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Biosphere">
  <rdfs:subClassOf rdf:resource="#Environmental_Domain"/>
</owl:Class>

<owl:Class rdf:about="#Environmental_Impacts">
  <rdfs:subClassOf rdf:resource="#Human_Dimensions"/>
</owl:Class>

<owl:Class rdf:ID="Precipitation_Amount">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Precipitation"/>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Potential_Evapotranspiration"/>
<rdfs:subClassOf>
    <owl:Class rdf:ID="Evapotranspiration" />
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Roads">
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Transportation" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Airfields">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Transportation" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Political_Boundaries">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Boundaries" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Economic_Resources">
    <rdfs:subClassOf rdf:resource="#Human_Dimensions" />
</owl:Class>
<owl:Class rdf:ID="Geological_Hazards">
    <rdfs:subClassOf rdf:resource="#Natural_Hazards" />
</owl:Class>
<owl:Class rdf:ID="Builtup_Area">
    <rdfs:subClassOf>
        <owl:Class rdf:ID="Infrastructure" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Geologic_Province">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Geology" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Soil">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Land_Surface" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Agriculture">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Agriculture" />
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Hydrosphere">
  <rdfs:subClassOf rdf:resource="#Environmental_Domain" />
</owl:Class>

<owl:Class rdf:ID="Inland_Water">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Surface_Water" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Transportation">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Infrastructure" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Surface_Water">
  <rdfs:subClassOf rdf:resource="#Hydrosphere" />
</owl:Class>

<owl:Class rdf:about="#Boundaries">
  <rdfs:subClassOf rdf:resource="#Human_Dimensions" />
</owl:Class>

<owl:Class rdf:ID="Buildings">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Infrastructure" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Population_Density">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Population" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Atmosphere">
  <rdfs:subClassOf rdf:resource="#Environmental_Domain" />
</owl:Class>

<owl:Class rdf:ID="Population_Size">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Population" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Service">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Geology" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Solid_Earth" />
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Geology">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Solid_Earth" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Minimum_Maximum_Temprature">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Atmospheric_Temprature" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Hydrological_Hazards">
  <rdfs:subClassOf rdf:resource="#Natural_Hazards" />
</owl:Class>

<owl:Class rdf:ID="Mean_Temprature">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Atmospheric_Temprature" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Tectonics">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Solid_Earth" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Fault">
  <rdfs:subClassOf rdf:resource="#Tectonics" />
  <rdfs:subClassOf rdf:resource="#Geology" />
</owl:Class>

<owl:Class rdf:ID="Volcanic_Eruption">
  <rdfs:subClassOf rdf:resource="#Geological_Hazards" />
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Volcanoes" />
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Solid_Earth">
  <rdfs:subClassOf rdf:resource="#Environmental_Domain" />
</owl:Class>

<owl:Class rdf:ID="Earthquake_Occurrence">
  <rdfs:subClassOf rdf:resource="#Geology" />
  <rdfs:subClassOf rdf:resource="#Seismology" />
  <rdfs:subClassOf rdf:resource="#Geological_Hazards" />
</owl:Class>

<owl:Class rdf:ID="Flood">
  <rdfs:subClassOf rdf:resource="#Hydrological_Hazards" />
  <rdfs:subClassOf rdf:resource="#Surface_Water" />
</owl:Class>

<owl:Class rdf:about="#Volcanoes">
  <rdfs:subClassOf rdf:resource="#Solid_Earth" />
</owl:Class>
<owl:Class rdf:ID="Atmospheric_Water_Vapor">
<rdfs:subClassOf rdf:resource="#Atmosphere" />
</owl:Class>

<owl:Class rdf:ID="Wetlands">
<rdfs:subClassOf rdf:resource="#Biosphere" />
<rdfs:subClassOf rdf:resource="#Surface_Water" />
</owl:Class>

<owl:Class rdf:ID="Drainage_Basin">
<rdfs:subClassOf rdf:resource="#Surface_Water" />
</owl:Class>

<owl:Class rdf:about="#Atmospheric_Temprature">
<rdfs:subClassOf rdf:resource="#Atmosphere" />
</owl:Class>

<owl:Class rdf:ID="Ground_Water">
<rdfs:subClassOf rdf:resource="#Hydrosphere" />
</owl:Class>

<owl:Class rdf:about="#Agriculture">
<rdfs:subClassOf rdf:resource="#Environmental_Domain" />
</owl:Class>

