K.N. Toosi University of Technology
Surveying Engineering Faculty (Geodesy and Geomatics)

Discovery of Field-Based Geo Services Based on Ontology

A Thesis in
Geographic Information systems (GIS)

by

Gholamreza Fallahi

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

Supervisors:
Dr. Abbas Rajabifard　　Dr. Mohammad Saadi Mesgari

Adviser:
Prof. Andrew U. Frank

June 2008

K.N. Toosi University of Technology
Surveying Engineering Faculty (Geodesy and Geomatics)

the Degree of Doctor of Philosophy in
Surveying Engineering

Title

Promoting Interoperability of Geo-Spatial Information Communities by Resolving Semantic Heterogeneities

(To approach resolving Semantic Heterogeneities in field-based geo-services)

by
Gholamreza Fallahi

Supervisors:
Dr. Abbas Rajabifard   Dr. Mohammad Saadi Mesgari

Adviser:
Prof. Andrew U. Frank

June 2008

Table of Content

Content **Page**

Department of Geoinformation and Cartography
Technical University of Vienna
Gusshausstr. 27-29 / 127
A - 1040 Vienna

o.Univ.Prof. Dr. Andrew U. Frank frank@geoinfo.tuwien.ac.at Tel: +43 (1) 588 01-127 01 Fax: +43 (1) 588 01-127 99

Vienna, April 28th, 2008

To Whom it May Concern,

I am unfortunately unable to attend the defense of the dissertation of Mr. Gholamreza Fallahi on may 24th in Tehran. Other, previous commitments do not allow me to travel on this date to Tehran.

I appreciate the work of Mr. Gholamreza Fallahi and expect that his dissertation passes through the defense without impediments.

Kind regards,

Andrew U Frank

Prof. Dr. Andrew U. Frank

بسمه تعالی

تاریخ : ۸۷/۳/۵
شماره : ۱۱۶/۵۲۰
پیوست :

دانشگاه صنعتی خواجه نصیرالدین طوسی

فرم شماره ۱۰

صورتجلسه دفاع از رساله دکتری(Ph. D)

بدینوسیله گواهی می شود آقای غلامرضا فلاحی دانشجوی دکترای رشته مهندسی نقشه‌برداری ورودی سال ۱۳۸۰ در تاریخ ۱۳۸۷/۳/۴ از رساله خود باعنوان ارتقای تعامل جوامع اطلاعات جغرافیایی از طریق حذف ناسازگاری‌های معنایی دفاع وبا اخذ نمره ۱۸/۶ به درجه: بسیار خوب نائل گردید.

الف : درجه عالی ☐ نمره ۱۹ - ۲۰	ب : درجه بسیار خوب ☒ نمره ۱۸ - ۱۸/۹۹
ج : درجه خوب ☐ نمره ۱۶/۵ - ۱۷/۹۹	د : قابل قبول ☐ نمره ۱۵ - ۱۶/۵
هـ : غیر قابل قبول ☐	

ردیف	هیات داوران	مرتبه علمی	سمت	امضاء
۱	دکتر محمدسعدی مسگری	استادیار	استاد راهنما	
۲	دکتر عباس رجبی فرد		استاد راهنما/ مشاور	از طرف
۳	دکتر اندرو فرانک		استاد راهنما/ مشاور	از طرف
۴	دکتر حمیدرضا ربیعی	دانشیار	استاد داور داخلی / خارجی	
۵	دکتر محمدعلی رجبی	استادیار	استاد داور داخلی / خارجی	
۶	دکتر محمدجواد ولدان زوج	دانشیار	استاد داور داخلی / خارجی	
۷	دکتر علی منصوریان	استادیار	استاد داور داخلی / خارجی	
۸	دکتر حمید ابریشمی مقدم	دانشیار	نماینده تحصیلات تکمیلی دانشگاه	

* رساله احتیاج به اصلاحات جزئی دارد ☒
* رساله احتیاج به اصلاحات کلی دارد ☐
* رساله احتیاج به دفاع مجدد دارد ☐

مدیریت محترم تحصیلات تکمیلی دانشگاه

ضمن تائید مراتب فوق مقرر فرمائید اقدامات لازم بعمل آید.

معاون آموزشی و تحصیلات تکمیلی دانشکده

Acknowledgments

Thank God it's finally over! The Ph.D. process has been such an extremely demanding experience, sometimes more on the emotional level than the intellectual one. And truth be told, I could never have gone through it without help and support of several people around me.

I would like to thank Dr. Mohammad Saadi Mesgari and Dr Abbas Rajabifard as my research advisors for their valuable guidance throughout this study.

I'm especially indebted to my consultor, Prof. Andrew U. Frank, who gave me the opportunity to work at TU Wien and promote my research. His support and patience throughout my graduate study are acknowledged.

I would like to express my gratitude to all the colleagues at the National Cartographic Center of Iran- in particular to Dr. Mohammad Madad and Mr. Mohammad Sarpulaki- who provide me an opportunity to study in Ph.D.

My thanks also go to all the colleagues at K.N. Toosi University of Technology- specially the colleagues at Surveying Engineering Faculty- for their cooperation and collaborations during my graduated study at the university.

Last, but not least, I would like to thank my wife and my children for their understanding and love during the past years. Their support and encouragement was in the end what made this dissertation possible.

Abstract

Physical processes like surface flow, soil erosion and infiltration can be occurred in the environment. These physical processes are modeled for foreseeing and analyzing relation between them in order to study and research natural system. Scientists organize the structure of physical process in the form of a model which is a formal representation of relations between defined qualities and quantities. Most of physical processes have spatial nature. Geospatial Information System (GIS) as a repository of spatial data and library of geospatial operations for analyzing, representing and interpreting spatial data can produce qualities and quantities required by physical processes. Therefore, linking environmental models and GIS is an interesting research field for environmental modelers and GIS experts.

Currently, different methods are used for linking GIS and environmental models. Tightly coupled method for linking GIS and environmental models is one of these methods which lead to a standalone system to satisfy requirements and characteristics of a specified model and GIS. In the loosely coupled method, developer and user of system are confronted with tedious batch conversion operations, or import, export and access obstacles to distributed sources which have been imposed by heterogeneous data and processing environment.

Distributed computing architecture by using technologies such as COM, DCOM or CORBA establishes a tightly coupled relation between client and server. Therefore, in addition to tightly coupled drawbacks, it can not use advantage of existing World Wide Web (WWW).

Geospatial data and geo-service sharing from heterogeneous resources and GISs is main requirement for modeling physical processes in order to study and research natural environment. Distributed computing architecture based on loosely coupled geo-services as a new method for linking GIS and environmental models can satisfy needs of various disciplines in environmental and GIS domains. Basic tasks including geo-service publishing, discovering and requesting can be fulfilled through establishing communication and exchanging messages between service requester, provider and broker in this architecture. Message can be exchanged in a standard manner based on a set of computer network protocols. However this set of protocols can support syntactic interoperability between geo-service requester (modeler), geo-service provider (GIS expertise) and geo-service broker (publisher) rather than semantic interoperability.

In this research, ontology is introduced as a solution for describing semantic ambiguities in order to enhance semantic interoperability in distributed computing architecture based on loosely coupled geo-services. An ontology, typically, include describing taxonomy of domain specific concepts. Ontology can provide semantic description for geo-services and facilitate machine-oriented relation

between geo-service requester and geo-service provider. In this case geo-services can be more flexibly interpreted by intelligent machine agent. In this regards, required ontologies are developed in from of a layered base structure which decrease semantic ambiguities and play role of a knowledge base for discovery of geo-services. These common ontologies decrease semantic ambiguities in reasoning systems while discovery of geo-services commits to them. They contain axioms and constrains for describing general concepts and concepts which are used in measurement theory and geo-service domains and required for expressing geo-services.

Concepts and relations of ontologies must be formalized so that they become understandable for machine. The degree of formality employed in capturing the descriptions of concepts and relationships can be quite variable, ranging from natural language to logical formalisms, but increased formality and regularity clearly facilitates machine understanding. Description logics provide a base for new ontology languages and have come into focus in the knowledge engineering and ontology literature due to the rise in popularity of object-oriented design and proposals for the intended functionality of the Semantic Web. Therefore, in this research DL family is studied and OWL-DL is used for formalizing the proposed ontologies due to its expressiveness and existence of reasoning to support it.

During the current research a methodology based on ontology for discovery of geo-services is introduced and its architecture is discussed. A prototype of methodology is tested by building main ontologies, requested and provided ontologies of sample geo-services and implementing inference between correspondence concepts and measuring their similarities. In this regard, an application program in Java language is developed in order to compute the similarity between ontologies of provided and requested geo-services.

List of Figures

List of Tables

Chapter 1 **Introduction**

1.1 Motivation

The natural environment includes physical processes such as surface flow, soil erosion or infiltration. Environmental degradation occurs when nature's resources (such as trees, habitat, earth, water and air) are being consumed faster than nature can replenish them, when pollution results in irreparable damage done to the environment or when human beings destroy or damage ecosystems in the process of development. With this in mind the goal of environmental protection is to minimize such degradations.

In order to response to the goal of environmental protection, it is needed to model the physical processes in natural environment for prediction and analysis of relationship between phenomena. Models are simplified abstractions of reality representing or describing its most important/driving elements and their interactions. On the other words the scientists perceive and organize the structure of physical process such as water surface flow or infiltration as a model which is a formal representation of the relationships between defined quantities or qualities [Jeffers, 1982].

Most environmental phenomena and human activities do have spatial nature. The quantities or qualities (e.g. amount of flow used in water surface flow or soil type used in infiltration process) can be produced by GIS (Geospatial Information System) as a repository of geospatial data and library of geo-operations (e.g. calculating flow or interpolation) for analyzing, visualizing and interpreting geospatial data. Therefore, linking environmental models with GIS as analytical system is currently a well established field of environmental research.

The integrated linkage of GIS and environmental model as two standalone systems lead to again a standalone system. Tightly coupled approach satisfies the needs and characteristics of specific models and GISs. However a disadvantage of these approaches is the difficulty involved in modifying and integrating system components. In loose coupling linkage of two stand-alone systems, developer and the user are confronted with tedious batch conversion tasks, import/export obstacles, and distributed resource access barriers imposed by heterogeneous processing environments and heterogeneous data [Buehler and McKee, 1996].

Distributed computing architecture based on technologies such as COM (Component Object Model), DCOM (Distributed Component Object Model), RMI (Remote Method Invocation) or CORBA (Common Object Request Broker Architecture) over a communicating network which follows the principles of Client/Server architecture can be used to reduce the drawbacks of these methods. However due to the reason that these technologies use a tightly coupled relationship between client and server in addition to mentioned drawbacks they can not inherently take advantage of the existing World Wide Web [Newcomer, 2002].

To achieve the goal of environment protection, sharing geospatial data and geo-services from heterogeneous resources and GISs is basic requirement to model physical process. In this regards, a broad range of users belong to environmental disciplines and GIS are encouraged to exchange geospatial data and geo-services. An integrated infrastructure composed of geospatial data, geo-services and end-users, as well as policies and networks with regarding of interoperability is a basic means to access geospatial data and geo-services.

On the other words, spatial data infrastructure (SDI) facilitates access to, and responsible use of geospatial data at affordable cost in support of sustainable development [Groot, 1997]. The core components of an SDI are policy, access network, technical standards, people and data [Rajabifard and Williamson, 2001]. SDIs from global to local levels rely on interoperable geo-services. On the other words, the distributed computing architectures based on loosely coupled services as a new method for linking GIS and environmental model can properly be installed over an SDI as an enabling platform in which to serve different jurisdictions from global to local levels.

In this architecture, the geo-service requester (e.g. modeler), the geo-service provider (e.g. GIS experts or modeler) and the broker (people of SDI) are agents that deploy, discover, publish or invoke geo-services by exchanging messages produced according to standard industrial protocols. However these protocols can not provide sufficient means for automatic service discovery. They establish syntactic interfaces but do not specify the intended meaning of their terms in machine-understandable form and prevent the service requester and provider from understanding service semantics. There are implicit details in describing geo-services which may be exist in the requester's or provider's head or embodied in a piece of software [Uschold and Gruninger, 1996].

Therefore it is needed to explicitly describe implicit details of geo-services in a way such that the meaning become understandable not only for the human but also for the machine agents.

1.2 Problem Statement

Discovery of the appropriate geo-service is the first step of linking GIS and environmental models in the distributed computing architecture based on loosely coupled services. The research focuses on geo-services such as interpolation service which operate on spatially distributed attribute and produce new attributes. These kinds of geo-services are called field-based geo-services.

Suppose a modeler is looking for interpolation service whose task is to produce a continuous surface from some known locations. However interpolation methods depend on the type of attributes of the known points. For example algorithm for interpolating categorical data such as rock type is different from one which is used for numerical data such as amount of potassium. The characteristics such as numerical or categorical are implicit details of data that have a basic role in discovery of geo-service. Neglecting the characteristics such as measurement scale and unit of measurement in geo-service description may lead to discovery of the interpolation service with wrong interpolation method.

In current approaches of linking GIS and environmental models, these kinds of semantic plausibility checks are rare and mostly not easy to add [Bernard and Krüger 2000; Pundt et al. 1996]. Since the diversity of data sets and models makes it impractical to add code for each of them. This problem is worsened by the fact that a model is constantly modified to provide a better modeling result.

Solving the problem of semantic ambiguities and implicit details can help sharing models among different disciplines. Sharing models not only saves time but also allows sharing knowledge across disciplines and between and among different organizations [Dangermond, 2004].

According to above explanation the research question to be approached by this research is as follow:

Can the discovery of field-based geo-services be enhanced by describing the characteristics of their input and output data such as measurement scale and unit of measurement using ontology?

1.3 Research Hypothesis

Linking environmental models and GIS in a distributed computing architecture based on geo-services can be enhanced if the details of geo-services are described. Ontology can be used as means for describing the details of geo-services.

“An ontology is a specification of a conceptualization” [Gruber 1993].Ontology (capital “O”) is a philosophical discipline which studies the nature and structure of possible entities. An ontology (lowercase “o”) is a specific artifact designed with the purpose of expressing the intended meaning of a vocabulary in terms of the nature and structure of the entities it refers to [Borgo et al. 2005]. The ontologies can be used to negotiate meaning, either for enabling effective cooperation between multiple artificial agents, or for establishing consensus in a mixed society where artificial agents cooperate with humans.

Applying ontologies does not imply defining a global, standard communication, data dictionaries and used definitions to describe details of geo-services and compelling providers or requesters to have exactly such interpretation. The goal of using ontologies is to provide requesters and providers of geo-services with the freedom to communicate based on their own defined ontologies.

1.4 Research Objectives

In order to response to the research hypothesis the important objectives, to be approached by this research are as follow:

1. *To introduce the distributed GIS architecture based on loosely coupled geo-services and semantic issues associated with geo-services*

There is a new architecture for linking GIS and environmental models in which GIS consist of geo-services and modeler can discover the appropriate geo-service and apply it in the model. Most of the modelers use special type of geo-services which are according to their conceptualization from the environment. The objective is to introduce the architecture and clarify the semantic issues associated to these kinds of geo-services.

2. *To identify and document the concepts and relationships of the environmental models and GIS*

The first step is recognizing concepts and relationships and identifying details in the domain of environmental model and GIS. Concepts and relationships can be general, domain specific or related to a certain application. The existing ontologies which are compatible with these concepts are investigated.

3. *To build ontologies which includes the recognized concepts and relationships*

An ontology approximates the concepts and their relations involved in linking environmental models and GIS. It can get closer to a conceptualization: by developing a richer axiomatization, and by adopting a richer domain or a richer set of relevant conceptual relations [Guarino, 1997].

A contribution of the research is to propose an ontological structure to include all ontologies needed for geo-service discovery. These ontologies contain concepts and relationships which are consumable in GIS and environmental models. They can convey the details of geo-services in a way that leaves the no interpretation to the users' context and background.

4. *To study the specifications of Description Logic (DL) and identify an ontological language with sufficient constructors in order to be able to formalize the proposed ontologies*

The ontologies must be formalized using an ontological language so that they become understandable by machine as well as human. A language must be expressiveness enough to support a richer axiomatization. The more expressive the language, however, leads to the harder the reasoning.

The aim is to study the specifications of DL languages and to identify an ontological language which is expressiveness enough to formalize the concepts and their relationship. The language must also be able to support the inference between the ontologies of requested and provided geo-services at a reasonable cost.

5. *To propose a methodology in order to compute the similarity between requested and provided geo-services*

The main focus of this thesis is to propose methodology for evaluating the requested and provided geo-services. The methodology discovers the appropriate geo-service according to requirement of the modeler independent of modeler's context and background knowledge.

In order to clarify the proposed methodology, the provided and requested ontologies for the sample geo-services are built. Then the matching between these ontologies is performed and compared with a specified range of matching degrees in order to measure their similarities.

The following flowchart shows the relation between chapters and research objectives of this thesis:

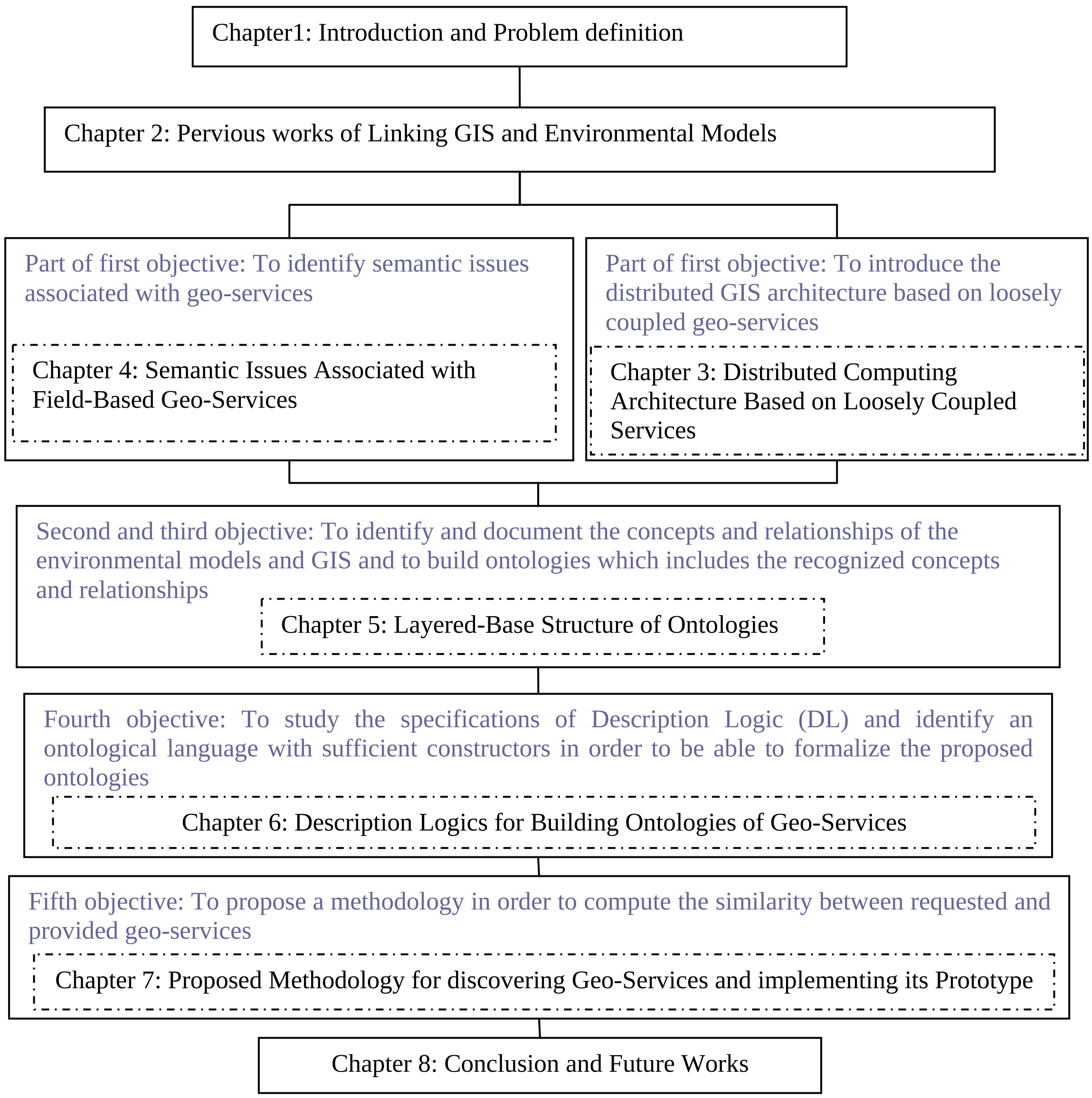

Fig. 1.1: The flowchart of relation between research objectives and chapters of the thesis

1.5 Research Approach

1. *<u>Study on different approach of linking GIS and environmental models for extracting their drawbacks</u>*

Currently there are different approach for linking GIS and environmental models. Knowing the drawbacks and advantages of these approaches as guidelines can assist us to propose more accurate and powerful approach by weakening the drawbacks and strengthening advantages of current approaches.

This study approaches the first objective in section 1.4.

2. *Study on GIS and environmental models for recognizing the details of geo-services*

The environmental models and GIS are studied in order to identify different concepts, and relations which may have key role in both domains. This study distinguishes between concepts which are general i.e. do not depend on a specific domain as well as concepts related to domain and application levels.

This study approaches the second objective in section 1.4.

3. *Study on existing ontologies for proposing and building ontologies which are applicable in geo-service discovery*

A study on existing upper ontologies is fulfilled and concepts at general and domain levels related to GIS and environmental models are recognized. Ontologies containing these concepts and relationships that are applicable in geo-service discovery are built. An ontological structure containing these ontologies is proposed.

This study approaches second and third objectives mentioned in section 1.4.

4. *Investigate logical languages for formalizing the ontologies at different levels as well as ontologies of requested and provided geo-services*

The expressiveness of Description Logics (DL) and difficulty of reasoning over language and enabling to work on web are considered as factors for selecting appropriate languages for this research. Alternatives such as ER (Entity Relation) or UML (Unified Model Language) diagrams, Conceptual Graphs [Sowa, 2000] or RDF (Resource Description Frame) schemata [Decker et al., 2000] are not studied, because, they have no reasoning systems or they are not expressive enough to analyze the knowledge represented by them. In this regards a DL language according to mentioned factors is identified in order to be used for formalizing the ontologies proposed by this research.

This study approaches the fourth objective in section 1.4.

5. *Proposing a methodology for geo-service discovery and development of a prototype for building ontologies and implementing the methodology*

A methodology for discovering geo-service based on the developed ontologies is proposed. In order to perform building ontologies and implementing the geo-service discovery an environment consist of the ontology editor and inference engine is represented.

Methodology is implemented by importing ontologies at different levels along with the requested and provided ontologies of sample geo-services and performing matching between the requested and provided ontologies.

This phase is related to fourth and fifth objectives in section 1.4.

6. *Assessment of the methodology*

The prototype developed in last phase for matchmaking as well as ontologies are used to illustrate the capabilities of the proposed methodology. The result is evaluated in terms of the problem definition and guidelines for future works are provided.

This phase basically relates to last four objectives and it shows strength and weaknesses of the methodology.

1.6 Thesis Structure

The structure of the thesis is as follow:

Chapter 2 Researches Related to Linking GIS and Environmental Models: This chapter introduces researches about linking GIS and environmental models. It focuses on researches, which take into account ontology for linking GIS and environmental models. The chapter pays special attention to research using ontologies, since aims of the research is to develop an approach based on ontology for discovery and composition of geo-services which can be consumed in environmental model by matching between ontology of provided geo-services and ontology of requested geo-services.

Chapter 3 Distributed Computing Architecture Based on Loosely Coupled Services for Linking GIS and Environmental Models: This chapter explains technical levels of linking GIS and environmental models based on different architectures. In this regard, the advantages and drawbacks of standalone GIS, client/server GIS and distributed GIS including old and new distributed computing technologies are discussed. This research pays attention to the distributed GIS architecture which is based on loosely coupled geo-services. The geo-services as programs interacting with each other can be discovered over the web. Thus, the protocols which are needed for publishing, discovering and invoking web services are introduced. The classification of geo-services is discussed based on different views of GIS. Finally a few atomic and complex services as case studies are introduced.

Chapter 4 Semantic Issues Associated with Field-Based Geo-services: This chapter after discussion about natural system and relation between environmental model and GIS pays attention to conceptualization of natural environment and especially focuses on how environmental modelers or scientists conceptualize natural environment. In this regards notion of field, field-based geo-services as well as characteristics of fields are introduced. Then semantic ambiguities of field-based geo-services are identified by regarding some field-based geo-services which were represented as case studies in the pervious chapter. Then classifications of semantic ambiguities which may arise due to lack of characteristics of fields are studied. The chapter also discusses about interoperability and its levels and concentrates on semantics issues as a mean for promoting interoperability in interaction model between field-based geo-services.

Chapter 5 Design and Methodology for Building Ontologies of Geo-services: This chapter introduces ontology as solution to describe semantic ambiguities. It proposes a layered-based structure of ontologies which reduces semantic ambiguities and promotes interoperability between geo-services in distributed computing architecture based on loosely coupled interactions. After studying various existing upper ontologies, an upper ontology which is the top layer of the structure is proposed containing general concepts for discovering of field-based geo-services. Then the ontology containing descriptions of measurement concepts related to geospatial data is discussed.

The chapter also discusses about problematic aspects of OWL-S as ontological language for describing capabilities and properties of geo-services, and proposes the core ontology of geo-services as fundamental ontology for overcoming the drawbacks of OWL-S. A contribution of the research is to develop this ontology and align it to the upper ontology. To align core ontology of geo-services to upper ontology the Descriptions and Situations (D&S) ontology is used to fill conceptual gaps between core ontology of geo-services and upper ontology.

Chapter 6 Description Logics for Building Ontology of Geo-service: This chapter introduces Description logics (DL), syntax and semantic of its concepts as well as constructs as key characteristics of DL for establishing the relationship between concepts. In this regard the OWL as a DL based ontology and central standard for ontologies on the Semantic Web is discussed. Then the ontology of requested and provided geo-services as case studies is developed. After that the various levels of matchmaking are introduced.

Chapter 7 Prototypical Implementation of Solution: This chapter introduces the architecture of proposed methodology for discovering geo-services. The methodology is based on ontology and a layered-based structure of ontologies is building blocks of the methodology. In this regards, the approach for formalizing these ontologies are discussed and the relation between them are clarified. Then tools for implementing prototype of this research including tools for building and management of ontologies and for implementing match between ontologies of required and provided geo-services are explained. Further, demonstration of proposed methodology by implementing prototype is discussed. The capabilities are evaluated in terms of the problem definition. This means that it is concentrated on how the prototype can handle the problems of discovering field-based geo-services.

Chapter 8 Conclusion and Future works: An evaluation of the achievements is presented here. This chapter summarizes abilities of the proposed methodology and presents weaknesses. Then the position of this work with respect to other related works in the scientific community is determined. It also presents the possible future improvements as well as conclusions.

Chapter 2 Pervious Works of Linking GIS and Environmental Models

2.1 Introduction

This chapter reviews and introduces researches about linking GIS and environmental models. It focuses on researches, which take into account ontology for linking GIS and environmental models. The chapter pays special attention to research using ontologies, since aims of the research is to develop an approach based on ontology for discovery and composition of geo-services which can be consumed in environmental model by matching between ontology of provided geo-services and ontology of requested geo-services.

This chapter reviews researches concerned with linking GIS and environmental models. Afterwards, it introduces research works addressing notions of communication, implicit details and semantic in linking GIS and environmental models and Geo-Information (GI) domain. Then it presents researches introducing notion of knowledge-based system in linking of models.

Further ontology projects and research works about using ontology in knowledge sharing, internet and methodology for building ontology are presented. Then the chapter discusses about relation between pervious works and the concern of this thesis.

2.2 Linking GIS and Environmental Models

Spatial data used in environmental models are usually derived from different available source such as data on topography, climate and weather, soil properties, geological properties, land cover, land use, hydrography and water quality. A review of these data sources is given in Moore et al (1993). Different research has focused on integration of geographic information. The importance of semantics in geographic information is well documented [Bishr, 1998; Egenhofer, 2002; Fabrikant et. al., 2001; Kuhn, 2002; Kuhn, 2005; Hakimpour, 2003]. Visser et al. (2002) describe how exchanging data between systems often fails due to confusion in the meaning of concepts.

On the other hand, environmental disciplines have entered modeling arena to develop environmental models with a variety of approaches and results. Many research works have been done to develop new models or link models.

Dangermond (2004) discusses that the geographic information is the descriptive part of GIS and functional part of GIS describes how geography changes (e.g. the models of erosion, flooding, vegetation growth and changes and urbanization).

Different approaches have been used so far to integrate GIS with environmental modeling. Fig 2.1 illustrates these approaches which are grouped into four categories. Most of researches in the field of GIS and environmental models use one of these methods for linking [Sydelko, et al, 2000, Fedra, 1996, Djokic, 1996].

According to Goodchild (2000) the integration of GIS and models can be classified to full integrated (embedding), loose coupling and tight coupling, each of them has its own advantages and drawbacks. Loose coupling means the linkage of two stand-alone systems by data transfer. According to categorized way of data transfer, several degrees of coupling systems can be distinguished [Nyerges 1992, Fedra 1996]. In this case, developer and the user are confronted with tedious batch conversion tasks, import/export obstacles, and distributed resource access barriers imposed by heterogeneous processing environments and heterogeneous data [Buehler and McKee, 1996].

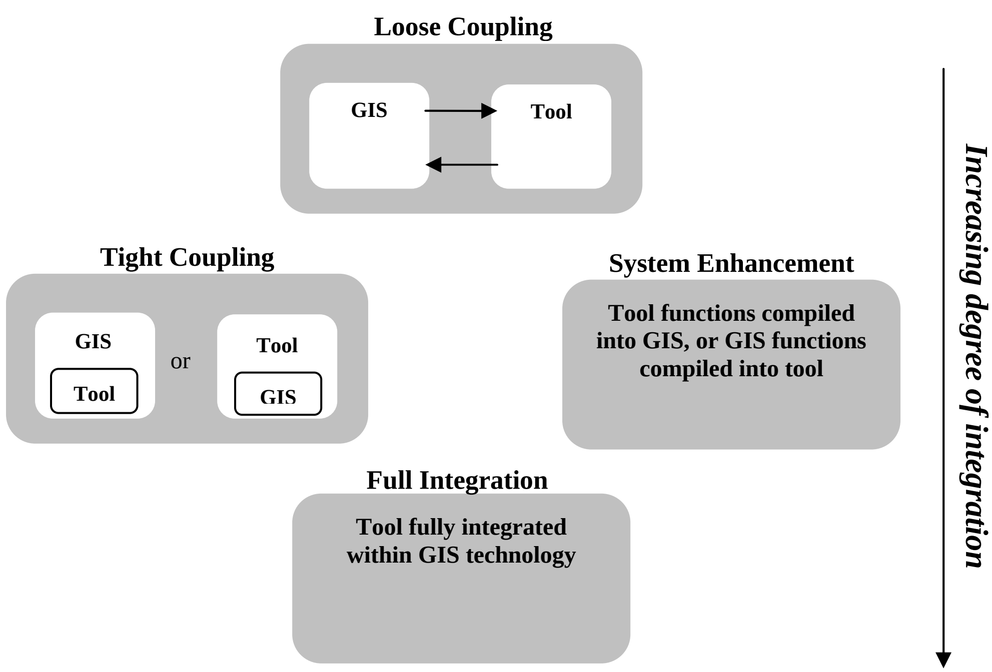

Fig. 2.1: Four ways of linking GIS and modeling technologies (after Fischer, 1994, P.237)

Tight coupling needs mechanisms in the running programs for data exchange, for instance remote procedure calls, while both system enhancement and full integration require access to source code. The CLIMEX system, presented by Fedra (1996), is an environmental decision support system used for global change modeling and assessment. It incorporates natural environment models such as agricultural and atmospheric models into a GIS using a tight coupling approach.

However, a disadvantage of many tight coupling approaches is the difficulty involved in modifying and integrating system components.

Generally integration means the implementation of GIS tools and simulation models on top of a common data and method base. But integration based on the existing closed and monolithic GIS and simulation models includes a high risk in designing systems, being again closed, monolithic, and therefore costly [Fedra 1996].

Current modeling approaches suffer from a lack of interoperability. In all cases where GIS data are used in an environmental modeling context, it is necessary to develop algorithms or methods that provide for the translation between the two representations of spatially relevant information. Translation algorithms developed to date have been tightly designed to the needs and characteristics of specific models and GISs. As a result, two types of problems present themselves: (i) resource, (ii) technical [GEOLEM, 2006]. The resource problem occurs because each connection between a specific pairing of an environmental model and a GIS requires a unique translation algorithm, which, in turn, requires new resources to repeatedly solve the same conceptual problem. The technical problem occurs in GIS/model combination software resulting from incompatible translation algorithms since each translation algorithm is unique to a model/GIS combination.

Furthermore, from software view point, the resulted system of linking GIS and environmental model may be developed by using different approach. The research works related to system development may be categorized according to various programming methods such as object oriented method, framework and other methods.

Under German initiative, for example, for interoperable and open geo-scientific systems funded by the German Science Foundation, an object oriented framework called atomGIS (Integration of atmospheric models and GIS) was designed. The layers VirGIS and AtomGIS are the essential parts of the atomGIS system. VirGIS consists of a set of interface classes which must be implemented by an integrated GIS-Kernel. AtmoGIS objects access the GIS-Kernel exclusively by the interfaces defined in the VirGIS layer [Bernard and Krüger, 2000]. The semantic checks have been coded inside the system, thus, sharing and matching semantic of environmental models and GIS is hindered due to tightly coupled semantic with system.

Feng (2000) discusses that the diversity of data sets and hydrologic models makes it impractical to build interfaces for each of them. This problem is worsened by the fact that a hydrologic model is constantly modified to provide a better modeling result. This may be the case for other environmental models. Feng and Sorokine (2001) identified that the component-based approach is an efficient way to integrate GIS and hydrologic models. They suggest that OpenGIS or ISO/TC 211 can be used as a vessel to achieve this goal, but there is a gap between what is provided in these specifications and what is needed for GIS hydrologic model integration. Their proposal is that the integration of GIS and hydrologic models requires encoding more specialized spatial domain used in hydrologic models.

Harvey (2003a) related to integrated hydrology models argues that:

> *With all the current discussion about "whole system" or "integrated" modeling and the pressing need for tools to allow the simulation of more aspects of catchment behavior, it is easy to lose sight of the bigger picture. In fact the very notion of "whole system" modeling is surely meaningless – where does the whole system stop? By a similar token, "integrated" can only possibly mean "more integrated than our current models".*

Harvey concludes that the "toolbox approach" is the next generation of modeling system.

HarmonIT is a project whose aim is to develop a framework for linking existing model implementations. It involves 14 European organizations including three major commercial modeling software vendors. The project is led by the Centre of Ecology and Hydrology (UK) and is supported by the European Commission under the Fifth Framework Programme [OpenMI_Newsletter]. The product of HarmonIT project will be the OpenMI (stands for Open Modeling Interface and Environment) and allows the linking of water related models [Hutchings et al, 2002].

The extrinsic complexity is addressed by the HarmonIT project, and its OpenMI deliverable rather than intrinsic complexity [Gijsbers, 2003]. Intrinsic complexity is incompatibilities between component models and gaps of knowledge at their interfaces in integrating models and an essential feature of the model while extrinsic complexity is difficulties anyone faced when attempting to integrate two existing model implementations [Harvey, 2003b].

2.3 Semantic Issues

Communication is an essential aspect in distributed computing architecture based on loosely coupled interaction implemented between environmental models and GIS. The exchanged messages in communicating between agents should be in a machine and human understandable form. In this regards implicit details and semantic of message terms play a basic role to produce understandable message. The following sections review semantic or implicit details affected linking GIS and environmental models and GI domain.

2.3.1 Semantic Issues in Linking GIS and Environmental Models

In HarmonIT project it is indicated that one of the weaknesses of existing and developing modeling systems and frameworks is parameter meaning and for transferring parameter values from one model domain to another it is needed to use a common language as an essential component of any communications [Hutchings et al, 2002].

In related to interoperability subject of the third Environmental Modeling with GIS conference hosted by the NCGIA in 1996 [Goodchild, 1996], Kemp argues that the integration of specific computing environments has been overtaken by consideration of the development of dominating theory and implementations through which everyone and every system is able to communicate with each other [Kemp, 1996].

Feng (2000) addresses the semantic as one of the requirement for open hydrologic model, which ensures interoperability between hydrologic process components. The hydrologic process is defined as the basic unit for a component that is hydrologically meaningful and conceptually sound. The semantics inherent within a component are concerned with conceptualization of a hydrologic process, and the mathematical equation employed in describing a hydrologic process. The semantic between components includes relationships between hydrologic processes and semantic mapping between hydrologic processes.

Bernard et al (1998) conclude that it is needed to focus on semantic plausibility checks for the model inputs and the model parameters. Bernard and Krüger (2000) put some codes into their atomGIS system to insure that the interpolation is not used for nominal or ordinal scaled data.

Harvey and Han (2002) discuss that, if two models follow a similar concept or start from a similar set of assumptions, suffer from fewer intrinsic barriers to linking than if the opposite is true. They believe that software frameworks, and the associated ontologies of the physical systems being modeled, do in fact serve a role in reducing the intrinsic barriers to future model linking by imposing some level of world view on the developers of models as components within such frameworks.

To date, spatial data have received little or no attention within modeling frameworks. Models that simulate the physical response of spatial features have typically reduced those spatial features to a set of parameter arrays or objects. There is no understanding of what the individual parameters actually are, or what the parameters collectively represent. While this is sufficient for the model, it frequently creates problems for the integration of the model with GIS and other software and limits interoperability.

2.3.2 Semantic Issues in Geospatial Information (GI) Domain

Geospatial means geographically referenced and thus geospatial data is a special type of spatial data that relates to the surface of the Earth. Information can be defined as data plus context and the structure of interrelationships between data and how data is collected, processed, used, and understood within an application forms the context for data [Worboys and Duckham, 2004].

The OGC (Open Geospatial Consortium) is an organization with members from different application domains. Its main goal is to enable GIS users to use geodata and services over networks and inter-networks. Therefore, the OGC provides specifications for interchanging information and geo-processing services between systems. Such specifications are produced under the consensus of all OGC members.

OGC's Sensor Web Enablement (SWE) activity, which is being executed through the OGC Web Services (OWS) initiatives (under the Interoperability Program), is establishing the interfaces and protocols that will enable a "Sensor Web" through which applications and services will be able to access sensors of all types over the Web [OGC, 2003]. Sensor Web report position and observations and Observations & Measurements (O&M) is the general model and XML encodings for sensor observations and measurements. In this model concepts such as unit of measure and scale of measurement for values which are observed by sensor from observations are encoded in form of XML. A XML schema, similar to a relational database schema, formalizes the structure and allows documents to be machine-read and parsed [XML]. XML is a uniform method for describing and exchanging data using HTTP. It is like HTML, where you make up your own tags but in XML you can’t say what your tags mean. XML provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents since it can not be considered as an ontology language. An ontology enables independently developed programs to exchange data. It specifies intended meaning in a computer interpretable form.

This research also pays attention to the measurement theory and their roles in linking GIS and environmental models. Taxonomy of the theory including concepts and their relationships is built as domain ontology using a logical language to specify the intended meaning of concepts and enable performing inference between concepts.

2.4 Knowledge-Based Link of Models

Due to the reason that the model in the computer is a representation of the modeler’s view of the natural system, Harvey (2003b) argues that the modeling tool is a knowledge representation (KR) technology. In his work, Harvey synthesizes a generic architecture by appeal to the diverse fields of hierarchical systems, knowledge representation, and computer science and a prototype framework conforming to this architecture [Harvey and Han, 2002]. Moreover, Hitchcock et al (1996) believe that knowledge based approaches seem to be an appropriate tool to perform the semantic plausibility check in linking models.

A knowledge-based system consists of a data structure that encodes a body of object-level information (the knowledge base) and a collection of procedures that perform inferences on the information in the knowledge base (The knowledge base manager) [Davis, 1990].

The present research focuses on the ontology as special kind of knowledge base for describing implicit details and semantic ambiguities in GIS and environmental models. However, the difference between an ontology and a knowledge base is that an ontology is a particular knowledge base which its purpose is describing facts assumed to be always true by a community of users, in virtue of the agreed-upon meaning of the vocabulary used. A generic knowledge base, instead, may also describe facts and assertions related to a particular state of affairs or a particular epistemic state [Guarino, 1997]. This effort seeks to eliminate the need for GIS-specific knowledge in the environmental model.

2.5 Ontology

Ontology is a term which has become popular especially in domains such as knowledge engineering, natural language processing, cooperative information systems, intelligent information integration, and knowledge management [Smith, 2002]. The ontologies can be used to negotiate meaning, either for enabling effective cooperation between multiple artificial agents, or for establishing consensus in a mixed society where artificial agents cooperate with humans. In this research, ontology is the base of approach which is used for knowledge sharing between GIS and environmental models. Thus this section briefly introduces ontology projects, methodologies for building ontologies and major contributions related to using ontology in knowledge sharing.

2.5.1 Ontology Researches and Projects

1. **Modeling Surface Hydrology Concepts with Endurance and Perdurance**

In this research endurance, perdurance, and granularity notions are used to model the semantics of hydrologic processes in surface hydrology [Feng et al 2004]. Endurance notion is related to concepts that exist in full at all time, while the perdurance notion is related to concepts that evolve through time. Granularity notion refers to the structure of concepts with respect to their scale. The result is a set of primitive entities and relations between these entities. These concepts are prerequisite for the identification of semantic barriers. The comparison between concepts indicates that: (1) multiple terms are used for the same concept, (2) inconsistent typing system are applied in categorizing a concept, (3) one term is used for to multiple concepts, (4) one term can carry an ambiguous concept, and (5) two concepts that overlap partially in meaning [Feng, 2005].

2. Adaptive and Composable E-emergency and Geographic Information Services Project

One of the objectives of this project is to develop an architecture for semantic interoperability in service composition and to supply its components for semantic modeling and mapping in the project [Probst, and Lutz, 2004]. In this project the composition of different services for the e-Emergency pilot has been approached. In this case, they encountered three types of heterogeneity in composition of services namely conceptual, naming and data type heterogeneity. Ontology structure in this project consists of application, conceptual and grounding level. They propose image schemata to semantically ground the concepts used on the conceptual level. The relations between these levels do not follow the relations between top, domain and application ontology as could be inferred from [Guarino, 1997]. The conceptual level serves to represent the vocabulary humans use to communicate about domain such as meteorology which used for emergency pilot in this project.

However, this research proposes a methodology based on ontology for linking environmental modeling and GIS. A layered-base structure is proposed which follows the relations between top, domain and application level according to what is inferred from Guarino (1997). In this regard an upper ontology is developed which uses DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) as backbone. General concepts related to GIS and environmental models are aligned with concepts in upper ontology and concepts of measurement theory and their relationships are plugged into upper ontology.

3. ARION Project

ARION (Advanced Lightweight Architecture for Accessing Scientific Collections) is a European Commission project under the Fifth Framework Programme in the domain of ocean and meteorology. In this project two application scenarios (Climar methodology, EUROWAVES methodology) are used for extracting workflows to be included in the ARION Digital Library. It aims to develop digital library which allows access to data and models over the World Wide Web. It is based on the coupling of ontologies with metadata and workflows. The representation of metadata in the ARION system is realized by RDF (Resource Description Frame) and the representation of ontologies are realized by RDF Schemata. They also use ontology software to define relationships between data/model domains [ARION, 2003].

Approach of the research is different from ARION in some ways. Research approaches a methodology for linking GIS and environmental systems based on ontology according to which the proper geo-services can be discovered through matching between modeler's requested ontology and provided geo-services ontology. The methodology does not depend to specific domain or any software which runs the environmental models. This research also uses a formalized language rather than RDF (for more information refers to Section 2.5.4) in order to perform inference between different ontologies.

4. Next Generation Computer Modeling for the Prediction of Flood Level and Inundation Extent

In this project the complementary issue of how to approach the design and implementation of a software framework in support of decision making processes in flood risk management is considered [Harvey et al, 2005]. In this regard they are looking for an ontology of software entities which is often referred to as a meta-model.

They use the Model Description Framework, layered on top of the Resource Description Framework (RDF) (for more information refer to Section 2.5.4) of the World Wide Web Consortium to develop such an ontology and representation [Harvey et al 2004].

2.5.2 Methodologies for Building Ontologies

Since ontologies are the basis of the approach presented in this thesis, this section briefly introduces the major contributions for building ontologies. The following works suggest methodologies to build ontologies

1. **The Enterprise Ontology**

Some requirements for a comprehensive description of the Enterprise Ontology, methodology for building ontology and a skeletal methodology for the process of developing and using ontology is outlined and the experiences about two significant ontologies in the domain of enterprise modeling are summarized [Uschold and King, 1995, Uschold and Grüninger, 1996, Uschold et al, 1997]. They proceed by considering how to identify what the important concepts and idea are in a domain of interest. Then a procedure and suggested guidelines for producing the definitions and how to reach agreement is given. They also present a prior version of the Enterprise Ontology.

2. **OntoClean**

A research group on the ontological foundations of knowledge engineering and conceptual modeling is exploring the role of ontology in different fields. Work of Guarino and Welty in this group is one of the major contributions to building and evaluating ontologies. They define notions such as identity, unity, individuality, and rigidity [Welty and Guarino, 2001, Guarino and Welty, 2002]. These notions play an important role in qualification of taxonomy hierarchies. By applying such notions one can evaluate an ontology in term of its explication (i.e., how an ontology reveals implicit assumptions) and its accordance with the conceptualization of a community.

3. **Affordance**

Frank and Kuhn present a new approach towards defining semantics [Kuhn, 2001, Frank and Kuhn, 1998]. Unlike conventional approaches for defining ontologies mainly based on logical phrases, they define semantics based on activities. That is, defining categories of objects based on the actions they can afford (for more details about affordance refer to [Gibson, 1986]). An immediate advantage of defining categories of objects according to the activities is reducing the magnitude of the problem of detecting similarities between the categories. Frank and Kuhn are using a functional language (Haskell [Hudak, 2000]) for formalizing the activities [Kuhn, 2001, Frank and Kuhn, 1998].

2.5.3 Ontologies in Knowledge Sharing

1. **KIF (Knowledge Interchange Format)**

KIF is a variant of the language of the first-order predicate calculus, motivated by the goal of developing an expressive, flexible, computer- and human-readable medium for exchanging knowledge bases [Genesereth and Fikes 1992]. Although it is not conceived for ontological purposes, but the language KIF is a milestone in the development of ontology as a solution to the problems of knowledge sharing and knowledge integration.

Its notation offered a convenient means by which formulas of first-order logic could be entered at the keyboard, without the need to use special symbol fonts. KIF includes von Neumann-Bernays-Gödel set theory as a constituent part. The semantic side of KIF rests on the technical notion of conceptualization introduced by Genesereth and Nilsson (1987). Conceptualizations are built up out of two sorts of components: a universe of discourse is made up of objects hypothesized to exist in the world and a set of relevant properties, relations and functions, which are extensionally conceived as sets of ordered tuples. Objects are subdivided into individuals, on the one hand, and sets or classes on the other. More precisely, relations and functions are sets of (finite) lists of objects, lists themselves being finite sequences of objects.

Given a conceptualization, the individual terms of KIF denote objects in the associated universe of discourse; the predicate terms of KIF are assigned values from the set of associated properties and relations. Semantics is then relative to conceptualization in the sense that sentences are true or false according to the conceptualization with which one begins.

2. ONTOLINGUA

On the basis of KIF, Tom Gruber and his associates at the Stanford Research Institute developed a more serviceable language for ontology representation known as Ontolingua [Gruber 1992]. Ontolingua is built up on the basis of KIF 3.0, but it has a very distinctive purpose. Where KIF is conceived as an interface between knowledge representation systems, Ontolingua is intended as an interface between ontologies. Users can translate their ontologies into different languages (e.g., KIF, Loom or Prolog) to be used with existing knowledge based systems. It provides an environment and a set of software tools designed to enable heterogeneous ontologies to be brought together on a common platform via translation into a single language. The Ontolingua Web site provides users with tools to define their ontologies. The library of ontologies provided by Ontolingua is a useful reference to find and adopt existing ontologies as higher-level ontologies. For instance, the ontology of Physical-Quantity is applied in the upper ontology proposed in the research.

Ontolingua is highly expressive also provides the Frame-Ontology [Ontologua] for frame based (object-centered) knowledge representation. However, such an expressive language has resulted in the fact that no reasoning system is yet supporting the Ontolingua language [Hakimpour, 2003].

Neither KIF nor Ontolingua embraces the idea of a single shared ontology. There is no suggestion that its authors wanted to incorporate even such notions as time and process, matter and mind within their respective frameworks. The major goal of the authors of Ontolingua was rather to collect a large number of distinct, specialized ontologies, and to provide the linguistic resources for moving back and forth between them [Smith, 2002].

3. Cyc

Cyc started as a research project in the early 80's. It is an information systems ontology projects [Lenat 1995, http://www.cyc.com] which formalize common-sense knowledge in the form of a massive database of axioms covering all things, from governments to mothers. Cyc is intended to be able to serve as an encyclopedic repository of all human knowledge. ('Cyc' comes from en-cyc-lopedia.) As such it purports to provide a medium for the representation of facts and the inscription of rules about all existing and imaginable things.

CycL is the knowledge representation language associated with Cyc. It is sometimes presented as a second-order language, sometimes as an (unsorted) first-order language with higher-order capabilities. Cyc allows quantification over predicates and relations and, more generally, it admits collections, objects of arbitrary order built on the first layer of individuals (collections of individuals, collections of such collections, and so forth). It allows the expression of relations holding among objects, relations holding among collections and also relations holding among CycL sentences themselves. CycL also possesses some of the resources of modal logic, together with features derived from natural language such as generalized quantifiers ('every', 'most', 'many').

Cyc itself is a knowledge base written in CycL. Thus it is a set of CycL expressions. It is referred to as an ontology in the sense that it 'contains' objects, roughly speaking the CycL terms, articulated by axioms, which are CycL sentences.