<owl:Class rdf:ID="River_Stream">
<rdfs:subClassOf rdf:resource="#Surface_Water" />
</owl:Class>

<owl:Class rdf:ID="Vegetation">
<rdfs:subClassOf rdf:resource="#Biosphere" />
</owl:Class>

<owl:Class rdf:ID="Contour">
<rdfs:subClassOf rdf:resource="#Topography" />
</owl:Class>

<owl:Class rdf:ID="Rail_Roads">
<rdfs:subClassOf rdf:resource="#Transportation" />
</owl:Class>

<owl:Class rdf:ID="Actual_Evapotranspiration">
<rdfs:subClassOf rdf:resource="#Evapotranspiration" />
</owl:Class>

<owl:Class rdf:about="#Infrastructure">
<rdfs:subClassOf rdf:resource="#Human_Dimensions" />
</owl:Class>

<owl:Class rdf:about="#Land_Surface">
<rdfs:subClassOf rdf:resource="#Environmental_Domain" />
</owl:Class>
<owl:Class rdf:about="#Evapotranspiration">
  <rdfs:subClassOf rdf:resource="#Atmospheric_Water_Vapor" />
</owl:Class>

<owl:Class rdf:about="#Population">
  <rdfs:subClassOf rdf:resource="#Human_Dimensions" />
</owl:Class>

<owl:Class rdf:about="#Precipitation">
  <rdfs:subClassOf rdf:resource="#Atmosphere" />
</owl:Class>

<owl:Class rdf:ID="River_Basin">
  <rdfs:subClassOf rdf:resource="#Surface_Water" />
</owl:Class>

<owl:Class rdf:ID="Terrain_Elevation">
  <rdfs:subClassOf rdf:resource="#Topography" />
</owl:Class>

<owl:Class rdf:ID="Drought">
  <rdfs:subClassOf rdf:resource="#Hydrological_Hazards" />
</owl:Class>

<owl:Class rdf:ID="Potential_Temperature">
  <rdfs:subClassOf rdf:resource="#Atmospheric_Temperature" />
</owl:Class>
Appendix C. Web Services Implementation

C.1. Web Service in HTML Client viewer

Web Map Service instances are implemented for five themes. Africa_Climate_Web_Map_Service is one instance of Web Map Services. Here it is viewed from the HTML client viewers created by ArcIMS.

Figure C-1: Africa_Climate_Web_Map_Service
C. II. Web Service in Gaia Client viewer

The services can be displayed and integrated with other services in client viewers. This example shows the map display created for Africa_Geology_Web_Map_Service in the Gaia client viewer.

Figure C-2: Africa_Geology_Web_Map_Service in the Gaia client viewer
Appendix D. Web Services Description

D.I The capabilities documents are edited to include the semantic annotation for the content of the services. The edited capabilities document (Example for Africa_Web_Map_Service) is presented here.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://geoserver.itc.nl/Aster/wms/WMS_MS_capabilities.dtd">
<WMT_MS_Capabilities version="1.1.1" xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <Service>
    <Name>Africa Climate Web Map Service</Name>
    <Title>Africa Climate Web Map Service</Title>
    <Abstract>Africa climate Web Map Service contains features: ecological region, rain, temperature, precipitation, and evapotranspiration. This service is created using ArcIMS 9.0 and compatible with OGC Web Map Service standards.</Abstract>
    <KeywordList>
      <Keyword>ArcIMS</Keyword>
    </KeywordList>
    <ContactInformation>
      <ContactPersonPrimary>
        <ContactPerson>Aster Denekew Yilma</ContactPerson>
        <ContactOrganization>ITC</ContactOrganization>
      </ContactPersonPrimary>
      <ContactPosition>M.Sc. Student</ContactPosition>
      <ContactAddress>
        <AddressType/>
        <Address/>
        <City>Enschede</City>
        <StateOrProvince/>
        <PostCode/>
        <Country>The Netherlands</Country>
      </ContactAddress>
    </ContactInformation>
  </Service>
</WMT_MS_Capabilities>
```
Ontology Based Portals for Accessing Geospatial Data and Web Services: Solving Semantic Problem in Data Sharing

</Get>
</HTTP>
</DCPType>
</GetFeatureInfo>
</Request>
</Exception>
</VendorSpecificCapabilities>
<Namespace OntoRef:RefersTo="Ontology"/>
</VendorSpecificCapabilities>
<Layer queryable="0" opaque="0" noSubsets="0">
  <Title>Africa Climate Web Map Service</Title>
  <Namespace OntoRef:RefersTo="Ontology#Africa_Climate_Web_Map_Service"/>
  <SRS>EPSG:4326 EPSG:4267 EPSG:4269</SRS>
  <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90"/>
  <Layer queryable="1">
    <Name>0</Name>
    <Title>Africa Boundaries</Title>
    <Namespace OntoRef:RefersTo="Ontology#Africa_Boundaries"/>
    <SRS>EPSG:4326</SRS>
    <LatLonBoundingBox minx="-25.360555" miny="-34.822000" maxx="63.495754" maxy="37.340409"/>
    <MetadataURL type = "TC211">
      <Format>TC211</Format>
      <OnlineResource xlink:href=""></OnlineResource>
      <Namespace OntoRef:RefersTo="Ontology#MD_Metadata_Africa_Boundary"/>
    </MetadataURL>
  </Layer>
  <Layer queryable="0">
    <Name>2</Name>
    <Title>Africa Potential Evapotranspiration</Title>
    <Namespace OntoRef:RefersTo="Ontology#Africa_PotentialEvapotranspiration"/>
    <SRS>EPSG:4267</SRS>
    <LatLonBoundingBox minx="-17.050000000000001" miny="-34.54999999999969" maxx="57.54999999999997" maxy="37.05000000000026"/>
    <MetadataURL type = "TC211"/>
Ontology Based Portals for Accessing Geospatial Data and Web Services: Solving Semantic Problem in Data Sharing

<Format>TC211</Format>
<OnlineResource xlink:href=""></OnlineResource>
<Namespace
OntoRef:RefersTo="Ontology#MD_Metadata_Africa_PotentialEvapotranspiration" />
</MetadataURL>
</Layer>

<Layer queryable="0">  
{Name>3</Name>
<Title>Africa Actual Evapotranspiration</Title>
<Namespace
OntoRef:RefersTo="Ontology#Africa_ActualEvapotranspiration="/>
<SRS>EPSG:4267</SRS>
<LatLonBoundingBox minx="-25.360554545454544" miny="-34.82200000000003" maxx="63.539445454545465" maxy="37.378"/>
<MetadataURL type = "TC211">  
<Format>TC211</Format>
<OnlineResource xlink:href=""></OnlineResource>
<Namespace
OntoRef:RefersTo="Ontology#MD_Metadata_Africa_ActualEvapotranspiration" />
</Layer>  

<Layer queryable="0">  
{Name>4</Name>
<Title>Africa Precipitation</Title>
<Namespace
OntoRef:RefersTo="Ontology#Africa_Precipitation="/>
<SRS>EPSG:4267</SRS>
<LatLonBoundingBox minx="-17.050000000000001" miny="-34.54999999999969" maxx="57.54999999999997" maxy="37.05000000000026"/>
<MetadataURL type = "TC211">  
<Format>TC211</Format>
<OnlineResource xlink:href=""></OnlineResource>
<Namespace
OntoRef:RefersTo="Ontology#MD_Metadata_AfricaPrecipitation" />
</Layer>  

<Layer queryable="0">  
{Name>5</Name>
<Title>Africa Rain</Title>
<Namespace