4. $(KA)^2$

$(KA)^2$ [Benjamins and Fensel, 1998, Fensel et al., 1999] is an initiative for building ontologies in the knowledge acquisition community. The goal of the project is to develop an ontology for the participants. The ontology built by this initiative is mainly used in On2Broker for searching on the Web. The $(KA)^2$ Ontology is developed using Ontolingua tools. Participants in $(KA)^2$ need to manually annotate their Web pages based on a specified annotation to facilitate the search process by On2Broker. The annotations relate information on Web pages to the $(KA)^2$ ontology.

2.5.4 Ontologies on the Internet

Semantic problems have attracted attention on the Internet [Fensel, 2001]. Considering that the Internet is a rich and wide spread repository of data and services, searching data and services using the semantics of documents have become an important issue. The Semantic Web as described in [Berners-Lee, 1998] is a goal for the W3 Consortium.

The following projects are using ontologies to improve the ability to search the World Wide Web. Thus these projects including On2Broker, RDF and RDF schema, DAML+OIL and OWL are briefly introduced.

1. On2Broker

On2Broker [Fensel et al., 1999] (new release of Ontobroker [Decker et al., 1999]) uses ontologies represented in a language based on Frame-based logic. Reasoning in On2Broker is based on closed-world assumption and deals with a domain specific ontology for every query. On2Broker uses its own extension to HTML tags that should be added by the document's author.

On2Broker uses an ontology produced by the $(KA)^2$ initiative [Benjamins and Fensel, 1998]. It maintains a taxonomy hierarchy by means of IS-A relations and represents attributes in their definitions. It uses a formalism similar to Frame-based Logic for reasoning and querying the system. As a result, the means to define relations between concepts in On2Broker is by rule definition. Here are the main features for the definition of terms in On2Broker:

- is-a hierarchy of inheritance for terms (similar to hierarchy of concept definitions)
- attribute definition (similar to a role definition or a relation with only type constraint)
- rules can not only play the role of constraints but also can be used to establish relations

2. RDF

Resource Description Framework (RDF) [Lassila et al, 1999] is a foundation for processing metadata. It provides interoperability between applications that exchange machine-understandable information on the Web. RDF uses XML to exchange descriptions of Web resources and emphasizes facilities to enable automated processing. The RDF descriptions provide a simple ontology system to support the exchange of knowledge and semantic information on the Web. All knowledge represented in RDF is based on a data-model of triple: subject, property and object. Property establishes a directed relation between two resources (and/or literals). It provides a simple semantics for this data-model, and these data-models can be represented in an XML syntax. Thus RDF does not present many features and contain little predefined semantics. RDF Schema [Brickley et al, 2000] provides the basic vocabulary to describe properties and classes of RDF resources, with semantics for generalization-hierarchies of such properties and classes. RDF Schema offers a set of RDF resources such as: class, subclass-of, attribute-of, subproperty-of, etc. to define properties, subjects or objects.

It is recognized as an ontology language, but it is still too restricted. If machines are expected to perform useful reasoning tasks on web documents, the language must go beyond the basic semantics of RDF Schema [McGuinness and van Harmelen, 2004]. Suitable reasoning systems are required to be developed to process the knowledge presented by RDF. An example of such system is SiLRI [Decker et al., 1998] which is a Frame-based Logic based inference engine.

3. DAML+OIL

DAML+OIL is the ontology of the Defense Advanced Research Projects Agency, a combination of the DARPA Agent Markup Language (http://www.daml.org), with the so called Ontology Inference Layer (OIL: http://www.ontoknowledge.org/oil). OIL [Horroks et al., 2000, Fensel, 2001] is a standard language to support exchange of ontologies on the Internet. It mainly extends the capabilities of XOL [Karp et al., 1999]. OIL is based on both Description Logic and Frame-based Logic. It inherits positive aspects from both formalisms by supporting the modeling primitives of Frame based Logic and the formal semantics of Description Logic. [Horroks et al., 2000] presents OIL in both XML and RDF syntax.

DAML+OIL have no way of treating individuals. It can only deal with classes/concepts. This is how the official DAML+OIL doctrine responds to this problem:

> *Results from research in description and modal logics show that the computational complexity of such logics changes dramatically for the worse when domain instances are allowed in class definitions … . For this reason, OIL currently does not allow the use of instances in slot-values, or extensional definitions of classes (i.e., class definitions by enumerating the class instances). It is not clear how serious a restriction [this ban on referring to individuals] is for an ontology language, as ontologies should, in general, be independent of specific instantiations – it may be that in many cases, 'individuals' can more correctly be replaced with a primitive class or classes [Horrocks et al., n. d.].*

4. OWL

Web Ontology Language (OWL) is part of the growing stack of W3C Recommendations related to the Semantic Web [McGuinness and van Harmelen, 2004]. The Semantic Web is a vision for the future of the Web, in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. 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 what can formally describe the meaning of terminology used in Web documents.

OWL has more facilities for expressing semantics than XML, RDF, and RDF-S. It can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. Thus, it is intended to be used for creating ontologies that are representations of terms and their interrelationships. Also, OWL was designed to satisfy the requirements that are not satisfied by various efforts that preceded it including DAML+OIL.

OWL provides three expressive sublanguages designed for use by specific communities of users: OWL Lite supports creating classification hierarchies and enables simple constraints. For example, while it supports cardinality constraints, it only permits cardinality of values 0 or 1. Moreover, it has a lower formal complexity than its more expressive relatives. OWL DL is named due its correspondence with description logics. OWL DL provides maximum expressiveness while retaining computational completeness and deducibility. That is, all conclusions are guaranteed to be computable and all computations will finish in finite time. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions. OWL full is union of OWL syntax and RDF. OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees.

Furthermore, OWL Full can be viewed as an extension of RDF, while OWL Lite and OWL DL can be viewed as extensions of a restricted view of RDF [McGuinness and van Harmelen, 2004].

2.6 Discussion

The ongoing research concerns linking GIS and environmental models by means of ontology. This chapter focused on research works addressing the following questions:

- What are the current approaches for linking GIS and environmental models;
- What is the role of semantic interoperability in linking GIS and environmental model;
- What kind of semantic ambiguities can be confronted while discovering geo-services needed by environmental modeler;
- How ontology can be used for knowledge sharing between GIS and environmental models;
- How ontology can solve the semantic ambiguities in linking GIS and environmental models;

Projects introduced in section 2.2 are major research works related to current approaches used for combining GIS/models. Research works in section 2.3 state the role of semantic interoperability and address the kind of semantic ambiguities or implicit details in GIS and environmental models and GI domain. In contrast, this research focuses on the semantic ambiguities and implicit details related to geo-services consumed in environmental models.

Related works in section 2.4 and 2.5, explain the role of knowledge base and ontology as a means to perform the semantic plausibility check as well as describing concepts and their relationships belong to environmental models. In this regard image schemata is used to semantically ground the concepts of the conceptual level and the relations between ontological levels for composition of geo-services do not follow the relations as could be inferred from [Guarino, 1997] however layered-base structure proposed by the research follows it.

Further areas of research are related to this thesis from building ontologies and formalizing and processing ontologies.

Building ontologies refers to describe concepts and their relationships related to geo-services from modeler's conceptualization. Building ontologies concerns questions such as:

- How ontology can be used for describing semantic ambiguities;
- How taxonomy of upper ontology is built;
- How different levels of ontology is organized;

The answer to these questions also has direct consequences on managing the ontologies. This thesis uses the term design for structure of ontologies. This is because terms or definitions of terms on GIS and environmental models are imposed in this thesis. In this regards, chapter 5 discusses issues related to structure of ontology and various layer of ontologies which fill the gaps between upper and application ontologies. The existing upper ontologies are studied to be used as backbone for upper ontologies proposed in this research.

Formalization and processing of ontologies as last perspective of this thesis refer respectively to logical languages for describing ontologies and reasoning the formalized ontologies to recognize similarity between requested and provided geo-services. Such a following problems are addressed here:

- What logical language from expressiveness point of view can be used for formalizing concepts and their relationships;
- How implicit details in the ontologies of requested and provided geo-services can be formulated or presented;
- How the degree of similarity between ontologies of requested and provided geo-services can be evaluated by applying reasoning;

The tradeoff between expressive power of language and the difficulty of reasoning over language is a factor that plays a basic role in this work. Chapter 6 discusses issues related to formalizing and reasoning over ontologies in more details.

Chapter 3 Distributed Computing Architecture Based on Loosely Coupled Services

3.1 Introduction

The environmental models do have an obvious spatio-temporal nature. They are occurred in a part of spatial region like surface and subsurface water flow, soil erosion, impact assessment in the event of a chemical and/or oil spill or urban extend. Thus environmental modelers encourage using GIS for processing, analyzing and visualizing spatial data in their modeling activity. There are currently different technical levels of linking GIS and environmental models.

This chapter explains technical levels of linking GIS and environmental models based on different architectures. In this regard, the advantages and drawbacks of standalone GIS, client/server GIS and distributed GIS including old and new distributed computing technologies are discussed. This research pays attention to the distributed GIS architecture which is based on loosely coupled geo-services. The geo-services as programs interacting with each other can be discovered over the web. Thus, the protocols which are needed for publishing, discovering and invoking web services are introduced. The classification of geo-services is discussed based on different views of GIS. Finally samples of atomic and complex services as case studies are introduced.

3.2 Current Approach for Linking GIS and Environmental Models

Currently different approaches are used to link GIS with environmental models. According to Goodchild (2000) varieties of these technical levels are from linking GIS and models by enabling them to exchange files (also called loose coupling) to add modeling capabilities to GIS (also called tight coupling). The highest level of integration is an embedded system, one in which GIS and modeling functions are interwoven elements of a software system (or full integration).

Two systems are considered loosely coupled if the only mandate imposed on both systems is to understand the self-describing, text-based messages. Tightly coupled systems, on the other hand, impose a significant amount of customized overhead to enable communication and require a greater understanding between the systems.

Tight coupling and full integration of traditional GIS with environmental models lead to high risk in designing systems which are closed, monolithic, centralized and costly [Fedra, 1996]. As illustrated in Fig. 3.1, since these systems are incorporating interfaces, programs and data. Every element is embedded inside traditional GIS and can not be separated from the rest of the architecture. Each method is unique to a model/GIS combination, and, modifying of integrated components in tight coupling and full integration are difficult.

Regarding the broad range of users belong to environmental disciplines, link of GIS and environmental models can be realized through sharing data and GIS functionality over a communicating network. However, linking traditional standalone GIS and environmental models based on loosely coupled, tightly coupled or full integrations are inflexible or error prone and cannot inherently take advantage of physical network and internet.

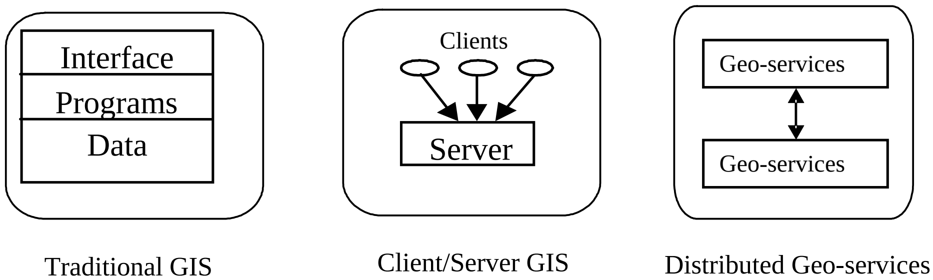

Fig. 3.1: Three alternative models for GIS architecture.

Distributed computing architectures which follow the principles of Client/Server architecture can be used for sharing data and GIS functionalities. The architecture based on tightly coupled relationship between client and server has included many technologies for building programs that can send data back and forth. COM (Component Object Model), DCOM (Distributed Component Object Model), RMI (Remote Method Invocation) or CORBA (Common Object Request Broker Architecture) are just a few. Due to the reason that these technologies use a tightly coupled relationship between client and server they can not inherently take advantage of the existing World Wide Web [Newcomer, 2002]. The intelligence for understanding how to map a message into a software program is contained within the interface itself, which tightly couple the service name to the program being invoked. CORBA is supposed to provide interoperable services in a heterogeneous multi-vendor environment, but according to Estrem (2003) it has not fulfilled its promise. One of the reasons why CORBA has not become mainstream is the demanding design process that SOA (Service Oriented Architecture) requires, and there are no appropriate development tools for SOA-enabled CORBA [Natis, 2003].

Due to the popular use of the Internet and the dramatic progress of communications and telecommunications technology, the paradigm of linking GIS and environmental models is shifting into increasingly distributed computing architecture. This architecture is based on loosely coupled relationship between client and server for deploying discovering and invoking web services. The loosely coupled interactions are better suited to integrating disparate software domains and bridging incompatible technologies. It also enables creating and defining APIs to support interoperable machine-to-machine interaction over a network without knowing anything about the client. Every service can become a client or a server based on the task at hand. On the other words there is no difference between a client and a server. A client is defined as the service requester in a network and server which provides a service is defined as the service provider. Such a flexible architecture is especially beneficial for scientific research and modeling where tightly coupled approaches are unlikely to have the desired breadth and flexibility.

3.3 Web Services

Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service (www-4.ibm.com/software/solutions/webservices/).

A service performs a specific task, in such way that the client is not expected to know anything about the internal functionality of the service [Capeclear, 2005]. A Web service is a software system identified by a URI (Uniform Resource Indicator). It has an interface described in a machine-processible format so that all service requesters are able to inspect the interface (e.g. protocol bindings and transport details). Web service is accessible only through its interface, usually in request/reply manner [Natis, 2003].

The interface description declares the operations which can be performed by the service, the types of messages being exchanged during the interaction with the service, and the physical location of ports, where information should be exchanged.

Programs written in any language, using any component model, and running on any operating system can access Web services. Furthermore, the flexibility of using a text format like XML makes possible the message exchange to evolve over time in a loosely coupled way.

3.4 Enabling Standard Protocols

Heterogeneous systems interact with the Web service in a manner prescribed by its interface using messages. These messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other Web-related standards. The only assumption made between the client and the server is that recipients will understand the messages they receive.

Key to understand the exchanged message is an adoption of a set of enabling standard protocols. The Web service protocol stack is the collection of computer networking protocols that are used to find, define, implement, and make web services interact with each other. The web services stack consists of five layers, as it is illustrated in Fig. 3.2.

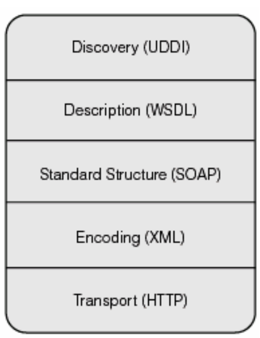

Fig. 3.2: A five layers stack of standard protocols for web services [Newcomer, 2002]

These layers consist of the following elements:

3.4.1 Transport - HTTP

At the lowest level, two components in a distributed architecture must agree on a common transport mechanism. Because of the near universal acceptance of port 80 as a less risky route through a firewall, HTTP became the standard for the transport layer. However, Web services implementations can run on other transport protocols such as FTP and SMTP, or even other network stacks, such as Sequenced Packet Exchange (SPX) or non-routable protocols such as NetBEUI.

3.4.2 Encoding -XML (eXtensible Markup Language)

The basic foundation on which Web services are built provides a language for defining data and how to process it. XML represents a family of related specifications published and maintained by the World Wide Web Consortium (W3C) and others.

After agreeing on the transport, components must deliver messages as correctly formatted XML documents. This XML dependence ensures the success of the transfer, because both provider and requester know to parse and interpret the XML standard.

3.4.3 Standard Structure - SOAP (Simple Object Access Protocol)

Although XML defines message encoding, it does not cover the structure and format of the document itself. To guarantee interoperability, both provider and requester must know what to send and what to expect. SOAP is a lightweight, message-based protocol to exchange information between nodes in a decentralized, networked environment built on XML and standard Internet protocols, such as HTTP. The SOAP protocol specification defines an XML structure for messages (the SOAP envelope), data type definitions, and a set of conventions that implement remote procedure calls and the format of any returned data (the SOAP body).

As shown in Fig. 3.3, a SOAP message consists of header and body information. A SOAP message travels between SOAP nodes on a SOAP message path from an initial sender through one or more intermediate nodes to an ultimate receiver. Each node on the path may process the message in some way based on information in the header blocks. The message body is processed by the ultimate receiver. SOAP does not define how messages are transported between nodes and how they are routed, but in the case of Web Services, relies on HTTP for this [Newcomer, 2002].

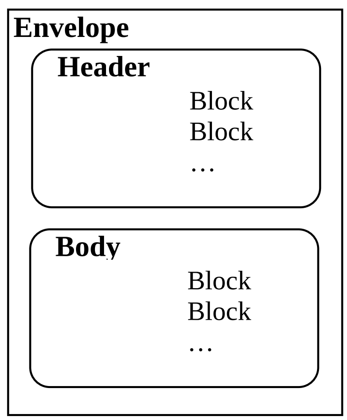

Fig.3.3: Three major parts of SOAP: envelope, header, and body [Newcomer, 2002]

3.4.4 Description - WSDL (Web Services Description Language)

According to W3C, WSDL defines service description about the message formats, data types, transport protocols, and transport serialization formats as a machine-process able specification of the Web service's interface [W3C, 2004].

The WSDL is an XML message format for describing the services offered by the server. WSDL file identifies the services and operations within each service provided by the service provider. For each of the operations, the WSDL file also describes the format that the service requester must follow in requesting an operation.

For example, suppose a WSDL file defines an ESRI's ArcWeb Service called ***mapImage*** (http://www.arcwebservices.com/services/v2006/MapImage.wsdl). This service describes operations such as ***getMaps***, ***getSavedMap***, and ***getValueMap***. This file can be placed on the server. A requester who wants to send a request to the provider first obtains a copy of this WSDL file from the server. The requester then uses the information in this file to format a request. The requester sends this request to the server. The server executes the requested operation and sends the result back to the requester as a response.

In a WSDL file, like an XML file has some definition.

```
<wsdl:definitions name="MapImage"
targetNamespace="http://www.arcwebservices.com/v2006"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:n4="http://www.webmethods.com/package/com.esri.is.services.soap.v3.mapimage
/"
xmlns:n5="http://www.webmethods.com/package/com.esri.is.services.soap.v3.common.g
eom/"
xmlns:n6="http://www.themindelectric.com/2001/schemaArray/"
xmlns:n7="http://www.webmethods.com/package/com.esri.is.services.soap.v3.common/"
xmlns:tns="http://www.arcwebservices.com/v2006"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
...
</wsdl:definitions>
```

Following the definition, there are five primary elements used in defining the service. These five elements appear in a WSDL file in the following order:

1. **<*types*>** element defines the various data types used in exchanging messages. For example **<*types*>** element for the ***getValueMap*** operation in ***MapImage*** service is as follow:

```
<wsdl:types>
<xsd:schema
      targetNamespace="http://www.webmethods.com/package/com.esri.is.services.soap.v
      3.mapimage/">
  <xsd:import
      namespace="http://www.webmethods.com/package/com.esri.is.services.soap.v3.co
      mmon/" />
  <xsd:import namespace="http://www.themindelectric.com/2001/schemaArray/" />
  <xsd:import
      namespace="http://www.webmethods.com/package/com.esri.is.services.soap.v3.co
      mmon.geom/" />
...
<xsd:element name="getValueMap">
      <xsd:complexType>
         <xsd:sequence>
           <xsd:element name="mapArea" type="n4:MapArea" />
           <xsd:element name="mapImageOptions" type="n4:MapImageOptions" />
           <xsd:element name="thematicData" type="n4:ThematicData" />
           <xsd:element name="codeColorValues" type="n7:ArrayOfKeyValue" />
           <xsd:element name="thematicOptions" type="n4:ThematicOptions" />
           <xsd:element name="token" type="xsd:string" />
         </xsd:sequence>
      </xsd:complexType>
</xsd:element>
<xsd:element name="getValueMapResponse">
      <xsd:complexType>
         <xsd:sequence>
           <xsd:element name="Result" nillable="true" type="n4:MapImageInfo" />
         </xsd:sequence>
      </xsd:complexType>
</xsd:element>
...
</xsd:schema>
</wsdl:types>
```

2. ***<message>*** element describes the messages being communicated. For example the ***<message>*** element that describes the input and output messages for ***getValueMap*** operation can be traced in the following segment of WSDL file. The input message is named ***getValueMap7In*** and the output message is named ***getValueMap7Out***.

```
...
<wsdl:message name="getValueMap7In">
  <wsdl:part name="parameters" element="tns:getValueMap" />
</wsdl:message>
<wsdl:message name="getValueMap7Out">
  <wsdl:part name="parameters" element="tns:getValueMapResponse" />
</wsdl:message>
...
```

3. ***<portType>*** element identifies a set of operations and the messages involved with each of those operations. ***<portType>*** which describe the ***getValueMap*** operation in ***MapImage*** service is as follow:

```
<wsdl:portType name="IMapImage">
      <wsdl:operation name="getValueMap">
         <input name="getValueMap7In" message="tns:getValueMap7In" />
         <output name="getValueMap7Out" message="tns:getValueMap7Out" />
      </wsdl:operation>
</wsdl:portType>
```

4. ***<binding>*** element specifies the protocol details for various service operations and describes how to map the abstract content of these messages into a concrete format. In the other words a binding then defines the machine and ports where messages should be sent. The ***<binding>*** element for ***getValueMap*** operation is as follow.

```
<wsdl:binding name="IMapImage" type="tns:IMapImage">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
      ...
      <wsdl:operation name="getValueMap">
         <soap:operation soapAction="getValueMap" style="document" />
         <wsdl:input name="getValueMap7In">
               <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output name="getValueMap7Out">
               <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
       ...
</wsdl:binding>
```

5. ***<service>*** element groups a set of related ports together. The ***MapImage*** service is described by ***<service>*** element in its WSDL file as follow:

```
 ...
    <wsdl:service name="MapImage">
         <wsdl:documentation>Map Image Web Service offers dynamic map content for
         your Internet applications. With this service, you input various options (such as
         data source, size of image, map annotation) and receive the URL location of an
         output image file. You can also merge map images and save maps for later use.
         </wsdl:documentation>
         <wsdl:port name="IMapImage" binding="tns:IMapImage">
               <soap:address
               location="http://www.arcwebservices.com/services/v2006/MapImage"
               />
         </wsdl:port>
    </wsdl:service>
```

WSDL is the key concept in developing and deploying Web services. The WSDL file sets up requirements including name, data types, methods and parameters for each exposed component. It is like a contract between the provider and the requester. WSDL describes the data types of the input-output variables, the name of the set of operations in each service, the format that the client must follow to invoke a service, and so on.

3.4.5 Discovery - UDDI (Universal Description, Discovery, and Integration)

Catalog service is a component in service oriented architecture to discover the types of service and data and their relevant instances. A catalog service plays a ‘directory’ role in helping providers to describe and advertise the resources availability and requestors to discover the right resources.

Catalog services or registry allow users and applications to classify, register, describe, search, maintain, and access information about Web services (see www.opengeospatial.org/ogcSpecs.htm). A Web services registry and discovery mechanism is used for retrieving pointers to Web services interfaces. On the other words, Web service discovery is the process of locating, or discovering, one or more related documents that describe a particular Web service using the WSDL. It is through the discovery process that Web service requesters learn that a Web service exists and where to find the Web service's description document.

Discovery attempts to answer the question "Where." If you want to connect to a Web service at an Internet location you can enter the URL manually. However, URLs are somewhat unwieldy and not very user friendly, so it would be better if the requester could just request the name of Web Service. To do this, service provider could publish the service on a Universal Description, Discovery and Integration (UDDI) server. Finding interpolation service is now just a question of connecting to the UDDI server to locate the URL for the service.

UDDI provides three basic functions, popularly known as publish, find, and bind:

- Publish: How the provider of a Web service registers itself.
- Find: How an application finds a particular Web service.
- Bind: How an application connects to, and interacts with, a Web service after it's been found.

A UDDI registry contains three kinds of information, described in terms of telephone directories:

- White pages: Information such as the name, address, telephone number, and other contact information of a given business.
- Yellow pages: Information that categorizes businesses. This is based on existing (non-electronic) standards.
- Green pages: Technical information about the Web services provided by a given business.

When working with a UDDI registry, there are four information types that are important:

- Business information: Contains information about services, categories, contacts, URLs, and other things necessary to interact with a given business.
- Service information: Describes a group of Web services.
- Binding information: The technical details necessary to invoke a Web service. This includes URLs, information about method names, argument types, and so on.
- Information about specifications for services: This is metadata about the various specifications implemented by a given Web service. These are called tModels in the UDDI specification.

Fig. 3.4 shows the how the basic architectural elements of a typical Web service work together.

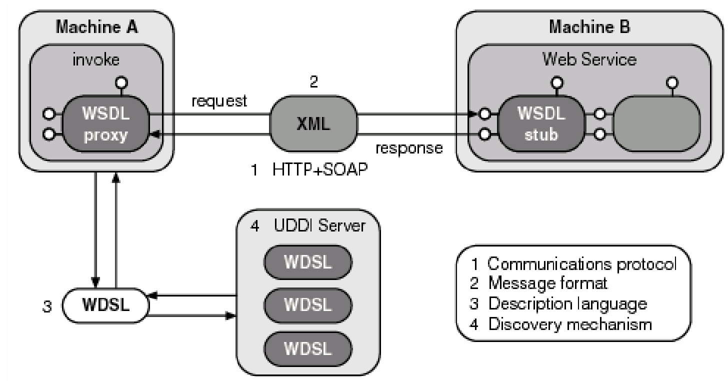

Fig. 3.4: Typical architectural elements from a Web service [Microsoft, 2003]

3.5 Services Interaction Model

Web services are deployed in web servers such that they can be discovered and invoked by any web application, web agent or Web service independently of their implementations. In the scenario of discovering web services, an agent may have different roles including service requester, service provider and service broker. There are three interactions including find, publish and bind which can occur between these agents.