OntoRef:RefersTo="Ontology#Africa_Rain="/>
<SRS>EPSG:4326</SRS>
<LatLonBoundingBox minx="-20.037497999999999" miny="-34.822000000000031" maxx="51.9886320000000038" maxy="51.9886320000000038"/>
<MetadataURL type = "TC211">  
<Format>TC211</Format>
Ontology Based Portals for Accessing Geospatial Data and Web Services: Solving Semantic Problem in Data Sharing

171 <OnlineResource xlink:href=""></OnlineResource>
172 <Namespace
173 OntoRef:RefersTo="Ontology#MD_Metadata_AfricaRain"/>
174 </MetadataURL>
175 </Layer>
176 <Layer queryable="0">
177 <Name>6</Name>
178 <Title>Africa MeanTemprature</Title>
179 <Namespace
180 OntoRef:RefersTo="Ontology#Africa_MeanTemprature"/>
181 <SRS>EPSG:4267</SRS>
182 <LatLonBoundingBox minx="-17.050000000000001" miny="-34.549999999999969" maxx="57.549999999999997" maxy="37.050000000000026"/>
183 <MetadataURL type = "TC211">
184 <Format>TC211</Format>
185 <OnlineResource xlink:href=""></OnlineResource>
186 </MetadataURL>
187 </Layer>
188 <Layer queryable="1">
189 <Name>7</Name>
190 <Title>Africa EcologicalRegions</Title>
191 <Namespace
192 OntoRef:RefersTo="Ontology#Africa_EcologicalRegions"/>
193 <SRS>EPSG:4326</SRS>
194 <LatLonBoundingBox minx="-25.116026 " miny="-34.822000 " maxx="37.340409 " maxy="63.495755 ">
195 <MetadataURL type = "TC211">
196 <Format>TC211</Format>
197 <OnlineResource xlink:href=""></OnlineResource>
198 </MetadataURL>
199 </Layer>
200 <Layer queryable="1">
201 <Name>8</Name>
202 <Title>Africa Boundaries</Title>
203 <Namespace
204 OntoRef:RefersTo="Ontology#Africa_Boundaries"/>
205 <SRS>EPSG:4326</SRS>
206 <LatLonBoundingBox minx="-25.116026 " miny="-34.822000 " maxx="37.340409 " maxy="63.495755 ">
207 <MetadataURL type = "TC211">
208 <Format>TC211</Format>
209 <OnlineResource xlink:href=""></OnlineResource>
210 </MetadataURL>
211 </Layer>
212 </Capability>
213 </WMT_MS_Capabilities>
OGC uses DTD to define the elements and attributes used in Web Map Services Capabilities documents.

The OGC DTD located in http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd is modified locally to include the element <Semantics> which is introduced here for semantic reference and stored in the ITC geoserver. The DTD modified is presented here for reference.

<!ELEMENT WMT_MS_Capabilities (Service, Capability) >
<!ATTLIST WMT_MS_Capabilities
  version CDATA #FIXED "1.1.1"
  updateSequence CDATA #IMPLIED>

<!-- Elements used in multiple places. -->
<!-- The Name is typically for machine-to-machine communication. -->
<!ELEMENT Name (#PCDATA) >

<!-- The Title is for informative display to a human. -->
<!ELEMENT Title (#PCDATA) >

<!-- The abstract is a longer narrative description of an object. -->
<!ELEMENT Abstract (#PCDATA) >

<!-- An OnlineResource is typically an HTTP URL. The URL is placed in the xlink:href attribute. The xmlns:xlink attribute is a required XML Semantics declaration. -->
<!ELEMENT OnlineResource EMPTY>
<!ATTLIST OnlineResource
  xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"
  xlink:type CDATA #FIXED "simple"
  xlink:href CDATA #REQUIRED >

<!-- An ontology reference to the resources in the online ontology. -->
<!ELEMENT Semantics EMPTY>
<!ATTLIST Semantics
  xmlns:rdfls CDATA #FIXED "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdf CDATA #FIXED "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl CDATA #FIXED "http://www.w3.org/2002/07/owl#"
  xmlns:Sref CDATA #FIXED "http://www.ibm.com/xmlns/WebServices/WSSemantics"
  xmlns:Ontology CDATA #FIXED "http://geoserver.itc.nl/aster/ontology/EnvironmentalServicesOntology.owl#"
  Sref:RefersTo CDATA #REQUIRED >

<!-- A container for listing an available format's MIME type. -->
<!ELEMENT Format (#PCDATA) >

<!-- General service metadata. -->

<!ELEMENT Service (Name, Title, Abstract?, KeywordList?, OnlineResource,
  ContactInformation?, Fees?, AccessConstraints?) >

<!-- List of keywords or keyword phrases to help catalog searching. -->
<!ELEMENT KeywordList (Keyword*) >

<!-- A single keyword or phrase. -->
<!ELEMENT Keyword (#PCDATA) >

<!-- Information about a contact person for the service. -->