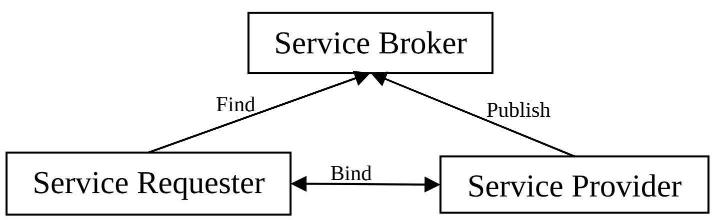

Fig. 3.5: The basic model of service interaction

As illustrated in Fig. 3.5, the service interaction model can be defined by three interactions called publish, find, and bind between three agents called service requester, service provider and service broker.

The service broker enables the service requester to dynamically find service providers or registry where services can be published or advertised. A service provider publishes a description of a service it provides to a service broker. This description (or advertisement) includes a profile on the provider of the service (e.g.

company name and address); a profile about the service itself (e.g. name, category); and the URL (Uniform Resource Locator) of its service interface definition.

The service requester then find the desired service and interprets the meaning of its interface description (typically through the use of meaningful label or variable names, comments, or additional documentation) and binds to (i.e. includes a call to invoke) the discovered service within the application they are developing.

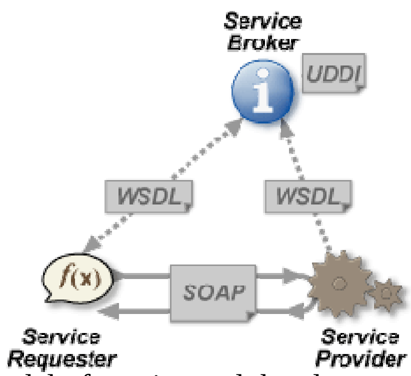

Fig.3.6: The basic model of service and the elements of Web services stack (http://en.wikipedia.org/wiki/Web_service).

As illustrated in Fig 3.6 a mapping can be established between players of service interaction model as well as their three phases on one hand and basic standard protocols for exchanging messages between web services on the other hand. In this case service provider could publish the service on a UDDI server by using WSDL, then service requester can take the advantage of search in UDDI and find appropriate services by using WSDL and finally, the service requester bind to the service provider using SOAP.

3.6 Geo-Services

There are three different views of GIS including Geo-database view, Geo-visualization view and Geo-processing view. In geo-database view a GIS is a spatial database containing datasets that represent geographic information in terms of a generic GIS data model (features, raster, topologies, networks, and so forth).

In geo-visualization view a GIS is a set of intelligent maps and other views that show features and feature relationships on the earth's surface. In the Geo-processing view a GIS is a set of geo-operations that derive new geo-spatial datasets from existing datasets. These geo-operations take information from existing datasets, apply functions, and write results into new derived datasets [ESRI, 2004].

The geo-services can be categorized into geo data services which typically are tightly coupled with specific data sets and offer access to customized portions of that data and geo-operation services which provide operations for processing or transforming data in a manner determined by user-specified parameters. The geo-operation services are not associated with specific data sets [Alameh, 2003]. In addition, they can be data nonspecific and used again. Each of these services is explained in more details in the next sections.

3.6.1 Geo Data Services

Geo data services offer access to customized portions of the data. Examples of data services include the Web Mapping Service (WMS), which produces maps as two-dimensional visual portrayals of geospatial data; the Web Coverage Service (WCS), which provides access to un-rendered geospatial information as needed for client-side rendering; and the Web Feature Service (WFS), which lets a client retrieve geospatial data encoded in Geography Markup Language (GML). These services can be implemented according to The OpenGIS WMS, WCS and WFS Implementation Interface Specifications respectively [OGC, 2000].

The OpenGIS WMS Implementation Interface Specifications provide guidelines for current Web Map Servers with the specifications of HTTP contents and Uniform Resource Locators (URLs) communication syntax. The WMS Specifications also lay out the major tasks of Internet Map Servers.

The OpenGIS WMS Specifications standardize the syntax of the URLs for Web Map Servers and focus on the three major tasks: produce a map, answer basic queries about the content of the map, and tell other programs what maps it can produce and which of those can be queried further. In general, a standard web browser can ask a Map Server to do these things just by submitting requests in the form of Uniform Resource Locators (URLs). The content of such URLs depends on which of the three tasks is requested [OGC, 2000].

In order to accomplish these three major tasks, the OpenGIS WMS Implementation Interface Specification provides three types of interface: GetMap, GetFeature, and GetCapabilities.

1. The Map Request (GetMap) Interfaces. The design of Map Request Interfaces focuses on the display and production of Web-based map services. "To produce a map, the URL parameters indicate which portion of the Earth is to be mapped, the coordinate system to be used, the type(s) of information to be shown, the desired output format, and perhaps the output size, rendering style, or other parameters" [OGC, 2000].

2. The Feature Request (GetFeature) Interfaces. The Feature Request Interfaces identify the request mechanisms for map contents and feature attributes. To query the content of the map features, the URL parameters indicate what map (layer) is being queried and which location on the map is of interest (X, Y coordinates).

3. The Capabilities Request (GetCapabilities) Interfaces. The Capabilities Request Interfaces are used to provide extensive map services, such as catalog services or metadata queries, in addition to the basic map display and attribute query.

For example the following hypothetical URL requests the capabilities of web map services in a server called localhost:

http://localhost/test/basic-wms2.py?request=capabilities&wmtver=1.0.0

The response of this request is illustrated in Fig. 3.7.

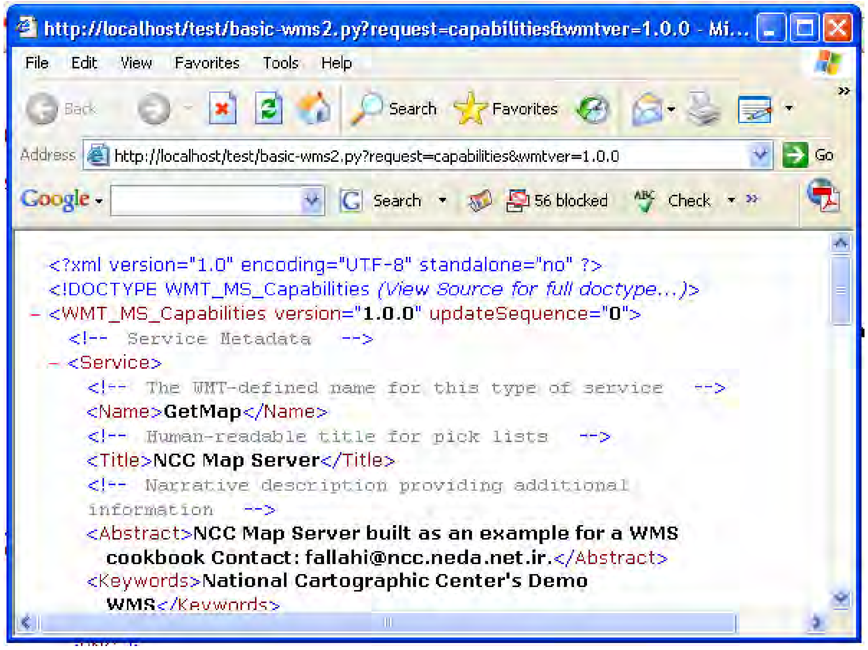

Fig. 3.7: The response of map server to GetCapabilities operation

And the following hypothetical URL requests the image on that server:

http://localhost/test/basic-wms2.py?WMTVER=1.0.0&REQUEST=map&LAYERS=RELIEF&STYLES=default&SRS=EPSG:4326&BBOX=-2.197265625,39.55078125,20.302734375,50.80078125&WIDTH=256&HEIGHT=128&FORMAT=JPEG

The response to this request by map server is illustrated in Fig. 3.8.

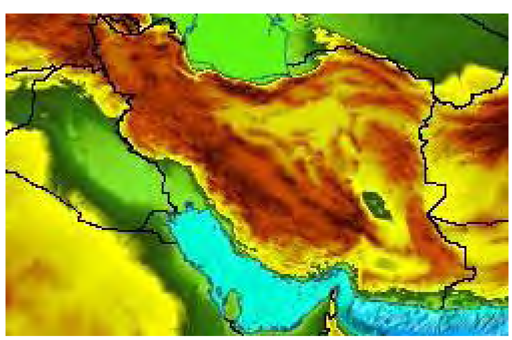

Fig. 3.8: The response to GetMap operator

Geo data services such as Web Map and Web Feature service, as generic services, must provide additional, descriptive metadata in order to be useful. A requester may interact with two different Web Feature Services in exactly the same way (the WSDL is the same), however the Web Feature Services may hold different data. Requestors must be able to query information services that encode (in standard formats) all the necessary information, or metadata, that enables the requester to connect to the desired service. This is an example of the very general problem of managing information about geo data services.

3.6.2 Geo-Operation Services

Geo-operation services provide operations for processing or transforming data in a manner determined by user-specified parameters. According to definition from ISO 19119 geo-services can be defined as a collection of geo-operations, accessible through an interface.

Geo-operation services can be classified into primitive, or atomic services, and complex or composite services [OWL-S, 2004]. Atomic services are ones where a single Web-accessible computer program, sensor, or device is invoked by a request message, performs its task and perhaps produces a single response to the requester. With atomic services there is no ongoing interaction between the user and a primitive service. A complex service is composed of multiple primitive services, often requiring an interaction or conversation between the user and services, so that the user can make decisions [McIlraith et. al., 2001]. For example, a service that returns a postal code or the longitude and latitude when given an address or another service that create a new buffer feature around a given polygon feature at a user specified distance would be in this category.

From application point of view the atomic geo-service can be classified into generic services such as interpolation or specialized services such as flow direction which is applicable in hydrology domain.

In the case of complex services that are called chaining services, there are three types of chaining defined in ISO 19119 and OGC [ISO, 2001] including user-defined (transparent) that the human user defines and manages the chain, workflow-managed (translucent) that the human user invokes a service that manages and controls the chain where the user is aware of the individual services in the chain and aggregate (opaque) that the human user invokes a service that carries out the chain, where the user has no awareness of the individual services in the chain.

A workflow expressing the composition of atomic services can be defined by using appropriate control constructs. This description would be grounded on a syntactic description such as BPEL4WS (Business Process Execution Language for Web Services) (http://www.ibm.com/developerworks/library/ws-bpel).

3.7 Chapter Summary

This chapter discussed about the distributed computing architecture based on loosely coupled services. Following the discussion, the chapter concentrated on need of communication between service requester, service provider and broker. In this regard the basic standard protocols including WSDL, SOAP and UDDI were explained. To perform communication the exchanged messages according to these standards should be understood by service requester, service provider and service broker. However, the chapter discussed that these standards establish the syntactic interfaces and protocols for behavior of web services and suffer from lack of semantic.

Then Geo-services and their classification including geo data services and geo-operation services were discussed. This research work focuses on geo-operation services which can produce a new geospatial data from other geospatial data sets. It especially pays attention to the field-based geo-services which will be discussed on the next chapters.

The next chapter after discussion about semantic interoperability pay attention to the concepts related to input and output data of geo-services whose are widely used in environmental modeling disciplines. The concentration is on the semantic ambiguities as barriers for discovering and composition of field-based geo-services which are raised due to implicit details related to inputs, outputs or operations of geo-services. In this regards, the semantic ambiguities related to the sample geo-services are studied.

The success of the distributed computing architectures based on loosely coupled geo-services for automatic discovery and composition depends to a large extent on specifying the intended meaning of their terms in machine-understandable form which is the contribution of this research and concentrated on it in the rest of thesis.

Chapter 4 **Semantic Issues Associated with Field-Based Geo-Services**

4.1 Introduction

Distributed computing architecture based on loosely coupled geo-services is a new approach to implement the linking geo-services and environmental models. The fundamental tasks including publishing, discovering and invoking geo-services can be fulfilled in this architecture by performing communication and exchanging message between service requester, service provider and broker. The message can be exchanged in a standard manner according to a set of computer networking protocols including UDDI, WSDL, and SOAP. These set of protocols, however, support the syntactic interoperability between geo-service requester (modeler) geo-service provider (GIS) and broker (publisher) rather than semantic interoperability.

This chapter discusses natural system and relation between environmental model and GIS. Then it pays attention to conceptualization of natural environment and especially focuses on how environmental modelers or scientists conceptualize natural environment. In this regards notion of field and field-based geo-services are introduced. Semantic ambiguities of field-based geo-services are investigated by illustrating samples of field-based geo-services. Then classifications of semantic ambiguities which may arise due to lack of details of the geo-services are explained. The chapter then discusses interoperability and its levels and concentrates on semantics issues as a mean for promoting interoperability in interaction model between field-based geo-services.

4.2 Environmental Models

Many physical processes occur in natural system and environmental models attempt to model these processes. For example, water cycles processes and their relations with other components are conceptually illustrated in Fig. 4.1.

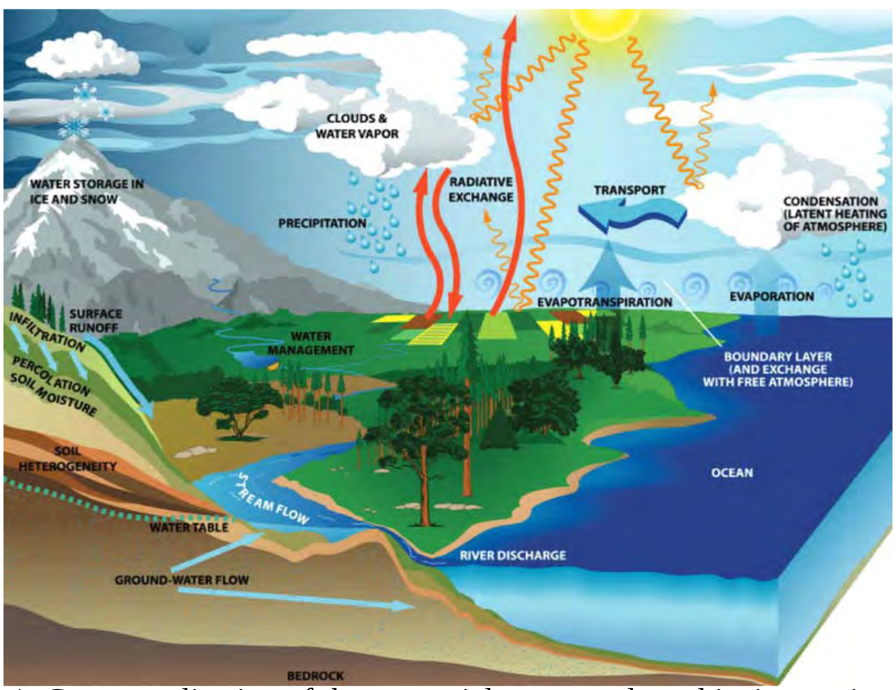

Fig. 4.1: Conceptualization of the terrestrial water cycle and its interactions with all other components of the earth-climate system Source: Houser and Schlosser, NASA HGSFCH [CCSP, 2003].

When physical processes are modeled, it is concentrated on a set of states which describe the conditions of natural system under study [Casti, 1989]. A model is a formal representation of the relationships between defined quantities or qualities [Jeffers, 1982]. For example the amount of flow in water surface flow depends on the height variations of watershed surface or permeability in infiltration process depend on soil type.

Feng et al (2004) argued that processes such as infiltration and evapotarnspiration are perdurants and have certain relations with endurants such as catchment.

Endurants are characterized as entities that are ‘in time’, they are ‘wholly’ present at any time of their existence. On the other hand, perdurants are entities that ‘happen in time’; they extend in time by accumulating different ‘temporal parts’ [Masolo et al, 2003].

4.2.1 Environmental Models and GIS

The determination of the states such as slope or land cover is one of the challenges in environmental modeling that encourage the modeler to use GIS. GIS can be considered as a repository of geospatial data and geo-operations to manipulate geospatial data as well [Frank, 2005]. With refer to Fig 4.2 the geo-operations produce the new geospatial data from old one which can be used as a quantitative or qualitative state of the physical process.

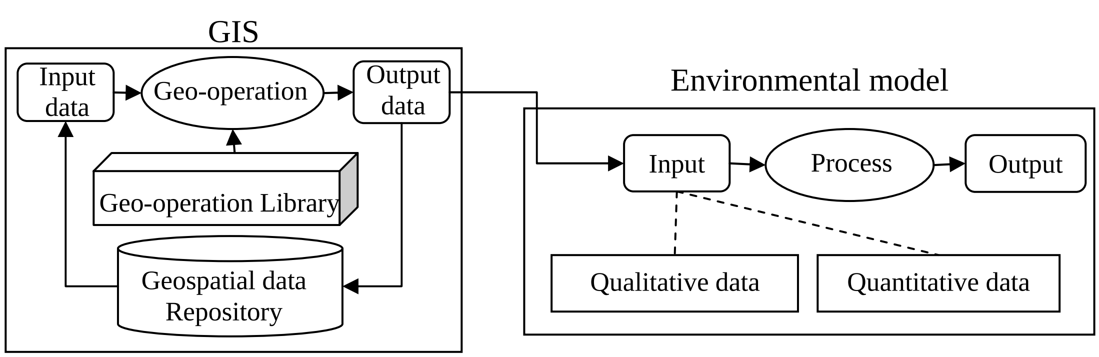

Fig. 4.2: The relation between the environmental model and GIS

Formalizing the states of the physical process depend on the conceptualization of scientists from natural system.

4.2.2 Conceptualization of Natural System

Conceptualization is formal structure of (a piece of) reality as perceived and organized by an agent, independently of the vocabulary used and the actual occurrence of a specific situation [Borgo et al, 2005].

In general, there are two broad and opposing conceptualization about the natural system called object and field conceptualization [Couclelis, 1992; Peuquet et al, 1999; Galton, 2001; Smith and Mark, 2003; Worboys and Duckham, 2004]. The people have object-based conceptualization from their environment. In this conceptualization, they use prototypical objects and the ones learn about in infancy for things in the environment [Mark and Smith, 2001]. These things can move or be moved from one place to another such as pets, chairs, spoons, bottles, pieces of fruit.

In contrast, many scientists conceptualize the environment as fields. The characteristics of environmental modeling are best explained by field, rather than object [Smith and Mark, 2003]. The field-based approach treats the environment as a collection of fields. Each field defines the spatial variation of an attribute as a function from the set of locations to an attribute domain [Worboys and Duckham, 2004]. Patterns of temperature, population density, pH of the soil, soil type or tree-coverage fit neatly into this conceptualization.

In the field conceptualization, each state of the physical process can be observed and measured in each location and according to the measurement a value can be associated to it and represented by a number. These numbers may describe soil type, land use type, elevation, distance of each location from a phenomenon, noise levels from an airport, and radiation levels from a nuclear plant.

This research also focuses on GIS operations which can operate on field-based spatial data. It means that GIS operations which perform a mapping between new and old values of physical fields.

4.2.3 Field-Based Geo-Operations

A field-based geo-operation takes as input one or more fields and returns a resultant field. In the other words, geo-operations are mappings or transformations applied to the fields (e.g. altitude) in order to derive new field (e.g. flow or slope) which can be used in an environmental model (e.g. erosion model).

In the GIS packages there is a rich suite of field-based geo-operations. Some of field-based geo-operations have mathematical expressions such as equality, boolean, relational, and arithmetic operations. The other geo-operations perform functions in order to fulfill complex tasks.

The map algebra in cartographic modeling approach consists of a set of primitive operators that induce a change in the properties of the fields, where the change in properties is calculated on the basis of four major operations given as focal operations, incremental operations, local operations and zonal operations [Tomlin, 1990]. For example, the IncrementalDrainage is an operation of the cartographic modeling which accepts altitude whose values represents the elevation in each location and produces upstream whose values represent the amount to the paths of runoff over the altitude surface.

Geomorphological and hydrological operations as specific domain operations include functions for visibility analysis, catchment analysis and routing of transport (drainage) of material in a catchment.

4.3 Semantic Ambiguities in the Field-Based Geo-Services

Semantic of geo-operations promise to provide solutions to the challenges associated with automated discovery using service-based systems. The following sections clarify the semantic ambiguities associated with geo-services.

4.3.1 Geo-Services in Environment Modeling

Suppose an environmental modeler concerns to identify all locations in an areas that are forested, owned by the state government, and have a certain runoff rating [ArcGIS, 2006]. The modeler needs services that overlay the field-based datasets in order to produce susceptibility map.

Fig. 4.3 shows a solution for this problem. The workflow of the model contains geo-operations such as "EQUAL" as well as "AND" operations and geospatial data which are used as inputs and outputs of geo-operations.

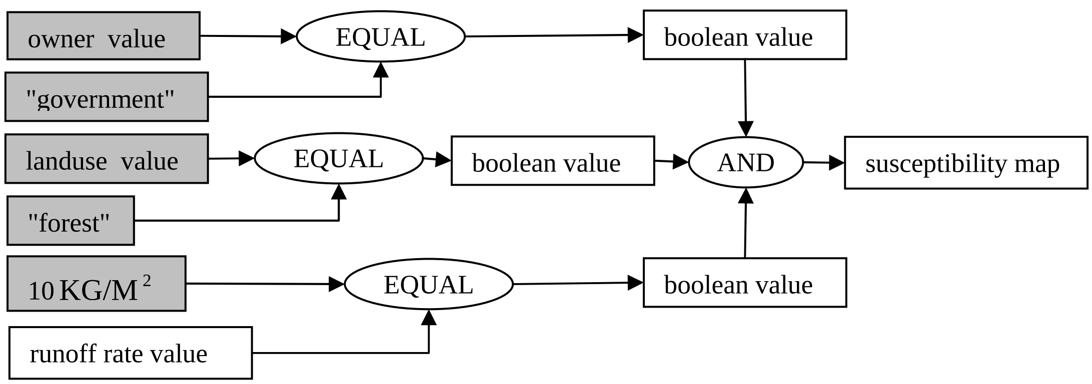

Fig. 4.3: The workflow of the model. The boxes show primary and derived field-based geospatial data. The ellipses show the field-based geo-operation.

To make the model, the modeler must discover appropriate geo-service. For this, the modeler must describe the desired geo-service precisely. Suppose a modeler is looking for a geo-operation which produces runoff rate value. There may be several geo-services which produce the runoff as output by using interpolation operation or according to a formula like the following [NCGIA 1998]:

$$R = \frac{S \cdot C \cdot P}{160} \qquad (1)$$

where S is the surface slope, C the ground cover coefficient, P the Precipitation in millimeters, and R indicates runoff volume of water, (in liters per square meter).

As it is illustrated in Fig. 4.4, the equation 1 can be implemented as a composite service consist of atomic services such as "SLOPE", "RECLASSIFY", "MULTIPLE" and "DIVIDE" operators.

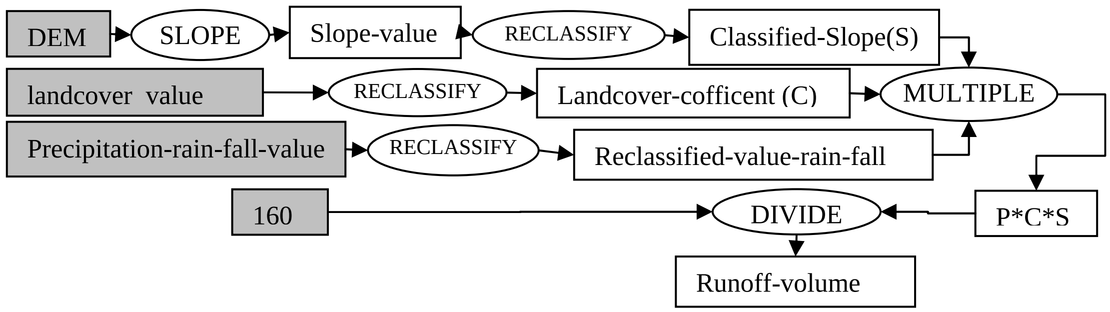

Fig. 4.4: The workflow for runoff volume

4.3.2 Semantic Ambiguities

With refer to above example the unit of runoff rate in the model is *kg/m²* while the unit of runoff volume produced by geo-services according to equation is *l/m²* . If details in the descriptions of requested and provided geo-services such as unit of measure and type of measurement are missing the modeler may select the wrong geo-service.

In the case of interpolation service, the interpolation algorithm depends on the measurement type of geospatial data used as input of the geo-service. "Numerous algorithms for point interpolation have been developed in the past. The selection of an appropriate interpolation model depends largely on the types of data, the degree of accuracy desired, and the amount of computational effort afforded" [Lam 1983]. The interpolation operation for runoff rate on a ratio scale is different from the interpolation for land use type on a nominal scale. Continuous fields of categorical data cannot be generated from points using any of the mathematical interpolation techniques since values cannot be interpolated between classes [Kemp 1993].