<!ELEMENT ContactInformation (ContactPersonPrimary?, ContactPosition?,
  ContactAddress?, ContactVoiceTelephone?,
  ContactFacsimileTelephone?,
  ContactElectronicMailAddress?) >

<!-- The primary contact person. -->
<!ELEMENT ContactPersonPrimary (ContactPerson, ContactOrganization) >

<!-- The person to contact. -->
<!ELEMENT ContactPerson (#PCDATA) >

<!-- The organization supplying the service. -->
<!ELEMENT ContactOrganization (#PCDATA) >

<!-- The position title for the contact person. -->
<!ELEMENT ContactPosition (#PCDATA) >

<!-- The address for the contact supplying the service. -->
<!ELEMENT ContactAddress (AddressType,Address,City,StateOrProvince,PostCode,
  Country) >

<!-- The type of address. -->
<!ELEMENT AddressType (#PCDATA) >

<!-- The street address. -->
<!ELEMENT Address (#PCDATA) >

<!-- The address city. -->
<!ELEMENT City (#PCDATA) >
<!ELEMENT StateOrProvince (#PCDATA) >

<!ELEMENT PostCode (#PCDATA) >

<!ELEMENT Country (#PCDATA) >

<!ELEMENT ContactVoiceTelephone (#PCDATA) >

<!ELEMENT ContactFacsimileTelephone (#PCDATA) >

<!ELEMENT ContactElectronicMailAddress (#PCDATA) >

<!ELEMENT Fees (#PCDATA)>  
<!ELEMENT AccessConstraints (#PCDATA)>  

<!-- A Capability lists available request types, how exceptions may be reported, and whether any vendor-specific capabilities are defined. It also includes an optional list of map layers available from this server. -->  
<!ELEMENT Capability (Request, Exception, VendorSpecificCapabilities?, UserDefinedSymbolization?, Layer?) >  

<!-- Available WMS Operations are listed in a Request element. -->  
<!ELEMENT Request (GetCapabilities, GetMap, GetFeatureInfo?, DescribeLayer?, GetLegendGraphic?, GetStyles?, PutStyles?) >  

<!-- For each operation offered by the server, list the available output formats and the online resource. -->  
<!ELEMENT GetCapabilities (Format+, DCPType+)>
<!ELEMENT GetMap (Format+, DCPType+)>
<!ELEMENT GetFeatureInfo (Format+, DCPType+)>
<!ELEMENT DescribeLayer (Format+, DCPType+)>
<!ELEMENT GetLegendGraphic (Format+, DCPType+)>
<!ELEMENT GetStyles (Format+, DCPType+)>
<!ELEMENT PutStyles (Format+, DCPType+)>
Available Distributed Computing Platforms (DCPs) are
listed here. At present, only HTTP is defined. -->
<!ELEMENT DCPType (HTTP) >

Available HTTP request methods. One or both may be supported. -->
<!ELEMENT HTTP (Get | Post)+ >

URL prefix for each HTTP request method. -->
<!ELEMENT Get (OnlineResource) >
<!ELEMENT Post (OnlineResource) >

An Exception element indicates which error-reporting formats are supported. -->
<!ELEMENT Exception (Format+)>}

The semantics of the service is referring to the Ontology reference using Semantics element defined as
vendor-specific capabilities.-->  
<!ELEMENT VendorSpecificCapabilities (Semantics)>

Optional user-defined symbolization (used only by SLD-enabled WMSes). -->
<!ELEMENT UserDefinedSymbolization EMPTY >
<!ATTLIST UserDefinedSymbolization
    SupportSLD (0 | 1) "0"
    UserLayer (0 | 1) "0"
    UserStyle (0 | 1) "0"
    RemoteWFS (0 | 1) "0" >

Nested list of zero or more map Layers offered by this server. -->
<!ELEMENT Layer ( Name?, Title, Abstract?, KeywordList?, Semantics, SRS*,
    LatLonBoundingBox?, BoundingBox*, Dimension*, Extent*,
    Attribution?, AuthorityURL*, Identifier*, MetadataURL*, DataURL*,
    FeatureListURL*, Style*, ScaleHint?, Layer* ) >

Optional attributes-->
<!ATTLIST Layer queryable (0 | 1) "0"
    cascaded CDATA #IMPLIED
    opaque (0 | 1) "0"
    noSubsets (0 | 1) "0"
    fixedWidth CDATA #IMPLIED
    fixedHeight CDATA #IMPLIED >

Identifier for a single Spatial Reference Systems (SRS). -->
<!ELEMENT SRS (#PCDATA) >

The LatLonBoundingBox attributes indicate the edges of the enclosing
rectangle in latitude/longitude decimal degrees (as in SRS EPSG:4326 [WGS1984 lat/lon]).

<!ELEMENT LatLonBoundingBox EMPTY>
<!ATTLIST LatLonBoundingBox
 minx CDATA #REQUIRED
 miny CDATA #REQUIRED
 maxx CDATA #REQUIRED
 maxy CDATA #REQUIRED>

<!-- The BoundingBox attributes indicate the edges of the bounding box
in units of the specified spatial reference system. -->