A modeler could confuse logical "AND" with addition and discover an addition geo-service (Fig. 4.5):

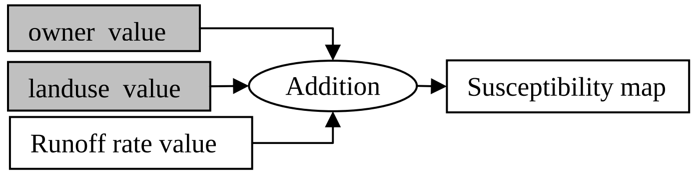

Fig. 4.5: The workflow of model with "Addition" operation

However, for a numerical addition operation, the measurement type and the unit for its input and output data do not fit to what is provided. Land use value and owner value have nominal type and no units; They can not be numerically added to runoff rate value with a ratio type and unit of kg/m^2. The result would be meaningless.

4.4 Classification of Semantic Ambiguities in Field-Based Geo-Services

In general, various possible semantic problems may be occurred due to implicit details in field based geo-services. Some of them are explained as follow:

4.4.1 Discrepancy in type of measurement

The type of measurement is one of the characteristics of physical qualities which determine criteria under which GIS operations are allowed to operate on field-based spatial data. For example, with refer to Fig. 4.5. Addition operation can be an attempt to solve the environmental problem. However, the result of adding land use value, owner value and runoff rate value is meaningless since they have different type of measurement.

4.4.2 Difference in name

Different name may be used for calling an aspect of phenomena in GIS and environmental discipline. For example a field may be called upstream in GIS, which is the runoff generated at all points upstream of a location while in the hydrologic domain, they may use runoff or outflow for this concept or as it is mentioned in above example whether or not the runoff volume is the same as runoff rate. These heterogeneities in name for a parameter or variable must be exactly identified for users of different domain to avoid them of incorrect use of the parameter or variable in models.

4.4.3 Lack of unit of measure

The lack of measurement units' semantics of field-based spatial data in geo-operations prevents requester from understanding the input and output of service. The common sense supposition of the user makes the requester capable to understand that the unit is used for measuring distance and not for some other measurement. However, in a varied application domain, the units may be used for other purposes, such as measuring temperature, precipitation, pressure, weight, etc. The user cannot differentiate the meaning of the unit by its name.

In other case the value of a property may be represented in different domain by different unit of measure. For instance population density may have person per acre for inhabitants of a city, person per square mile for inhabitants of a city, person per flat for residents of a building, person per 100 cubic meters for residents of a building and patient per room for patients in a hospital.

These heterogeneities in unit of measurement for a parameter or variable must be explicit for users of different domain to avoid them of incorrect use of the parameter or variable in models.

4.5 Semantic Ambiguities in Standard Protocols

The industrial standard protocols such as WSDL, SOAP and UDDI used in service oriented architecture is not able to provide sufficient means for automatic service discovery.

The WSDL description as key concept in discovering geo-services establish syntactic interfaces and protocols for invoking system behavior, but do not specify the intended meaning of their terms in machine-understandable form and prevents the service requester from understanding the service semantics.

WSDL file provide the signature of the operations of the service, that is, the name, parameters and the types of parameters of the service. Trying to discover services by name may not be always very meaningful since a service name could be anything and in any language. It also does not support the definition of logical constraints between its input and output parameters.

Protocols such as WSDL and SOAP contain data type for data structure used as input and output of geo-services. However in practice, “knowing the type of a data structure is not enough to understand the intent and meaning behind its use” [W3C, 2004]. These protocols do not contain the intended meaning of their terms and thus suffer from the lack of semantic interoperability.

UDDI does not represent service capabilities. The tModels they use only provide a tagging mechanism, and the search performed is only done by string matching on some fields they have defined. It is not possible in UDDI to enforce a relationship between the service names and their functionality. Thus, it is of no use for locating services on the basis of a semantic specification of their functionality [Waris, 2005].

Semantics issues have basic role for promoting interoperability in communication interaction model of field-based geo-services. The next section discusses about interoperability and its levels especially semantic level.

4.6 Interoperability

The communication between requester, publisher and provider of geo-services, in a service interaction model, can be performed by establishing interoperability between systems. The communication in an interoperable environment can enhance the automatic discovery of geo-services.

4.6.1 Definition and Classification of Interoperability

Interoperability can be defined in a technical way as (ISO TC204, document N271):

The ability of systems, units, or forces to provide services to and accept services from other systems, units or forces and to use the services so exchanged to enable them to operate effectively together.

In the other words two components X and Y (e.g. modeler and GIS) can interoperate (are interoperable) if X can send requests R for services to Y, based on a mutual understanding of R by X and Y, and if Y can similarly return mutually understandable responses S to X (see Fig 4.6) [Brodie, 1992].

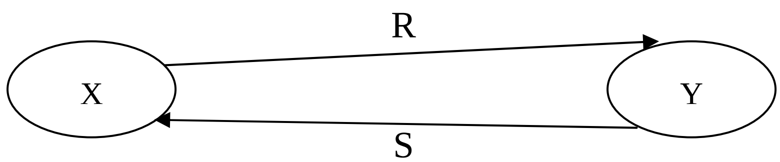

Fig. 4.6: The interoperability between X and Y components.

Semantic of geo-services promise to provide solutions to the challenges associated with automated discovery using service-based systems. Bishr (1998) lists six levels of interoperability in communication between two systems; semantic interoperability is at the highest level (Fig. 4.7). Semantic description of capabilities and properties of field-based geo-services is crucial for automatic discovery of geo-services.

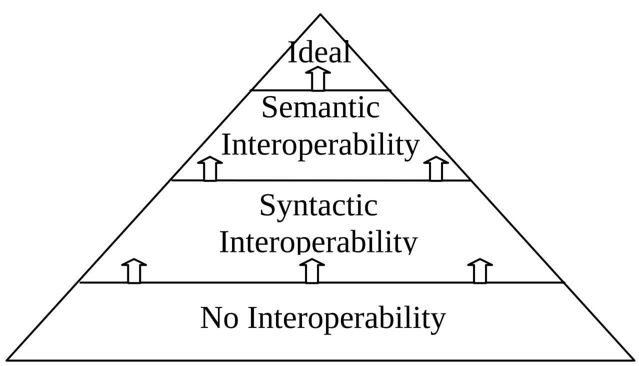

Fig 4.7: The pyramid of increasing interoperability

As it is illustrated in Fig 4.7 the interoperability is increased from down to the top. Increasing interoperability may increase the potential customer for geo-service provider and enhance the automatic discovery of fit to use geo-services and quality of geo-services for requester.

4.6.2 Semantic Interoperability

The term “semantics” here refers to the meaning of expressions in a language [Kuhn 2005]. Expressions can be single symbols (the “words” of a language) or symbol combinations. The meaning triangle defines the interaction between symbols or words, concepts and things of the world (see Fig. 4.8).

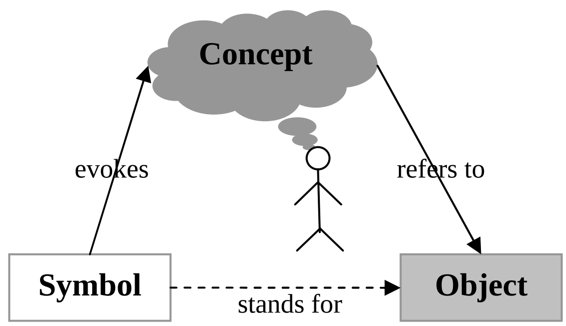

Fig. 4.8: The Meaning Triangle [Ogden et al, 1923]

The meaning triangle illustrates the fact that the relationship between a word and a thing is indirect and words cannot completely capture the real meaning of a thing. For example, the term “jaguar” can evoke a concept of an animal, car, or jet fighter. The correct linkage is only accomplished when an agent interpret the word invoking a corresponding concept in a context picking out the intended interpretation and discarding others. The corresponding concept establishes the proper linkage between symbol and the appropriate thing in the world. Thus linkage between object, word, concept and context can be defined as follow.

$$object = word + (concept + context)$$

The corresponding concept which is concept plus context is shaped by human experience with real-world entities.

Fig. 4.9 represents a vision of future of the web. It holds the belief that when both the syntactic and semantic descriptions of web resources are available, computers may without user aid discover and invoke services.

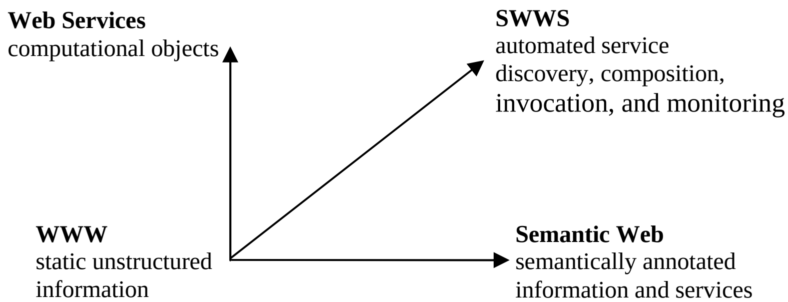

Fig. 4.9: Future with Semantic Web Services [Davies et. al., 2004]

However the formal semantic descriptions will always remain partial compared to human conceptions. Therefore degree of automatic discovery and invocation of the services will depend on the semantic description of concepts.

4.7 Chapter Summary

This chapter introduced the natural system and its relation to GIS. Field-based and object-based approaches have been investigated for conceptualization of natural system in such a way that the scientists use the former one to treat the natural system and the latter is used by nave people for describing the environment. This chapter focused on distributed properties related to field-based geospatial data. In this regard semantic ambiguities raised due to implicit details of geo-services have been discussed. The role of interoperability in communication between requester (modeler) and provider (GIS) of geo-services, especially at semantic level, have been investigated.

The next chapter represents a definition for ontology and proposes layered-base structure of ontologies for solving semantic ambiguities of the geo-services. An upper ontology is developed by investigating and combining general concepts from existing ontologies. The domain specific ontologies containing concepts and relationships for geo-services are built and aligned to the new general concepts.

Role of these ontologies is reducing ambiguities of reasoning systems while discovering geo-services by committing to them.

Chapter 5 **Layered-Base Structure of Ontologies**

5.1 Introduction

This chapter introduces ontology as solution to describe semantic ambiguities. It proposes a layered-base structure of ontologies which reduces semantic ambiguities and promotes interoperability between geo-services in distributed computing architecture. The ontological structure is a knowledge base for discovering geo-services and consists of four layers of ontologies. Role of common ontologies is to reduce ambiguities of reasoning systems while discovering geo-services by committing to them. After reviewing existing ontologies, the ontology containing descriptions of unit of measure and type of measurement is developed.

Then, along with discussing problematic aspects of OWL-S (Ontology Web Language for Services), the core ontology of geo-services is proposed for overcoming the drawbacks of OWL-S. The ontology of measurement theory is also developed which contains concepts for describing measurement scale and unit of measure of the input and output of the geo-services. A contribution of the research is to develop these ontologies and align them to the upper ontology. An upper ontology at the top layer of the structure is proposed which contains general concepts for discovering of field-based geo-services. In order to align core ontology of geo-services to upper ontology the Descriptions and Situations (D&S) ontology is used to fill conceptual gaps between core ontology of geo-services and upper ontology.

5.2 An Ontology as a Means of Describing Semantic

Describing semantics means to fix the intended meaning of vocabulary terms. Standardized vocabularies are only a partial solution for semantic heterogeneities, because they tend to be ambiguous or circular. The meaning triangle establishes a linkage between an entity in the world and its symbol through a concept. Conceptualization is a description of (a piece of) reality as perceived and organized by an agent, independently of the vocabulary used and the actual occurrence of a specific situation [Borgo et al. 2005].

"An ontology is a specification of a conceptualization" [Gruber 1993].Ontology (capital "O") is a philosophical discipline which studies the nature and structure of possible entities. An ontology (lowercase "o") is a specific artifact designed with the purpose of expressing the intended meaning of a vocabulary in terms of the nature and structure of the entities it refers to [Borgo et al. 2005]. The ontologies can be used to negotiate meaning, either for enabling effective cooperation between multiple artificial agents, or for establishing consensus in a mixed society where artificial agents cooperate with humans. An ontology consists of axioms that express the meaning of terms for a particular community. Logical axioms are the means to specify a set of constraints, which declare what should necessarily hold in any possible world. They also introduce concepts, relations and their taxonomic hierarchies. An ontology typically contains two distinct parts:

names for important concepts and background knowledge/constraints in the domain [Drummond 2005].

5.2.1 Classification of Ontologies

Ontologies can be classified according to their level of details and their level of dependence on a particular task or point of view [Guarino 1997].

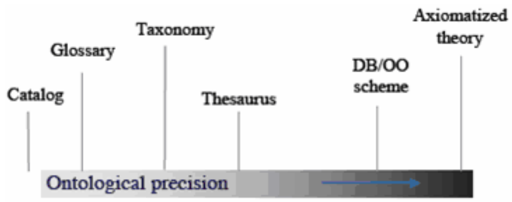

Fig.5.1: Levels of ontological precision from [Borgo et al. 2005].

The level of detail can be classified by the ontological precision from catalog to axiomatized theory (Fig.5.1). The dependence on a particular task or point of view distinguishes between top-level, domain, task and application ontologies (Fig.5.2).

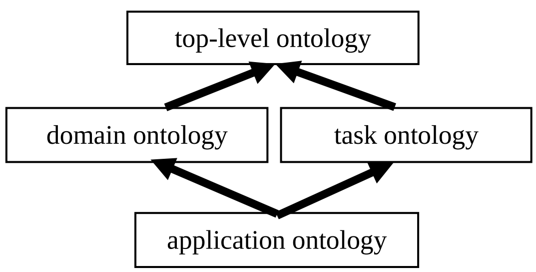

Fig.5.2: Kinds of ontologies. Thick arrows represent specialization relationships from [Guarino 1997]

In order to perform matching between ontologies of requested and provided geo-services at the application level, there must be an agreement between GIS and environmental modelers about general and domain specific concepts. In this research, the agreement is achieved by means of the proposed shared ontologies. The contribution is to develop the ontologies of the measurement theory and core ontology of geo-services at the domain level in order to describe concepts related to measurement scale and unit of measure which are crucial for field-based geo-service discovery.

These ontologies have a taxonomic structure which is lattice-like structure containing those entities in ontology that represent the concepts or categories found in the world that is being modeled. Within the taxonomy, concepts or categories are related by subsumption relationships which are reflexive, transitive and anti-symmetric [Farrar and Bateman, 2005]. Subsumption relationships represent subconcept / superconcept relationships.

5.3 Existing Ontologies

Building ontology from scratch is time consuming and labor intensive task. Therefore, the related concepts in the context of geo-services and environmental models in various existing ontologies such as Suggested Upper Merged Ontology (SUMO), MathEng ontology, General Formal Ontology (GFO), Generalized Upper Model (GUM) and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) are investigated.

5.3.1 Suggested Upper Merged Ontology (SUMO)

The SUMO contains abstract concepts needed for ontologies produced by Teknowledge's Knowledge Systems group [Nichols and Terry, 2003]. SUMO’s purpose is to promote data interoperability, information search and retrieval, automated inference, and natural language processing. Fig. 5.3 taken from the SUMO documentation lists the most general modules assumed.

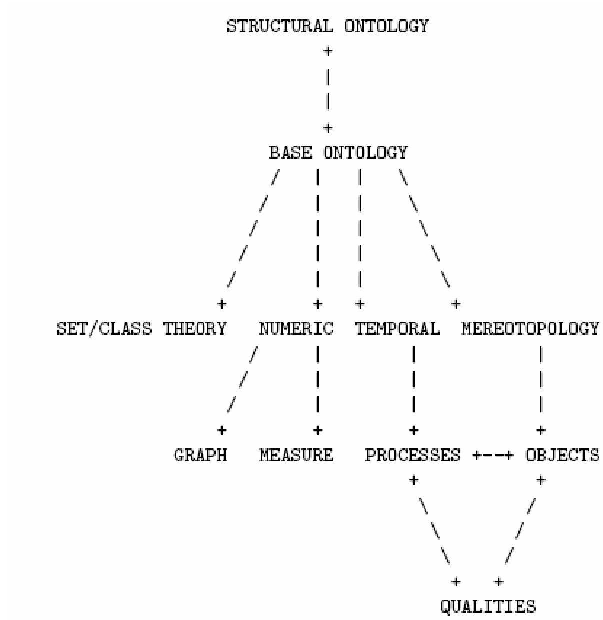

Fig. 5.3: Overall modular organization of SUMO

Process is intuitively, the class of things that happen and have temporal parts or stages. The hierarchy structure of process in SUMO is depicted in Fig. 5.4.

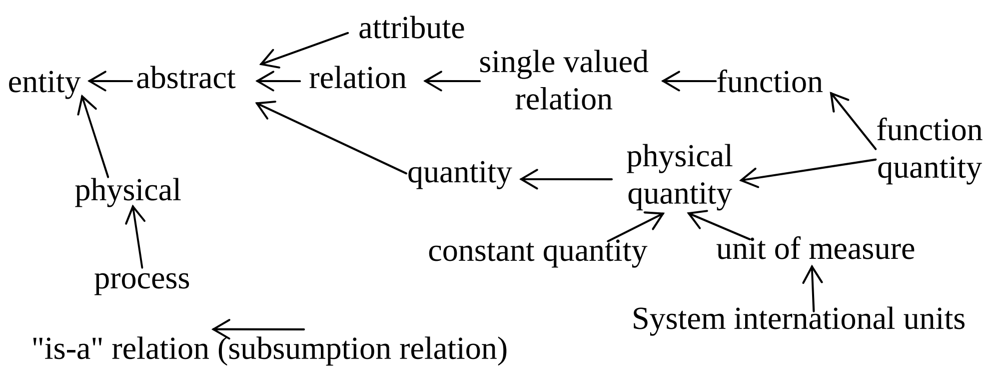

Fig. 5.4: The hierarchy structure of quantity and its related concepts in SUMO
http://virtual.cvut.cz/kifb/en/concepts/_function_quantity.html
http://virtual.cvut.cz/kifb/en/concepts/_attribute.html
http://virtual.cvut.cz/kifb/en/concepts/_process.html

The physical concept is defined as an entity that has a location in space-time. Properties or qualities are distinguished from any particular embodiment of the properties/qualities in a physical medium. Instances of abstract can be said to exist in the same sense as mathematical objects such as sets and relations, but they cannot exist at a particular place and time without some physical encoding or embodiment. Quantity is any specification of how many or how much of something there is. It is subconcept of abstract and disjoint from attribute. The positions of attribute and abstract concepts and quantity concepts in SUMO are illustrated in Fig. 5.4.

MeasureFn is a function used with a real number and an appropriate unit of measure to denote a measured quantity. A FunctionQuantity is a function that maps from one or more instances of ConstantQuantity to another instance of ConstantQuantity. For example, the velocity of a particle would be represented by a FunctionQuantity mapping values of time (which are ConstantQuantities) to values of distance (also ConstantQuantities). Concept hierarchy of FunctionQuantity and other related concepts is depicted in Fig. 5.4.

System international is a complete metric system of units of measurement for scientists; fundamental quantities are length (meter) and mass (kilogram) and time (second) and electric current (ampere) and temperature (kelvin) and amount of matter (mole) and luminous intensity (candela).

5.3.2 EngMath Ontology

EngMath shares and reuses engineering models, among engineering tools and their users [Gruber and Olsen, 1994]. The ontology built on abstract algebra and measurement theory, adapted to meet the expressive needs of engineering modeling. The ontology is divided into several theories which form inheritance hierarchies (a theory inherits the definitions of its parent theories) as it is illustrated in Fig 5.5.

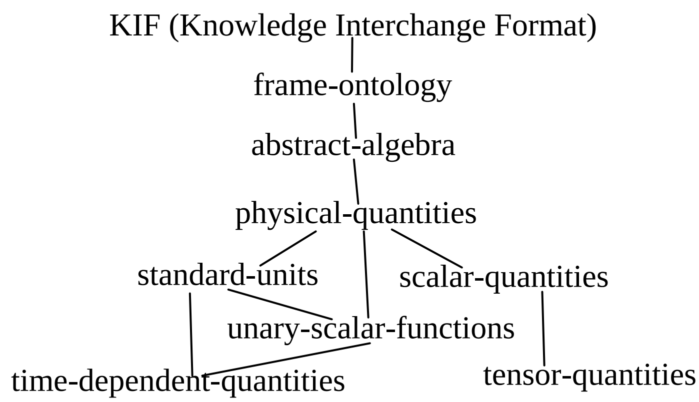

Fig.5.5: Inheritance hierarchy between theories in EngMath ontology
http://www-ksl.stanford.edu/knowledge-sharing/papers/engmath-tree.html

In EngMath physical quantities such as the length or the velocity are routinely modeled by variables in equations and numbers as its values. Constant quantity concept which is defined as a constant value of some physical-quantity, like 3 meters is subconcept of physical quantities and unit-of-measure is subconcept of constant quantities.

A function is a mapping from a domain to a range that associates a domain element with exactly one range element. It is a subconcept of relation and relation is subconcept of sets. A function-quantity is a function that maps from one or more constant-quantities to a constant-quantity. This is subconcept of function and physical-quantity. In addition the constant-quantity is also subconcept of physical quantity.

As a consequence concepts in EngMath have almost the same name, definition and position as concepts in SUMO. However the differences is that various theories of EngMath are committed to KIF as a logical language for expressing the ontology (see Fig. 5.5) while in SUMO there are different ontology modules which can be plugged into base ontology (see Fig. 5.3).

5.3.3 General Formal Ontology (GFO)

The GFO is a component of the Integrated System of Foundational Ontologies (ISFO), and ISFO is a part of the Integrated Framework for the Development and Application of Ontologies (IFDAO). The predecessor of IFDAO was the GOL project which was launched in 1999 as a collaborative research effort of the Institute of Medical Informatics, Statistics and Epidemiology (IMISE) and the Institute of Informatics (IfI) at the University of Leipzig [GFO, 2006].

The abstract top level (ATO) of GFO contains mainly two meta-categories: set and item. In this ontology idea is that the entities of the (real) world are subconcept of item which are divided into categories and individuals, i.e. everything in an ontology is either a category or an individual, and individuals instantiate categories. Moreover, among individuals it is distinguished between objects, attributes, roles and relators. Objects are entities that have attributes, and play certain roles with respect to other entities. Examples of attributes are particular weights, forms and colors. A sentence like “This rose is red.” refers to a particular object, a rose, and to a particular attribute, red.

With refer to Fig 5.6 occurrents are individuals. It centers around the more intuitive notion of processes. Some examples of occurrents include writing a letter; sitting in front of a computer viewed as a state extended in time. The category of processes captures those entities that develop over time or unfold in time.

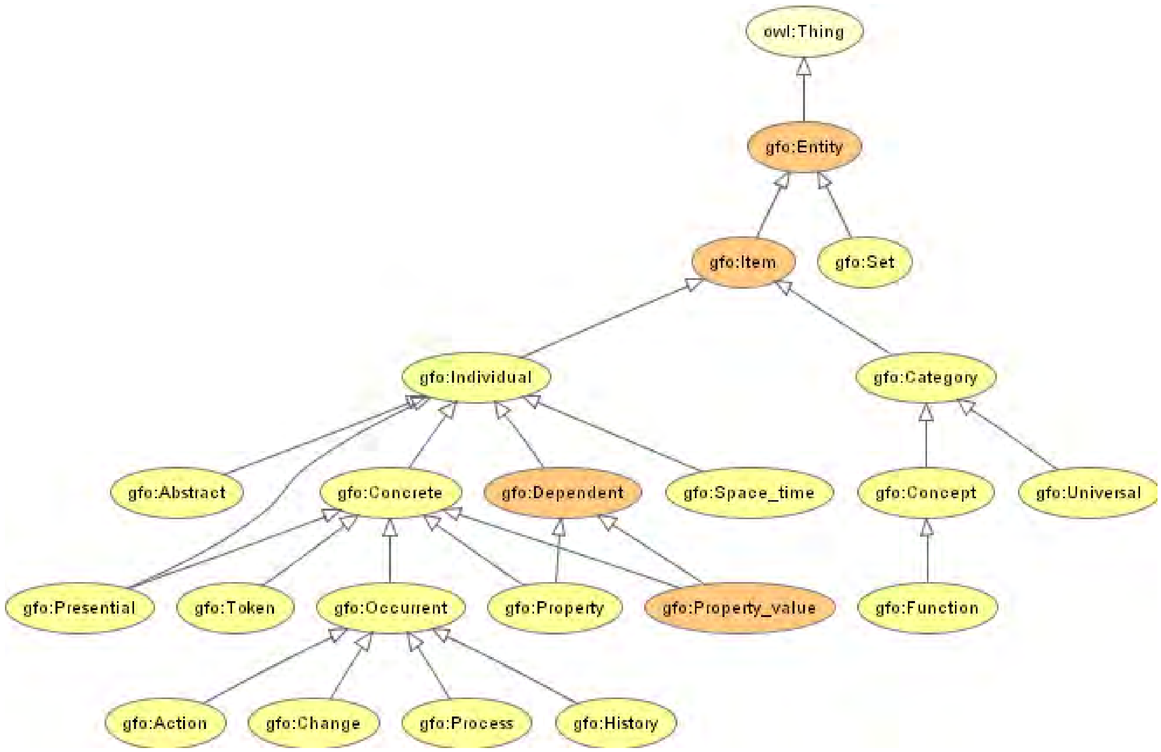

Fig. 5.6: The top-level categories of the General Formal Ontology (GFO) http://www.onto-med.de/gfo.owl

In GFO properties consists of two parts. First, it is distinguished between abstract property universals and their concrete instances, which are called property individuals. Second, both property universals and property individuals must be distinguished from their respective values [GFO, 2006]. For example in the phrases likes “the direction of wind” and “from west” the former phrase refers to a certain aspect of the wind while the later one refers to a value of the wind direction property.

An individual entity has a property means that there is a quality individual which is an instance of the property universal and that the property individual inheres in its bearer. So that the “direction of the wind on that area” is a property individual that inheres in the wind on that area while “direction” is a property universal, of which the quality is an instance. Values of property universals usually appear in groups which are called value structures or measurement systems.

Each of these structures corresponds to some property universal. More intuitively, one could say that the property may be measured with respect to some measurement system. For instance, direction may be measured with the values “to east” or “from west” which are the elements of one value structure. The notion of a value structure of a property is similar to a quality dimension in [Gärdenfors, 2000]. Further, value structures are related to quality spaces in [Masolo et. al., 2003].

5.3.4 Generalized Upper Model (GUM)

The GUM is a general task and domain independent `linguistically motivated ontology' intended for organizing information for expression in natural language. The categories of the ontology enforce a consistent modeling style on any domain which is also appropriately guaranteed for flexible expression in natural language.

Qualities are properties of SimpleThings and Processes. They participate in property ascription relations. Roughly speaking qualities include anything that can be expressed by an English adjective or adverb. Material world qualities can be thought of as those qualities which are evident when the referent is looked at, weighed, measured, etc. Examples include: `heavy', `blue', `readable', `efficient', `maintainable'. The referents or bearers of these qualities are things. The material world quality is divided to dynamic, stative, scaleable, nonscaleable, polar and taxonomic quality. A NonScaleableQuality is either possessed by an object or not. `Empty' is a nonscaleable quality. A quality is scaleable if an object may possess it to varying degrees. For example, `heavy' is a scaleable quality. We can describe objects as being `very heavy', or `more' or `less' heavy than other objects. GUM ontology did not distinguish between a quality and its value. In the mentioned examples heavy or blue are considered as quality however they are the value of qualities such as weight or color. Taxonomic structure of GUM ontology is illustrated in Fig. 5.7

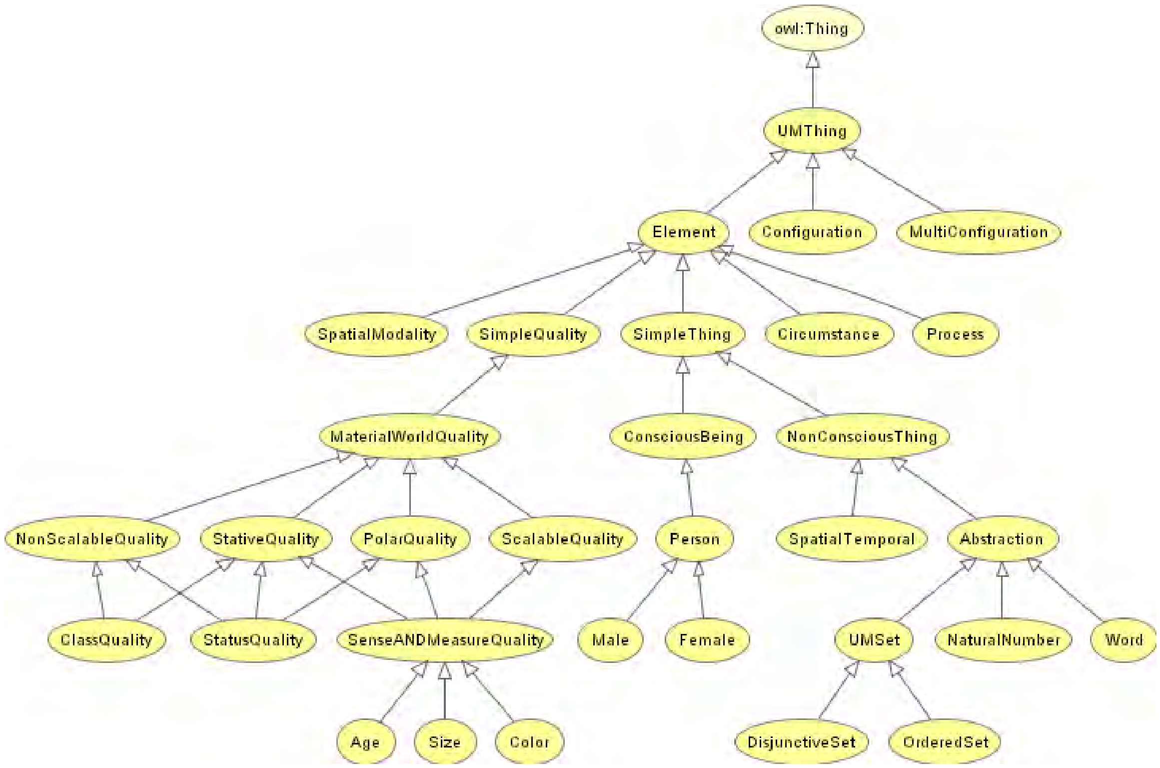

Fig. 5.7: The top-level categories of the Generalized Upper Model (GUM)
http://www.ontospace.uni-bremen.de/twiki/pub/Main/LinguisticOntology/GUM-3.owl

5.3.5 Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)

DOLCE, belongs to the WonderWeb project Foundational Ontology Library (WFOL) and is designed to be minimal in that it includes only the most reusable and widely applicable upper-level categories, rigorous in terms of axiomatization and extensively researched and documented [Masolo et al, 2003].

Currently, the DOLCE ontology includes various modules. The module hierarchy of DOLCE (for version 397) is illustrated in Fig. 5.8.

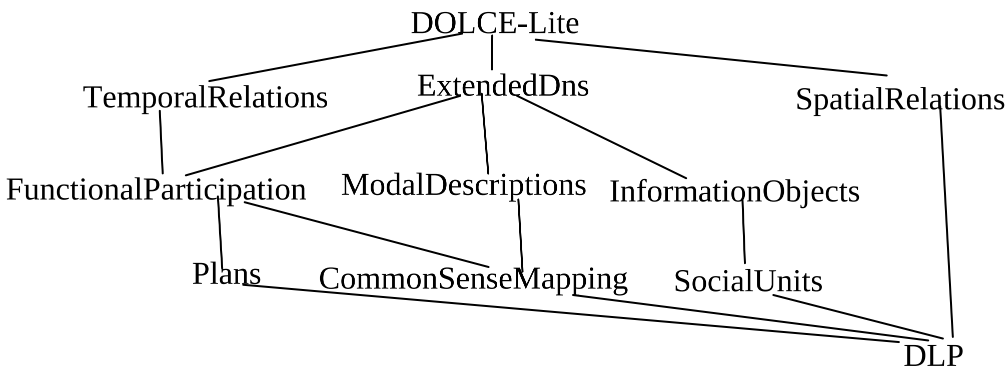

Fig. 5.8: Module hierarchy of DOLCE (for version 397)
http://dolce.semanticweb.org/

The backbone ontologies of DOLCE consist of DOLCE_Lite, TemporalRelations, SpatialRelations and ExtendedDnS. DOLCE-Lite is an encoding of most formalized predicates in DOLCE-Full.

TemporalRelations and SpatialRelations are respectively two sets of temporal relations defined over perdurants, and of spatial relations that simplify the expression of places and locations from particulars to regions. The DnS (Descriptions and Situations) ontology provides a vocabulary to talk of reified (social) entities such as relations, roles, contexts, situations, parameters, etc. TemporalRelations, SpatialRelations, and ExtendedDnS all inherit the DOLCE-Lite ontology.

DOLCE contains specifications of domain independent concepts and relations based on formal principles derived from linguistics, philosophy, and mathematics. The upper part or basic category of DOLCE's taxonomy is sketched in Fig 5.9.

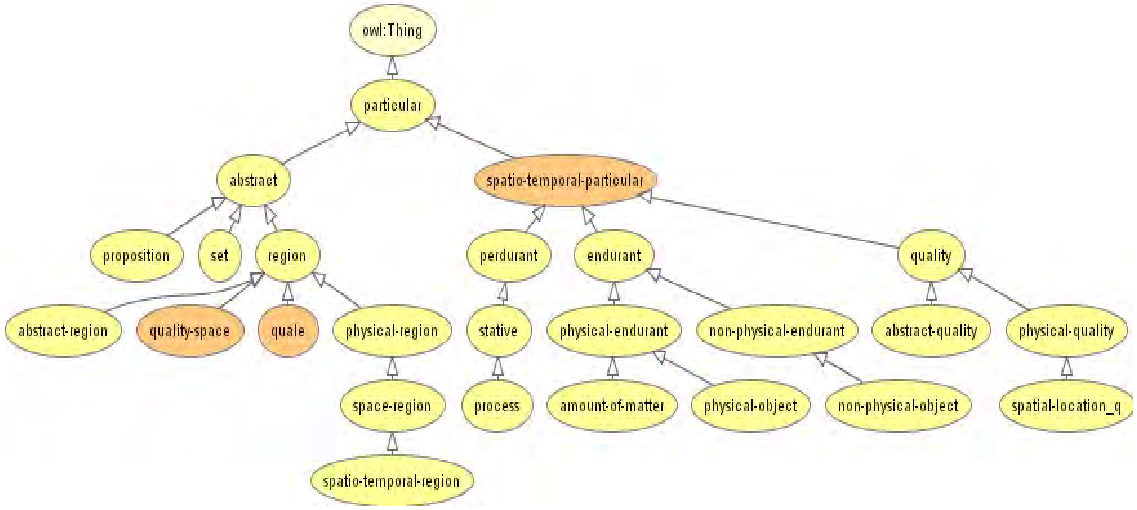

Fig 5.9: Taxonomy of DOLCE's basic categories.
http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl

DOLCE is based on a fundamental distinction between endurant and perdurant entities. The main relation between Endurants and Perdurants is that of participation: an endurant "lives" in time by participating in a perdurant. For example, a person, who is an endurant, may participate in a discussion, which is a perdurant. A person's life is also a perdurant, in which a person participates throughout all its duration.

Qualities as basic category in DOLCE taxonomy can be perceived or measured such as shapes, colors, sizes, sounds, smells, as well as weights, lengths. Notion of physical qualities is a sub concept of qualities and are those that directly inhere to physical endurants which have a clear spatial location (see Fig. 5.9).

In the DOLCE taxonomy it is distinguished between a quality (e.g., temperature, elevation or rock type), and its "value" (e.g., $30^{\circ c}$, 4810 meters or loam). The latter is called quale, and describes the position of an individual quality within a certain region. quale is an "atomic part" of a region and appears to be used to represent a region that does not have any distinguishable sub region. Quality region is indirectly the sub concept of abstract. The category of quality regions and its relation with basic category of DOLCE is depicted in Fig. 5.9.

After investigation of the existing ontologies, the development of the upper ontology, the ontology of measurement theory and the core ontology of geo-services are discussed in the following sections. These ontologies contain concepts and relationships which are necessary for discovery of geo-services.

5.4 Proposed Layered-Base Structure of Ontologies

In order to approach the objectives of the research, three ontologies at top and domain levels including the upper ontology, the ontology of measurement theory and the core ontology of geo-services are developed. The Description and Situation (D&S) ontology [Gangemi and Mika 2003] is added in order to fill the conceptual gap between the upper ontology and the ontology of measurement theory on one side and the core ontology of geo-services on the other side. These ontologies are related to each other in a layered-base structure (Fig. 5.10).

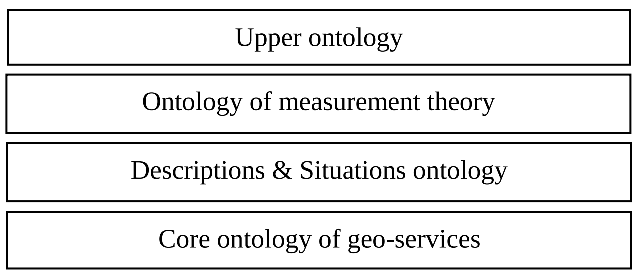

Fig. 5.10: Ontological structure

This structure is building block of the proposed methodology for discovery of geo-services. The following sub sections explain these ontologies and included concepts and relationships.

5.4.1 The Ontology of Measurement Theory

Every entity comes with certain qualities, which exist as long as the entity exists [Masolo et al. 2003]. In field conceptualizations, these qualities are a set of states for modeling the natural system which can be observed in each location. Field-based geospatial data can be used to record and represent qualities like temperature, population density or soil type which play the role of input or output for field-based geo-services. The characteristics of field including, type of measurement and unit of measurement are an important part of describing the semantic of input and output of a field-based geo-service.

5.4.1.1 Type of Measurement

The result of observation is recorded as magnitudes on a measurement scale. The attribute of field data is commonly classified into four scales of measurement namely ratio, interval, ordinal, and nominal [Stevens 1946]. For example, attributes such as runoff rate, flow rate, wind speed, infiltration rate and physical distance are expressed on a ratio scale. Attributes such as temperature, latitude, longitude, compass directions and times of day are expressed on an interval scales. These measurement scales differ in what arithmetic operators can be performed. For example, it is possible to divide, subtract, sum two values with ratio scales while it is just possible to sum or subtract two values with interval scales such as temperature in degree Fahrenheit. Attributes measured in ratio or interval scales are categorized as quantitative attributes (ratio quantity and interval quantity (Fig. 5.11)).

Attributes such as drainage class or erosion potential are usually on an ordinal scale often coded by numbers (e.g. 1 = good, 2 = medium, 3 = poor). Other attributes such as land cover, soil type, soil texture and rock type are on a nominal scale (e.g. 1 = rocky, 2 = loam). The ordinal and nominal values cannot be used in mathematical expressions, and are therefore classified as qualitative (ordinal quality and nominal quality (Fig. 5.11)).

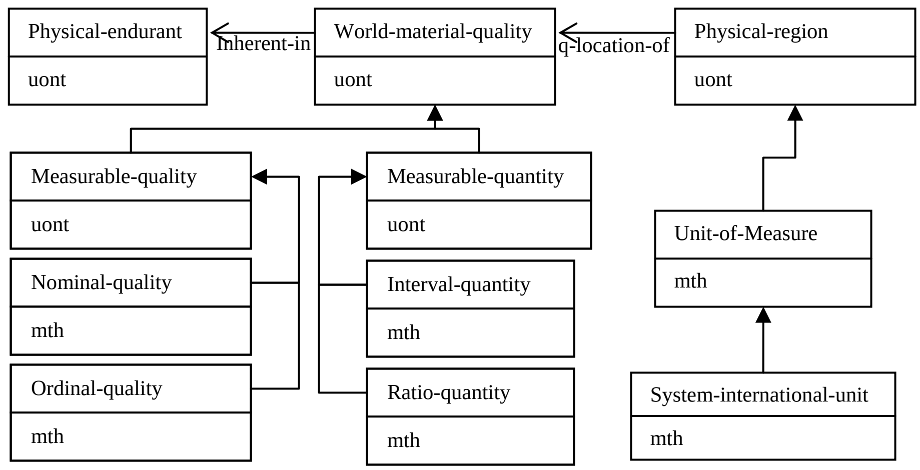

Fig. 5.11: The diagram shows the ontology of measurement theory (boxes with "mth" tag) and its alignment to concepts of upper ontology (boxes with "uont" tag). Filled arrows show the subsumption relationships (is-a or super-class/subclass relation).

5.4.1.2 Unit of Measurement

The unit of measurement is another characteristic used for describing the semantic of field's qualities. Magnitudes of quantitative attributes such as runoff rate may be compared with units of measurement such as l/m^2, kg/m^2, $pound/feet^2$. Therefore, measurement unit must be described in the ontology of measurement theory. The SI (system international) units are a subset of measurement unit. These concepts are respectively called *system-international-unit* and *unit-of-measure* as illustrated in Fig. 5.11.

For building sample ontologies and matching between them l/m^2, kg/m^2, m and mm are used as individuals of unit-of-measure concept (KilogramPerSquareMeter, LiterPerSquareMeter, Meter and Millimeter).

5.4.2 Core Ontology of Geo-services

An ontology containing geo-service's concepts is required to describe the properties and capabilities of geo-services. The Web-Ontology Working Group at the World Wide Web Consortium has produced an ontology of service concepts that supplies a web service designer with a core set of markup language constructs for describing the properties and capabilities of a Web service [OWL-S 2004]. But OWL-S seems to lack a formal semantic framework behind. Some of the missing semantics is in the text of the document [Mika et al. 2001]. A specified limitation is that for each Service, only one ServiceModel is expected to hold. This makes evaluating the relationship between a ServiceModel required by a requester and the one underlying the provider's system impossible [Mika et al. 2001].

In addition, OWL-S only allows defining parameter types for input or output parameters of geo-service by selecting a predefined type or a defined class. There is no possibility to describe other details of the input and output such as type of measurement or unit of measure. The Fig. 5.12 shows that, for example, a Boolean type was defined for atomic services using OWL-S.

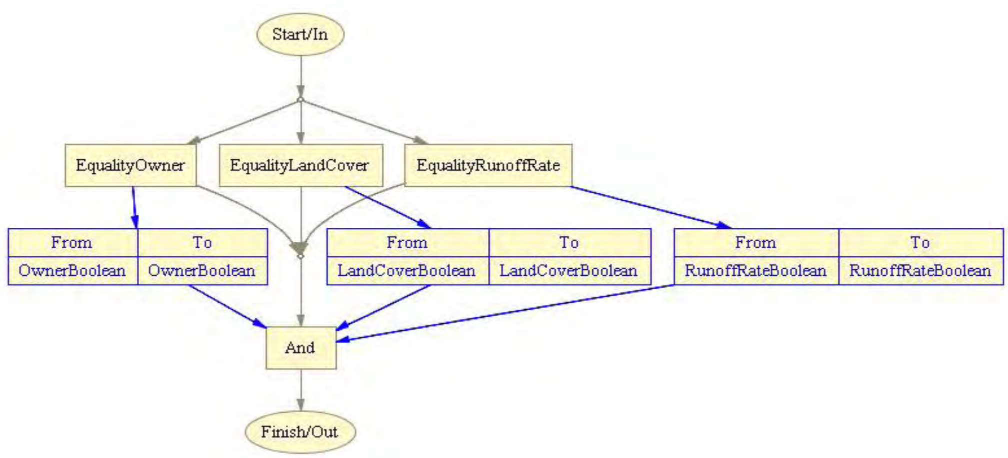

Fig 5.12: Process graph and dataflow for composite service

To overcome the limitations of OWL-S, the core ontology of geo-services must include concepts such as geo-service, geo-operation and service profile (Fig. 5.13). The evaluation of requested and provided geo-services can be performed by determining the degree of matching between these concepts.

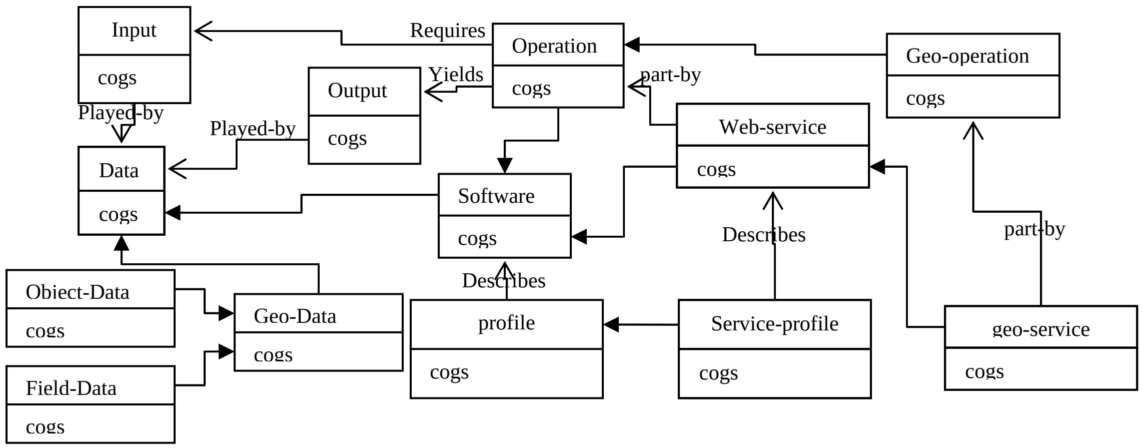

Fig. 5.13: The diagram shows the concepts and relationships for describing geo-services. Filled arrows show the subsumption relationships (is-a or super-class/subclass relation)

Fig. 5.14 illustrates the taxonomy of the core ontology of geo-services and alignment of this ontology to upper ontology.

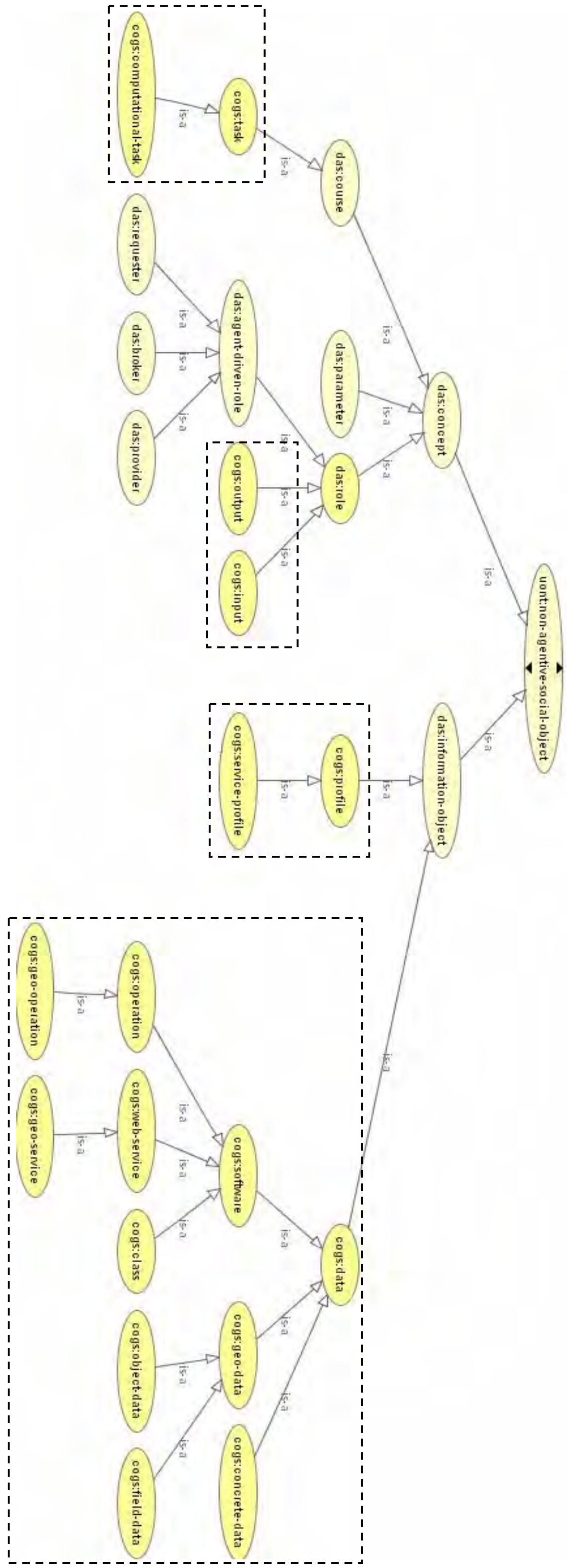

Fig. 5.14: The dashed boxes show the core ontology of geo services aligned with upper ontology.

5.4.3 The Upper Ontology

The concepts in the ontology of measurement theory and the core ontology of geo-services must be aligned with general concepts in an upper ontology (Fig. 5.11 and Fig. 5.15). Alignment to an upper ontology means relating the concepts and relations of an ontology to the basic categories of human cognition investigated by philosophy, linguistics or psychology [Mika et al. 2001].