<!ELEMENT BoundingBox EMPTY>
<!ATTLIST BoundingBox
 SRS CDATA #REQUIRED
 minx CDATA #REQUIRED
 miny CDATA #REQUIRED
 maxx CDATA #REQUIRED
 maxy CDATA #REQUIRED
 resx CDATA #IMPLIED
 resy CDATA #IMPLIED>

<!-- The Dimension element declares the _existence_ of a dimension. -->
<!ELEMENT Dimension EMPTY>
<!ATTLIST Dimension
 name CDATA #REQUIRED
 units CDATA #REQUIRED
 unitSymbol CDATA #IMPLIED>

<!-- The Extent element indicates what _values_ along a dimension are valid. -->
<!ELEMENT Extent (#PCDATA)>
<!ATTLIST Extent
 name CDATA #REQUIRED
 default CDATA #IMPLIED
 nearestValue (0 | 1) "0">

<!-- Attribution indicates the provider of a Layer or collection of Layers.
The provider's URL, descriptive title string, and/or logo image URL may be
supplied. Client applications may choose to display one or more of these
items. A format element indicates the MIME type of the logo image located at
LogoURL. The logo image's width and height assist client applications in
laying out space to display the logo. -->
<!ELEMENT Attribution ( Title?, OnlineResource?, LogoURL? )>
<!ELEMENT LogoURL (Format, OnlineResource)>
<!ATTLIST LogoURL
 width NMToken #REQUIRED>
<!ELEMENT MetadataURL (Format, OnlineResource, Semantics) >
<!ATTLIST MetadataURL
type ( TC211 | FGDC ) #REQUIRED>

<!ELEMENT AuthorityURL (OnlineResource) >
<!ATTLIST AuthorityURL
name NMTOKEN #REQUIRED >

<!ELEMENT Identifier (#PCDATA) >
<!ATTLIST Identifier
authority CDATA #REQUIRED >

<!ELEMENT DataURL (Format, OnlineResource) >

<!ELEMENT FeatureListURL (Format, OnlineResource) >

<!ELEMENT Style ( Name, Title, Abstract?,
LegendURL*, StyleSheetURL?, StyleURL? ) >

<!ELEMENT LegendURL*>
<!ELEMENT StyleSheetURL?>
<!ELEMENT StyleURL? >

<!ELEMENT LegendURL*>
<!ELEMENT StyleSheetURL?>
<!ELEMENT StyleURL? >

<!ELEMENT LegendURL*>
<!ELEMENT StyleSheetURL*>
legend. -->
<!ELEMENT LegendURL (Format, OnlineResource)>
<!ATTLIST LegendURL
   width NMTOKEN #REQUIRED
   height NMTOKEN #REQUIRED>

<!-- StyleSheetURL provides symbology information for each Style of a Layer. -->
<!ELEMENT StyleSheetURL (Format, OnlineResource)>

<!-- A Map Server may use StyleURL to offer more information about the data or symbology underlying a particular Style. While the semantics are not well-defined, as long as the results of an HTTP GET request against the StyleURL are properly MIME-typed, Viewer Clients and Cascading Map Servers can make use of this. A possible use could be to allow a Map Server to provide legend information. -->
<!ELEMENT StyleURL (Format, OnlineResource)>

<!-- Minimum and maximum scale hints for which it is appropriate to display this layer. -->
<!ELEMENT ScaleHint EMPTY>
<!ATTLIST ScaleHint
   min CDATA #REQUIRED
   max CDATA #REQUIRED>
Appendix E. Geo portal

E.1. Interface for the portal

One of the objectives of this research is to develop a prototype ontology-based geo portal to give users access to semantically described services. The portal implemented is shown here in Figure E-1. This page gives access to Web Map Services and the layers in each service, the service metadata to understand the service and its capabilities, and their semantics using the ontology.

Figure E-1: Portal Home Page
E.2. Interface for the Web services

The research implemented five Web Map Services are implemented for different environmental themes. The services and their metadata can be accessed from the services page in the portal.

Figure E-2 : Service page in the portal
E.3. Interface for the service access

The service accessed in the portal shows the web map services and their metadata in the HTML viewer. The semantics can be referred from the ontology browser or OWL files in the ontology page.

![Environmental GeoPortal](image_url)

**Figure E-3 : Service access page in the portal**
E.4. Interface for the ontology (OWL Doc)

The Ontology browser in the portal gives access to the concepts (classes and properties) and their relationships in the ontology.

![Ontology Browser in the Portal](http://geoserver.itc.nl/aster/Home/Ontology.html)

**Figure E-4: Ontology Browser in the Portal**