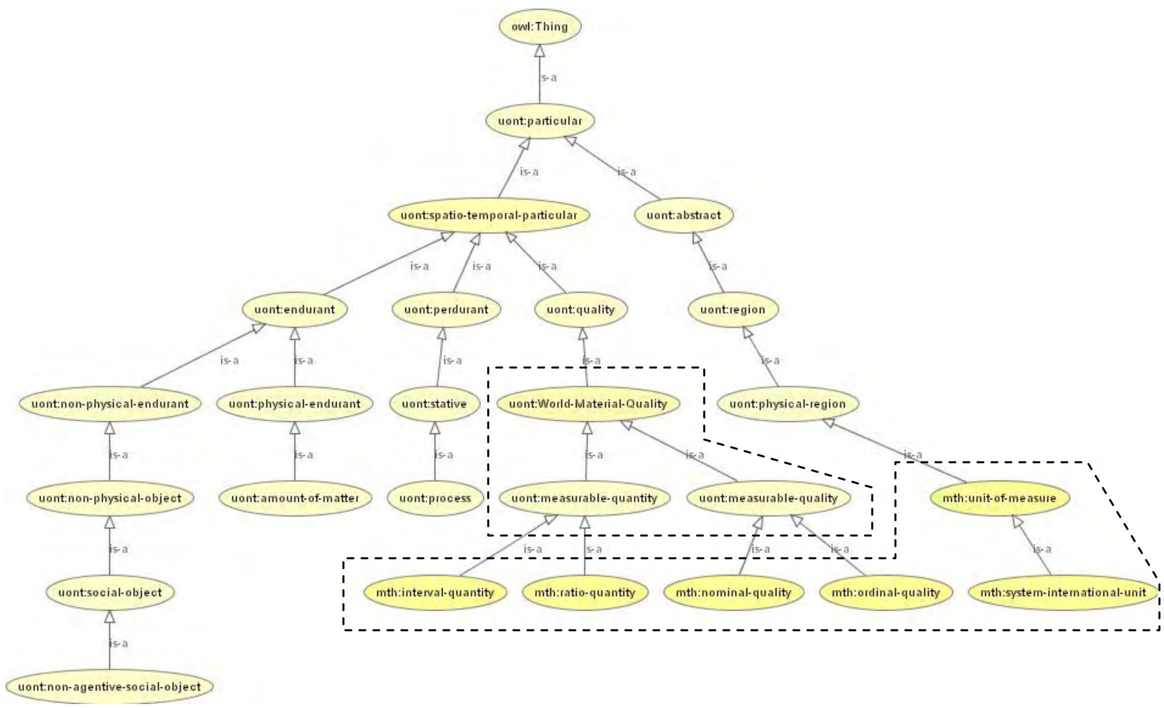

Fig 5.15: Taxonomy of the proposed upper ontology. Dashed boxes show the new added concepts with "uont" tag to DOLCE and concepts of the measurement theory ontology with "mth" tag

The DOLCE that belongs to the WonderWeb project Foundational Ontology Library (WFOL) [Masolo et al. 2003] has been selected as framework. In DOLCE, attributes of entities are called qualities [Masolo et al. 2003] and it is not distinguished between quantitative and qualitative aspects of attributes. To avoid a name conflict between quantitative and qualitative aspects of geo spatial attributes and the quality concept in DOLCE a specialized concept called world-material-quality is added as subclass of quality concept in the DOLCE taxonomy. Soil type, population density, precipitation-rain-fall and velocity of wind are a number of individuals of qualities which inheres in the entities such as soil, city, weather or wind. These individuals are also member of the world-material-quality. The world-material-quality is categorized into measurable quantity and measurable quality according to its quantitative and qualitative aspects (Fig. 5.15).

5.4.4 Descriptions and Situations (D&S) ontology

The intended meaning of non-physical objects e.g. service descriptions emerges only in the combination of other entities. A standard, a plan, a view or a social role is usually represented as a set of statements which inter-relate these notions [Navratil 2002].

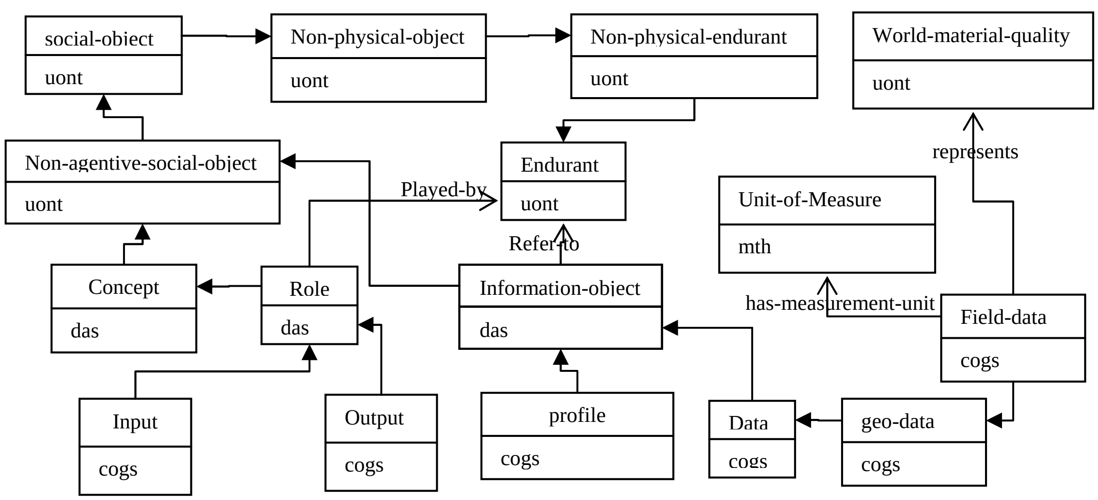

Fig. 5.16: Alignment of core ontology of geo-services (with "cogs" tag) to upper ontology (with "uont" tag) through the D&S ontology (with "das" tag) and relation of the ontology of measurement theory (with "uont" tag) with the core ontology of geo-services.

The concepts in the core ontology of geo-services are tied to the concepts of the upper ontology through the descriptions and situations (D&S) ontology which fills the gaps between the core ontology of geo-services and the upper ontology. For example, operation, web-service and service-profile in the core ontology of geo-services are sub-concepts of information-object concept which is in the D&S ontology. This concept is a sub-concept of non-agentive-social-object, a general concept in the upper ontology. The diagram illustrated in Fig. 5.16 shows the alignment of the core ontology of geo-services with upper ontology through the

D&S ontology. It also shows the unit-of-measure concept in the ontology of measurement theory which has a relation with field-data concept in the core ontology of geo-services.

5.5 Chapter Summary

An ontological structure with layered-based architecture resolve semantic ambiguities of geo-services and facilitate the automation of geo-services' discovery. This chapter provided a definition for ontology and proposed a layered-base structure of ontologies which contains the upper ontology, the D&S ontology, the core ontology of services and the ontology of measurement theory. By committing to these ontologies which have been developed semantic ambiguities in discovering of geo-services and difficulty on reasoning over ontologies of requested and provided geo-services can be reduced.

The next chapter pays attention to Description Logics (DL) formalism for representing the concepts and their relationships in an ontology. It is specially focused on OWL (Web Ontology Language) which has been developed by Web Ontology Working Group as part of the W3C Semantic Web Activity [OWL, 2004]. To discover an appropriate service for a requester require a matchmaking procedure. The next chapter also investigates the matchmaking approach and degrees of matching to compute the similarity between requested and provided geo services.

Chapter 6 **Description Logics for Building Ontologies**

6.1 Introduction

An ontology typically consists of a hierarchical description of concepts in a domain of discourse. It can semantically provide geo-service descriptions and facilitate machine-based communication between requester (modeler) and provider (GIS). In this case the geo-services can be more flexibly interpreted by intelligent agents. Required ontologies in a form of layered-base structure were proposed to maintain semantic framework for discovery of field-based geo-services. These ontologies are the upper ontology, the ontology of measurement theory and core ontology of geo-services for providing semantic framework and D&S ontology for filling conceptual gaps between these ontologies. They contain various axioms for describing concepts and relationships required for describing geo-services that were discussed in the pervious chapter.

The degree of formality employed in capturing the descriptions of concepts and relationships can be quite variable, ranging from natural language to logical formalisms, but increased formality and regularity clearly facilitates machine understanding [Guarino, 1997]. Description Logics (DLs) provide the formal foundation for modern ontology languages. More recently, Description logics have come into focus in the knowledge engineering and ontology literature due to the rise in popularity of object-oriented design and proposals for the intended functionality of the Semantic Web [Berners-Lee et al. 2001].

This chapter explains logic. Further it introduces DLs, and discusses syntax, semantic and terminology of their concepts as well as their constructors as key characteristic of DLs' expressive power. The OWL as a DL based ontology language and central standard for ontologies on the Semantic Web is also discussed. Then concepts in the proposed ontologies required for describing the provided and requested geo-services are explained. In order to discover the appropriate geo-service it is needed to perform matching between service profiles of provided and requested geo-services. Therefore it discusses the matchmaking process and degree of matching.

6.2 Logics

A logic is a formal language that allows the axiomatization of the domain information, and the drawing of conclusions from that information. Logics are characterized by what they commit to as primitives. In knowledge base a logic commits to what an agent believes about facts, but in an ontology as a special kind of knowledge-base a logic commits to what exist in the world.

The basic elements of the logic representation are characterized as unary predicates, denoting sets of individuals or atomic concepts, and binary predicates, denoting relationships between individuals or atomic roles. Two disjoint alphabets of symbols are used to denote atomic concepts and atomic roles. Terms are then built from the basic symbols using several kinds of constructors. A logic should be expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

Many logics (including standard First Order Logic (FOL)) use a model theory based on Zermelo-Frankel set theory [Drummond, 2005]. In a logical based approach, the representation language is usually a variant of FOL, and reasoning amounts to verifying logical consequence. FOL can be described according to first-order predicate calculus.

6.3 Description Logics (DLs)

To describe the details of field-based geo-services needs an ontology language which introduces concepts (also known as classes, entities), properties of concepts (also known as slots, attributes, roles), relationships between concepts (also known as associations), and constraints.

DLs are subsets of First Order Logic (FOL) [Borgida 1996]. DLs are a well-known family of knowledge representation formalisms. They are based on the notion of concepts (unary predicates, classes or types) and roles (binary predicates, relations or properties), and are mainly characterized by constructors that allow complex concepts and roles to be built from atomic ones [Baader et al. 2002]. Constructors determine the expressive power of DLs.

The use of a DL based language for formal representation of ontology is central to the approach of this research. Therefore, some details of formalism will be helpful in understanding the remainder of the thesis. Further for the purpose of clarity and compactness, the DL notions are used for representing statements describing concepts, properties, constraints and axioms for the rest of this thesis. In the following sections, expressive power and terminology of DLs are discussed.

6.3.1 Syntax of DLs

DLs are distinguished by the constructors they provide. The language $\mathcal{AL}$ (acronym for attributive language) is a minimal DL that is of practical interest. Elementary descriptions are atomic concepts and atomic roles. Complex descriptions can be built from them inductively with concept constructors. Table 6.1 summarizes the constructors and syntax rules supported by $\mathcal{AL}$ description language, where C (possibly subscripted) is an atomic concept, R is an atomic role [Baader et al., 2002].

DL Syntax	Constructor
C_1	Atomic Concept
$\top$	Universal Concept
$\perp$	Bottom Concept
$\neg C_1$	Atomic Negation
$C_1 \cap C_2$	Intersection
$\forall R.C$	All value Restriction
$\exists R.\top$	Limited Existential Quantification

Table 6.1: DL syntax of $\mathcal{AL}$ language's constructors

For example, $Person$ and $Female$ are atomic concepts. Then a $\mathcal{AL}$ concept describing, those persons that are female can be illustrated as follow:

$$female \subseteq person$$

The $\subseteq$ symbol indicates that female is subclass of person. The more details about this symbol are represented in section 6.3.4.

If it is supposed that *hasChild* is an atomic role, then the concept denoting those persons whose children are female can be represented as follow:

$$Person \cap \forall hasChild.Female$$

Using bottom ($\perp$ means nothing); also those persons without a child can be described as follow:

$$Person \cap \forall hasChild.\bot$$

For expressing these concepts an "All value restriction" is used. It states that x is an instance of $\forall R.C$ if all objects related to x via R are instances of C. It also states that all objects with no relation to x are instances of $\forall R.C$ as well [Drummond, 2005].

Those persons that have at least a child can be represented as:

$$Person \cap \exists hasChild.\top$$

An existential quantification is used to express this concept. Existential quantification states that for an object x to be instance of $\exists R.C$, there has to exist an object, say y, which belongs to C and is related via R to x. In this case this member can have relations with instances which are not member of C [Drummond, 2005]. These restrictions make constrains on the R and C concept is the filler of the constrain.

6.3.2 More Expressive Description Logic

In the $\mathcal{AL}$ only top concept or $\top$ can be used as filler for existential quantification. Thus complex description can not be expressed by filler rather than top concept. However concepts with complex description like following concept are frequently used for expressing geo-operation:

$$operation \cap \exists yeilds.output$$

The expressive power of the $\mathcal{AL}$ language is restricted and not sufficient to express geo-service concepts with complex description. In order to create complex expression, the DL language used in this research must contain cardinality restriction, $hasValue$ restriction and union ($C \cup D$) in addition to $\mathcal{AL}$ constructors as well as full existential quantification ($\exists R.C$). More expressive languages are obtained if further constructors are added to $\mathcal{AL}$. Extending $\mathcal{AL}$ by any subset of constructors yields a particular $\mathcal{AL}$-language. The logics resulting from the kinds of extensions are traditionally described using a naming scheme in which each extension adds its own distinctive label to the base name $\mathcal{AL}$. $\mathcal{AL}$-languages can be named by a string of the following form:

$$\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$$

where a letter in the name denotes for the presence of the corresponding constructor [Baader et al., 2002]. These constructors and their syntax are depicted in table 6.2:

Constructor	**Letter**	**DL Syntax**
Union of concepts	$\mathcal{U}$	$C \cup D$
Full existential quantification	$\mathcal{E}$	$\exists R.C$
At-least restriction	$\mathcal{N}$	$\geq nR$
At-most restriction	$\mathcal{N}$	$\leq nR$
Negation of concept	$\mathcal{C}$	$\neg C$

Table 6.2: DL syntax of additional constructors

By adding full existential quantification ($\exists R.C$) (with $\mathcal{E}$ label) and union ($C \cup D$) (with $\mathcal{U}$ label) to $\mathcal{AL}$ a language is developed which according to naming scheme is called $\mathcal{ALUE}$. (It is equivalent to $\mathcal{ALC}$ because union and full existential quantification are equivalent to negation (vice versa) [Baader et al. 2002]). By using this language the statements like "those geo-data that have at least a unit-of-measure and represents at least a world-material-quality" can be expressed as follow:

$$geo-data \cap (\exists has-m-unit \cdot unit-of-measure \cap \exists represents \cdot world-material-quality)$$

This axiom describes that a geo-data at least represents a real world quality which has a unit of measure. The language which is used in this research for expressing axioms and constrains about concepts is an extended $\mathcal{ALC}$ language. It has other constructors in addition to $\mathcal{ALC}$ constructors which are explained in next sections.

6.3.3 Semantics of DLs

In the pervious sections it is paid attention to syntactic of DL. However, ontology must contain semantics of concepts and their relationships. Therefore, in this section the semantics of the DL elements are discussed in order to be used for interpreting statements of the ontology.

The set Δ represents the domain of discourse which is the part of the world being modeled. Objects, instances or individuals of domain of discourse are elements of Δ. Classes, types or concepts are subsets of Δ. Relation, properties or roles are subsets of $\Delta \times \Delta$. The meaning of a language is defined by a set-theoretic semantic [OWL 1.1]. Semantics is based on interpretations where an interpretation $I = \langle \Delta^I, \cdot^I \rangle$ consists of a nonempty set Δ^I denoting the domain and a function $\cdot^I$ denoting the interpretation function. The domain set is divided to two disjoint sets, object domain Δ_o^I and datatype domain Δ_d^I. Interpretation function maps classes or concepts to subsets of the object domain, individuals into elements of object domain, datatypes into subsets of datatype domain and data values into elements of datatype domain.

In addition, two disjoint sets of properties are distinguished: object properties and datatype properties. The interpretation function maps the former into subsets of $\Delta_o^I \times \Delta_o^I$ and the latter into subsets of $\Delta_d^I \times \Delta_d^I$. An interpretation is called a model of an ontology O if it satisfies each of the axioms in O. An ontology O is said to be satisfiable if it has a model, and a class C is said to be satisfiable with regard to O if there is a model of O in which the interpretation of C is non-empty [Bechhofer and Horrocks, 2003].

Table 6.3 shows semantics of simple and complex elements which are the result of operating constructors on simple elements. The semantics of these elements are used for interpreting concepts in the ontologies proposed in the current research due to the fact that these elements exist the selected language.

DL Syntax	Semantics
C	$C^I \subset \Delta^I$
$\top$	$\top^I = \Delta^I$
$\perp$	$\perp^I = 0$
$C_1 \cap ... \cap C_n$	$(C_1 \cap ... \cap C_n)^I = C_1^I \cap ... \cap C_n^I$
$\forall R \cdot C$	$(\forall R \cdot C)^I = \{a \in \Delta^I \mid \forall \mathrm{b} \cdot (\mathrm{a}, \mathrm{b}) \in R^I \rightarrow b \in C^I\}$
$\exists R \cdot \top$	$(\exists R \cdot \top)^I = \{a \in \Delta^I \mid \exists \mathrm{b} \cdot (\mathrm{a}, \mathrm{b}) \in R^I\}$
$C_1 \cup ... \cup Cn$	$(C_1 \cup ... \cup C_n)^I = C_1^I \cup ... \cup C_n^I$
$\exists R \cdot C$	$(\exists R \cdot C)^I = \{a \in \Delta^I \mid \exists \mathrm{b} \cdot (\mathrm{a}, \mathrm{b}) \in R^I \wedge b \in C^I\}$
$\geq nR$	$(\geq nR)^I = \{a \in \Delta^I \mid \mid \{\mathrm{b} \mid (\mathrm{a}, \mathrm{b}) \in R^I\} \mid \geq n\}$
$\leq nR$	$(\leq nR)^I = \{a \in \Delta^I \mid \mid \{\mathrm{b} \mid (\mathrm{a}, \mathrm{b}) \in R^I\} \mid \leq n\}$
$\neg C$	$(\neg C)^I = \Delta^I \setminus \mathrm{C}^\mathrm{I}$

Table 6.3: DL syntax and semantics of constructors

6.3.4 Terminology of DLs

Traditionally, a DL-based system is composed of two distinct parts: the TBox (Terminology Box) and the ABox (Assertion Box) [Baader et al. 2002].

The TBox describes the relation between concept and role expressions. It is a collection of definitions for role and concept, or a set of axioms that restrict the models for the ontology. Because of the nature of the subsumption relationships among the concepts that constitute the terminology, TBoxes have a lattice-like structure [Baader et al. 2002]. The TBox is composed of a set of statements of the forms:

$$C \equiv D \quad (R \equiv S) \quad (1)$$
$$C \subseteq D \quad (R \subseteq S) \quad (2)$$

where C, D are concepts (and R, S are roles). The statement (1) is a concept definition and asserts that the concept expressions C and D are equivalent. It introduces a new concept in terms of other previously defined concepts. For example, a spatio-temopral-particular is defined as a perdurant, endurant or quality by the following equivalence:

$$spatio-temporal-particular \equiv perdurant \cup endurant \cup quality$$

The statement (2) is a (general) concepts inclusion axioms (GCIs) and asserts that concept expression C is more specific than (or included in) expression D. It constructs a taxonomic lattice. For example field-data is a geo-data can be declared as follow:

$$field-data \subseteq geo-data$$

The ABox contains assertional knowledge that is specific to the individuals of the domain of discourse usually called membership assertions. For example,

$$unit-of-measure(Ki\,log\,ramPerSquareMeter)$$

is a concept assertion and states that the individual KilogramPerSquareMeter is a unit of measurement. Similarly,

$$has-measurement-uni(DEM, Meter)$$

is a role assertion and specifies that DEM has Meter as a unit [Baader et al. 2002]. In this research, therefore, TBox contains axioms and constrains which describe the concepts and relations in the proposed ontologies since there no relation between them and domain members (i.e. sample geo-services or sample requested geo-services). Axioms and constrains which describe a sample geo-service or a sample requested geo-services must be in the ABox. However, according to discussion in section 6.5 these axioms and constrains will be also in TBox.

6.4 Web Ontology Language (OWL)

Due the fact that the OWL language is expressive enough [Li and Horrocks 2003] it is used to formalize developed ontologies in this research. In this regard, the following sections briefly explain three OWL sublanguages, compare them and discuss the reasons that OWL-DL is the appropriate sublanguage to formalize required ontologies developed by this research.

6.4.1 OWL Sub-Languages

OWL is a standard for ontologies on the Semantic Web from the World Wide Web Consortium (W3C). It is built on top of RDF (Resource Description Frame) (OWL semantically extends RDF(S) (Resource Description Frame Scheme)), with its predecessor language DAML+OIL (DARPA Agent Markup Language + Ontology Interface Layer) [OWL1.1]. The OWL is classified into three sublanguages which are as follow:

OWL-Lite supports those users primarily needing a classification hierarchy and simple constraint features. While OWL-DL supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL-DL is so named due to its correspondence with Description Logics. OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. It is unlikely that any reasoning software will be able to support every feature of OWL-Full.

The choice between OWL-Lite and OWL-DL may be based upon whether the simple constructs of OWL-Lite are sufficient or not. The choice between OWL-DL and OWL-Full may be based upon whether it is important to be able to carry out automated reasoning on the ontology or whether it is important to be able to use highly expressive and powerful modeling facilities such as meta-classes (classes of classes) [W3].

There is a tradeoff between the computational complexities of reasoning and the expressiveness of the language, which itself is defined in terms of the constructors that are admitted in the language [Brachman and Levesque, 1984].

Since OWL-DL is expressive enough to satisfy the needs of this research and existing inference systems are able to handle computational complexity of this sublanguage during finite time, OWL_DL which is a DL based language is selected as appropriate language to formalize the developed ontologies. The next section pays attention to syntax and semantics of OWL_DL in more details to understand its primary elements and constructors.

6.4.2 Syntax and Semantics of OWL-DL

OWL-DL is an extended logical language based on $\mathcal{ALC}$ (Fig. 6.1). More precisely OWL-DL is equivalent to $\mathcal{SHOIN(D)}$ [Farrar and Bateman 2005].

Key:
$\mathcal{I}$: inverses;
$\mathcal{N}$: number restrictions;
$\mathcal{Q}$: qualified restrictions;
$\mathcal{H}$: role hierarchies;
$\mathcal{R}+$: transitivity over roles;
$\mathcal{D}$: domains of specified data types;
$\mathcal{O}$: enumeration;

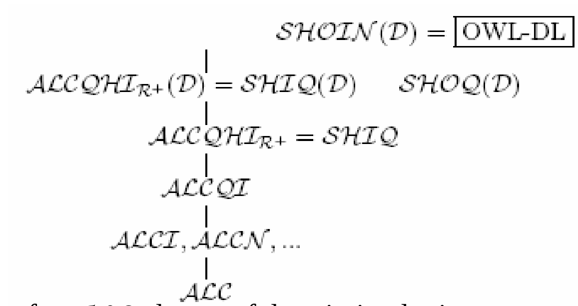

Fig. 6.1: Expressivity hierarchy for $\mathcal{ALC}$ classes of description logics [Farrar and Bateman, 2005]

According to name scheme for DLs languages $\mathcal{S}$ denotes that $\mathcal{ALC}$ is extended with transitive roles [Horrocks et al. 1999], $\mathcal{H}$ indicates role hierarchies (equivalently, inclusion axioms between roles), $\mathcal{O}$ stands for enumeration (classes whose extension are their individuals) [Blackburn and Seligman 1995], $\mathcal{I}$ indicates inverses, $\mathcal{N}$ represents unqualified number restrictions, and $(\mathcal{D})$ indicates datatypes (Fig. 6.1) [Horrocks and Sattler 2001]. A detailed discussion of OWL is, however, beyond the scope of this research.

OWL-DL has a rich set of constructors in order to express complex concepts related to geo-services. For example, with number or cardinality restrictions it is able to formalize statements like "a geo-operation is an operation that requires at least one input and yields exactly one output" as:

$$operation \cap (\, requires \cdot input \geq 1 \cap yields \cdot output = 1\,)$$

DL Syntax	Constructor	OWL Syntax
C	Atomic Concept	Class
$C_1 \cap ... \cap C_n$	Intersection or Conjunction	intersectionOf
$C_1 \cup ... \cup C_n$	Union or Disjunction	unionOf
$\neg C$	Atomic Negation	complementOf
$\exists R.C$	Quantifier Restrictions	someValuesFrom
$\forall R.C$		allValuesFrom
$\{a_1,..., a_n\}$	Enumeration	oneOf
$\geq nR.C$		minCardinalityQ
$\leq nR.C$	Number Restrictions	maxCardinalityQ
$= nR.C$		cardinalityQ
$\ni R.\{a\}$	Value Restriction	hasValue

Table 6.4: DL syntax and OWL syntax of OWL's constructors

DL Syntax	Semantic	OWL Syntax
$C_1 \subseteq C_2$	$C_1^I \subseteq C_2^I$	subClassOf
$C_1 \equiv C_2$	$C_1^I = C_2^I$	sameClassAs
$R_1 \subseteq R_2$	$R_1^I \subseteq R_2^I$	subPropertyOf
$R_1 \equiv R_2$	$R_1^I = R_2^I$	samePropertyAs
$C_1 \subseteq \neg C_2$	$C_1^I \subseteq \neg C_2^I$	disjointWith
$\{x_1\} \equiv \{x_2\}$	$x_1^I = x_2^I$	sameIndividualAs
$\{x_1\} \subseteq \neg\{x_2\}$	$x_1^I \neq x_2^I$	differentIndividualFrom
$R_1 \equiv R_2^-$	$R_1^I = \{(x,y) \mid (y,x) \in R_2^I\}$	inverseOf
$\mathcal{T} \subseteq\leq 1R$	$(x, y_1) \in R \cap (x, y_2) \rightarrow y_1 = y_2$	FunctionalProperty
$\mathcal{T} \subseteq\leq 1R^-$	$(x_1, y) \in R \cap (x_2, y) \rightarrow x_1 = x_2$	InverseFunctionalProperty

Table 6.5: DL syntax, semantic and OWL syntax of OWL-DL's axioms

Part of OWL-DL's constructors and relation between concepts and role expressions are respectively represented in table 6.4 and table 6.5 (for semantics of relations refer to table 3).

Next section pays attention to some notifications which are used for describing geo-services in the current research. These notifications must be taken into account when describing geo-services in order to increase precision of geo-services discovery.

6.5 Notifications for Describing Geo-services

Generally the profile contains functional description of a geo-service which is expressed in terms of Input, Output, Preconditions and Effects (IOPEs) [Li and Horrocks, 2003]. Preconditions are conditions that hold before the operation can be executed properly, and effects are conditions that hold after the successful execution of the service. These conditions are more applicable in web services performing business processes. For example, in the case of buying and selling services, the precondition is a minimum amount of money in the credit card and effect of executing the service is the reduction of money from the credit card. Geo-services directly have effect on a model of environment rather than environment itself. Further, preconditions also have no application in geo-services. Since a geo-service is discovered by performing evaluation between ontologies of the provided and requested geo-services. These ontologies describe input and output of geo-services with no ambiguities due to existence of a semantic framework which is supported by the proposed ontologies.

On the other hand, the requested and provided ontologies of sample geo-services are at application level. In fact, these ontologies should contain individuals in stead of concept. For examples:

$$cogs : calculate - runoff \ - service \in geo - service$$

where $\in$ denotes "member of".

They contain knowledge about individuals and therefore ABox comprise them. But, in the approach of matchmaking used in this research a range of degrees are used to measure the matching between the requested and provided of sample geo-services. If these ontologies contain individuals, therefore, it just can be determined that two individuals are the same or not. It is not possible to measure how well the functionality of provided geo-services coincides with the functionality of requested geo-service. Thus in this research the individuals in the ontologies of provided and requested geo-services have been expressed as concepts so that inferential power is increased. In the other words, by using concept instead of individual TBox reasoning can be applied which is much more effective than ABox reasoning [Tessaris, 2001].

6.6 Concepts and Roles used in Geo-services Evaluation

Actually the matching between the requested and provided ontologies of sample geo-services are occurred by evaluating the *service-profile*, *geo-service* and *geo-operation* concepts described in these ontologies. The ontologies of the layered-base structure contain constraints and axioms for describing these concepts. The *service-profile* can formally be described as follow:

$$cogs: service - profile \subseteq cogs: profile \cap$$
$$\exists cogs: describes \cdot cogs: geo - service$$

The service-profile concept describes geo-service concepts. The geo-service concept is formalized as follow:

$$cogs: geo - service \subseteq cogs: web - service \cap$$
$$(\forall cogs: part - by \cdot cogs: geo - operation \cap$$
$$\exists cogs: part - by \cdot cogs: geo - operation) \cap$$
$$\geq 1 cogs: part - by$$

In this case the intersection of universal and existential restrictions for part-by relationship has been used. This axiom describes that geo-service consists-of at least a geo-operation and only consist-of geo-operation. That is all instances which satisfy this restriction at least have a geo-operation and only have geo-operation.

The geo-operation is part of a geo-service. Therefore the following statement describes it:

$$
\begin{aligned}
cogs:geo-operation \subseteq\ & cogs:operation \cap \\
& \exists cogs:requires \cdot cogs:input \cap \\
& (\forall cogs:yields.output \cap \exists cogs:yields.output) \cap \\
& \geq 1 cogs:requires \cap \\
& = 1 cogs:yields
\end{aligned}
$$

Geo-service may have several input data sets and only have an output data set. This can be described by intersection of universal and existential restriction over yields relationship. The field-data play the role of the input and output of the geo-operation. The description of geo-operation depends on description of its input and output which are formalized as follow:

$$
\begin{aligned}
cogs:input \subseteq\ & das:role \cap \\
& \exists das:played-by \cdot cogs:field-data
\end{aligned}
$$

$$
\begin{aligned}
cogs:output \subseteq\ & das:role \cap \\
& \exists das:played-by \cdot cogs:field-data
\end{aligned}
$$

The field data are concrete geospatial data which describe a certain quality on the earth surface. Therefore the following statement shows its description:

$$
\begin{aligned}
cogs:field-data \subseteq\ & cogs:geo-data \cap \\
& \exists cogs:has-m-unit \cdot mth:unit-of-measure \cap \\
& \exists cogs:represents \cdot uont:world-material-quality
\end{aligned}
$$

When using OWL-S for describing a service, input and output data are related to a certain data type. In contrast, due to existence of semantic framework maintained by proposed ontologies of the layered-based structure the input and output of geo-services can be related to unit of measure and measurement scale in addition to a certain data-type or a concept. The following statements illustrate these relationships:

$$mth:unit-of-measure \subseteq uont:physical-region$$

$$\begin{aligned} uont:physical-region \subseteq\ & uont:region \cap \\ & (\exists uont:q-location-of \cdot uont:world-material-quality \cap \\ & \forall uont:q-location-of \cdot uont:world-material-quality) \end{aligned}$$

$$\begin{aligned} uont:world-material-quality \equiv\ & uont:quality \cap \\ & (uont:measurable-quantity \cup uont:measurable-quality) \subseteq \\ & uont:inherent-in \cdot physical-endurant \end{aligned}$$

where uont, mth, das and cogs are tags for uniquely identifying the concepts of the upper ontology, the ontology of measurement theory, the D&S ontology and the core ontology of geo-services respectively.

6.7 Matchmaking

In the current research, matchmaking is defined as process of discovering a geo-service which is similar to the requested geo-service. Matchmaking is the fundamental procedure enabling semantic interoperability. It is a reasoning process with goal of deciding whether a requested geo-service matches provided geo-services.

The following sections discuss the process of matchmaking and degree of matchmaking for measuring the amount of similarity between corresponding concepts of the *service-profile*, the *geo-service* and the *geo-operation* in the requested and provided ontologies.

6.7.1 Matchmaking Process

In the proposed methodology which is discussed in next chapter, the developed geo-services are described by expressing the functionality of the geo-services including their input and output. Further the requested geo-service can be described by formalizing the input or output of the ideal geo-service. Requester, at least, can describe the input and output of his ideal geo-service. The *service-profile*, the *geo-service* and the *geo-operation* concepts for requested and provided geo-services, in these ontologies, are described by using axioms or constrains existed in the proposed ontologies.

Suppose φ be the set of all provided ontologies of geo-services. For a given requested ontology O_r which describe the ideal geo-service of requester, the matchmaking process returns the set of all provided ontologies which their intersection with O_r are satisfiable. Formally this can be illustrated as follow:

$$match(O_r) = \{ O_p \in \varphi \mid satisfiable(O_p, O_r) \}$$

where O_p denotes for the provided ontology of geo-service. Two ontologies can be defined satisfiable where the intersection of their corresponding concepts does not necessarily denote the empty concept. According to this definition $satisfiable(O_p, O_r)$ defines as follow:

$$satisfiable(O_p, O_r) \Leftrightarrow \neg(O_p \cap O_r \subseteq \perp)$$

The non empty intersection of the requested and provided ontologies implies that the corresponding geo-services are similar to each other.

Therefore, it is important to determine amount of similarity that is to specify either a sample geo-service are completely identical with required geo-service or sample geo-service is subclass or super-class of required geo-service. In this regard, it is needed to define matching degrees for measuring amount of similarity which is discussed in the next section.

6.7.2 Degree of Matchmaking

The result of matching can be binary (match or not) or a measure for degree of match, i.e. for similarity [kuhn, 2005]. The degree of matching can be defined how well the functionality of any provided geo-services fits to that of a requested geo-services. In order to measure the matching of concepts, it is needed to introduce the degrees of match. By starting from the matching degree definition described in [Paolucci et al., 2002], the match levels according to “intersection to be satisfied” are extended as follow:

- If the *service-profile*, the *geo-service* and the *geo-operation* concepts in the provided ontologies of sample geo-services O_p are equivalent to those concepts in the requested ontology O_r , then the match is called Exact; formally: $O_p \equiv O_r$.
- If the *service-profile*, the *geo-service* and the *geo-operation* concepts in the requested ontology O_r are more specific than those concepts in the provided ontologies of sample geo-services O_p , then the match is called Plugin; formally: $O_r \subseteq O_p$.
- If the *service-profile*, the *geo-service* and the *geo-operation* concepts in the provided ontologies of sample geo-services O_p are more specific than those concepts in the requested ontology O_r , then the match is called Subsume; formally: $O_p \subseteq O_r$.
- If the intersection of the *service-profile*, the *geo-service* and the *geo-operation* concepts in the provided ontologies of sample geo-services O_p and their correspondence concepts in the requested ontology O_r are satisfied, then the match is called Intersection; formally: $\neg(O_p \cap O_r \subseteq \perp)$.
- Otherwise, the intersection of the *service-profile*, the *geo-service* and the *geo-operation* concepts in the provided ontologies of sample geo-services O_p and their corresponded concepts in the requested ontology O_r are empty and the match is called Disjoint; formally $O_p \cap O_r \subseteq \perp$.

6.8 Conclusion

The aim of current research is to develop the required ontologies and propose a methodology based on ontology for discovering field-based geo-service. Thus for formalizing the developed ontologies, various families of Description Logics (DLs) have been discussed and OWL as a DL based language for formal representation of web documents has been introduced. Its primary elements, constructors, expressive power and semantic have been explained. OWL_DL as sublanguage of OWL has been selected for formalizing concepts and relations of the proposed and sample ontologies. Since it is expressive enough to satisfy the needs of the current research, and there are reasoning systems to deal with this language.

For evaluating sample geo-services, *service-profile*, *geo-service* and *geo-operation* concepts in their ontologies are matched in order to measure similarity between geo-services. Therefore the axioms and constraints used to describe these concepts have been discussed. In the current research, the discovery of geo-services is based on a matching process and measuring matching between the corresponding concepts in the requested and provided ontologies of sample geo-services. Therefore, the matchmaking process and degrees of match have been discussed. In this regards five matching degrees have been defined in stead of binary matching which increase probability of discovering geo-services, since it is possible to discover geo-services with incomplete matching.

The next chapter discusses about the architecture of a methodology for discovering of geo-services based on ontology and its components. It also pays attention to steps followed by this research for building proposed ontologies and ontologies of sample geo-services. The software environment for building ontologies and performing inference between these ontologies are explained. Then implementing a prototype along with developing an application for performing this methodology is explained.

Chapter 7 Implementation a Prototype for the Solution

7.1 Introduction

This chapter discusses the architecture of the proposed methodology based on ontology for discovering geo-services. A prototype of the proposed methodology is tested by building sample ontologies of requested and provided geo-services and performing inference between their corresponding concepts and measuring their similarities. A layered-base structure of ontologies makes a semantic framework for the methodology.

This chapter introduces the architecture and explains the components of it. It also discuss about approach of building the ontologies and axioms and constraints required for describing concepts. Then the relations between OWL (Web Ontology Language) files are presented. Further it focuses on software environment applied to build ontologies and to perform match between concepts. After that it discusses about implementing a prototype for geo-service discovery.

The capabilities are evaluated in terms of the problem definition. This means that prototype concentrates on how to handle the problems of describing the requested and provided ontologies of sample geo-services and implementing match between their concepts.

7.2 Architecture of the Proposed Methodology

The proposed architecture for geo-service discovery consists of an ontology management, a matchmaker and a semantic framework supported by a layered-based structure of ontologies. Ontology management is a database mounted on a server and performs tasks of registering the ontology's specification of provided geo-service. Fig 7.1 shows the overall architecture of the methodology.

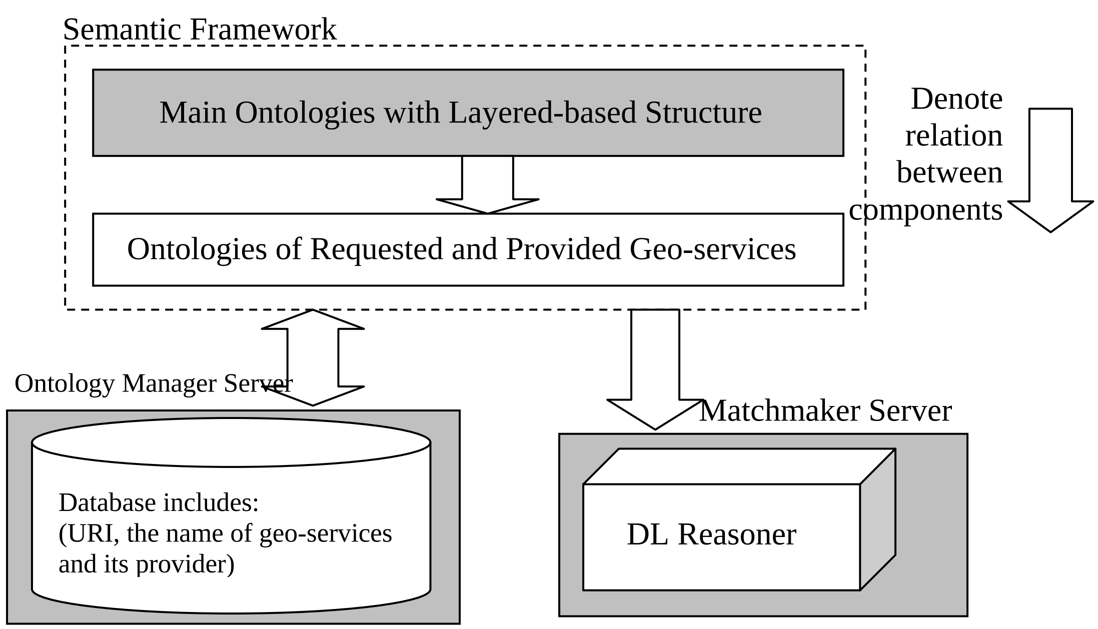

Fig. 7.1: Overall architecture of methodology for geo-service discovery

The process of registering a service consists of storing the Uniform Resource Indicator (URI) of service's ontology, the name of geo-service and geo-service provider into the database. An ontology of provided geo-service has a URI by which it is possible to access to ontology. For example the URI of the upper ontology is as follow:

http://www.ncc.org.ir/ontologies/UpperOnt.owl

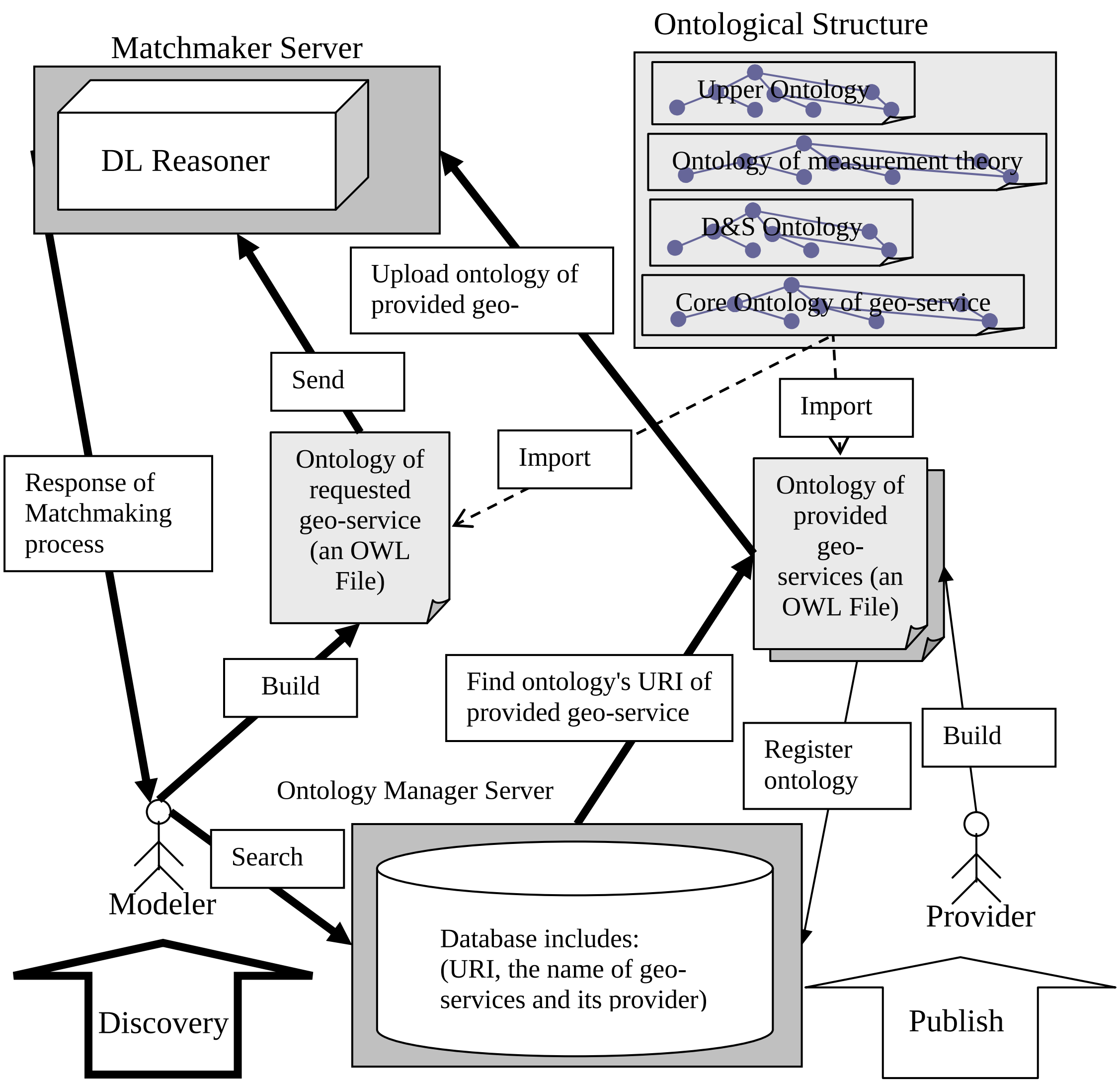

Fig. 7.2: Architecture of methodology based on ontology to discover appropriate geo-services.

Ontology management also supports searching the database based on query of the requester and uploading the appropriate ontologies of provided geo-services to the matchmaker server. DL reseaoner installed on the matchmaker server is the inference engine of the architecture which establishes reasoning between ontologies of requested and provided geo-services. After performing the process of reasoning, the matchmaker server sends the result of matchmaking to the requester.

The layered-base structure of ontologies supports a semantic framework for geo-service discovery. It is the building block of the architecture which ensures a common understanding between requester and provider by commit to these ontologies. In fact, general concepts and domain specific concepts which are used in ontologies of provided and requested geo-services have been described in one of these ontologies. Fig. 7.2 shows the architecture of the geo-service discovery based on ontology.

7.3 Building Ontologies

The upper ontology, the ontology of measurement theory, the D&S ontology and the core ontology of geo-services provide a semantic framework for geo-service discovery. Approach for building these ontologies and axioms and constraints used to describe their concepts are discussed through the following subsection.

7.3.1 Approach for Building Ontologies

Building an ontology is a pragmatic and fundamental topic for applying this architecture. Well-defined ontologies are needed to successfully practice the architecture for geo-service discovery. The following steps have been applied based on experimental practices for building ontologies.

Step1. Scope of application: When building an ontology it is needed to have an application in mind in order to scope the project. It should be considered that an ontology does not contain all the possible information about the domain. It is also no need to specialize or generalize more than the application requires.

The scope of this research has been specified as field-based geo-services discovery. With keeping this application in mind an overview and informal analysis of the domain has been performed.

Step2. Specification of a concept hierarchy: Names for important concepts in the domain is typically one of the distinct components of ontologies. A set of (non-relational) formal concepts relevant to describing domain have been identified. These concepts have been represented simply by class names or unary predicates in Description Logic. The taxonomy of ontology typically forms a super-class/sub-class or is-a hierarchy arising from inclusion relationships between classes — for example *measurable-quality* is a *world-material-quality* (or in DL form $measurable-quality \subseteq world-material-quality$). In this regards the taxonomies of upper ontology, D&S ontology, ontology of measurement theory and core ontology of geo-services have been formed. The disjoint classes i.e. classes which do not overlap with each other and also a set of sub-classes which forms a partition of a more general class have been identified.

Step3. Identification of relations or properties: it is no need to include all possible properties of a concept. Only the important properties of a concept should be included. These properties should be those properties that the application requires. Therefore the most significant properties such as $yields$, $measurment\text{-}unit\text{-}of$ which hold among concepts in the ontologies of structure have been identified.

Step4. Axioms and constraints: An ontology specifies a set of axioms or constraints which declare what should necessarily hold in any possible world or domain of discourse. They specify the meaning of one concept in terms of a logical combination of other concepts. In this regard the axioms and constraints needed to describe the meaning of concepts and properties have been specified. For example geo-operation concept is described as follow:

$$
\begin{aligned}
cogs:geo-operation \subseteq cogs:operation \cap \\
& \exists cogs:requires \cdot cogs:input \cap \\
& (\forall cogs:yields.output \cap \exists cogs:yields.output) \cap \\
& \geq 1 cogs:requires \cap \\
& = 1 cogs:yields
\end{aligned}
$$

Step5. Formalization: Finally axioms and constraints of concepts and properties have been formalized in OWL language that can be processed by reasoning systems and computers. The following illustrates the OWL form of the *geo-operation* concept.

```
<owl:Class rdf:ID="geo-operation">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="geo-service"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:TransitiveProperty rdf:ID="part-of"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class rdf:ID="output"/>
      </owl:allValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="yeilds"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="requires"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#yeilds"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="input"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#requires"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
```

```
    <rdfs:subClassOf>
      <owl:Class rdf:ID="operation"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#output"/>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#yeilds"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

Expressiveness of a language depends on the set of constructors and plays a basic role in this phase. The availability of a reasoning system to perform desired processes depends on the expressive power of language.

7.3.2 Axioms and Constraints for Describing Concepts

The core ontology of services, the ontological theory of measurement, the D&S ontology and the upper ontology are a collection of axioms and constraints that restrict the concepts and relationships about geo-services. Apart from subsumption (is-a) relationship, there may be other relationships between concepts such as the "yields" and "requires" relationships between "geo-operation", "output" and "input" concepts in order to state that every individual of geo-operation yields output and requires input.

The constraints are on relationships that the individuals participate in for a given property. For example the following are constraints on the "yields" and "requires" relationships:

$$cogs:yields = 1$$

$$cogs:requires \geq 1$$

These statements restrict the relationships and state that each individual of geo-operation concept yields an output and requires at least one input. The following statement describes the primary concept of field-data:

$$cogs: field - data \subseteq cogs: geo - data \cap$$

$$\exists cogs : has-m-unit \cdot mth : unit-of-measure \cap$$
$$\exists cogs : represents \cdot uont : world-material-quality$$

where "cogs" is a tag for uniquely identifying the core ontology of geo-services concepts. Primitive concepts are concepts that only have necessary conditions.

The following axiom states a definition for world-material-quality, and any individual that satisfies this definition will belong to the world-material-quality concept:

$$uont : world-material-quality \equiv uont : measurable-quality \cup uont : measurable-quantity$$

Concepts that have at least one set of necessary and sufficient conditions are known as defined concepts [Bergamaschi and Nebel 1994]. These conditions are used to check for class subsumption by the DL reasoner to automatically compute a classification hierarchy.

7.4 Inter-relation of OWL Files

The ontologies in the ontological structure are modular; the ontology of each layer is in a separate OWL file connected by the <owl:imports> statement. The Fig. 7.3 shows the relations between these ontologies. The line connecting two ontologies implies that the one above is imported by the one underneath.

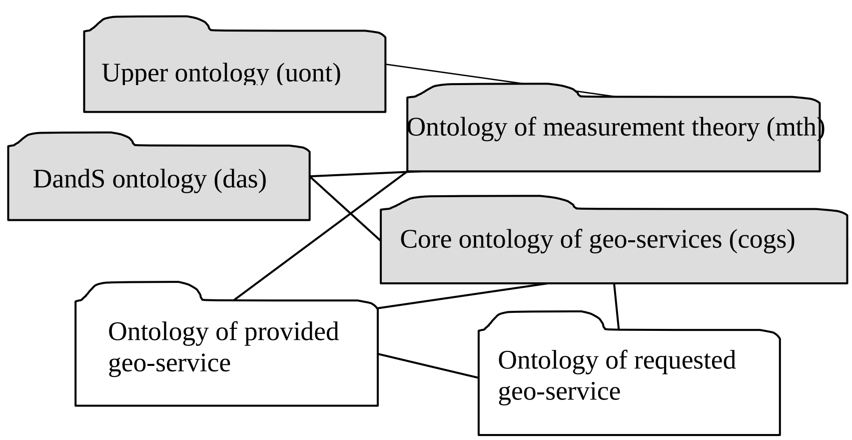

Fig. 7.3: Inter-relation between the proposed ontologies

7.5 Software Environment for Implementing the Proposed Methodology

The software environment consists of the ontology editor with capabilities of building ontology in OWL language and visualizing taxonomy of OWL ontologies. It also includes an inference engine for reasoning with ontologies. The following sub-sections discuss about ontology editor and inference engine used in this proposed architecture.

7.5.1 Protégé and OWL plug-in

Protégé [Protégé 2003] is an open source ontology editor that supports OWL-based ontology development and inference. Protégé is java-based and is extensible via plug-ins [Knublauch et al., 2004]. A lot of plug-ins have been developed and published for a wide variety of purposes, e.g., ontology export formats, visualization, reasoning, etc. Some of them have been included in the standard installation procedure of Protégé. In fact, OWL in Protégé is provided through a plug-in (Fig. 7.4); Protégé has its own internal representation mechanism for ontologies and knowledge bases, based on a meta-model, which is comparable to object-oriented and frame-based systems [Knublauch et al., 2004]. The Protégé version 3.2.1 Build 365 along with its OWL editor plug-in version 3.2.1 Build 365 have been selected and extensively used for editing ontology in this research.

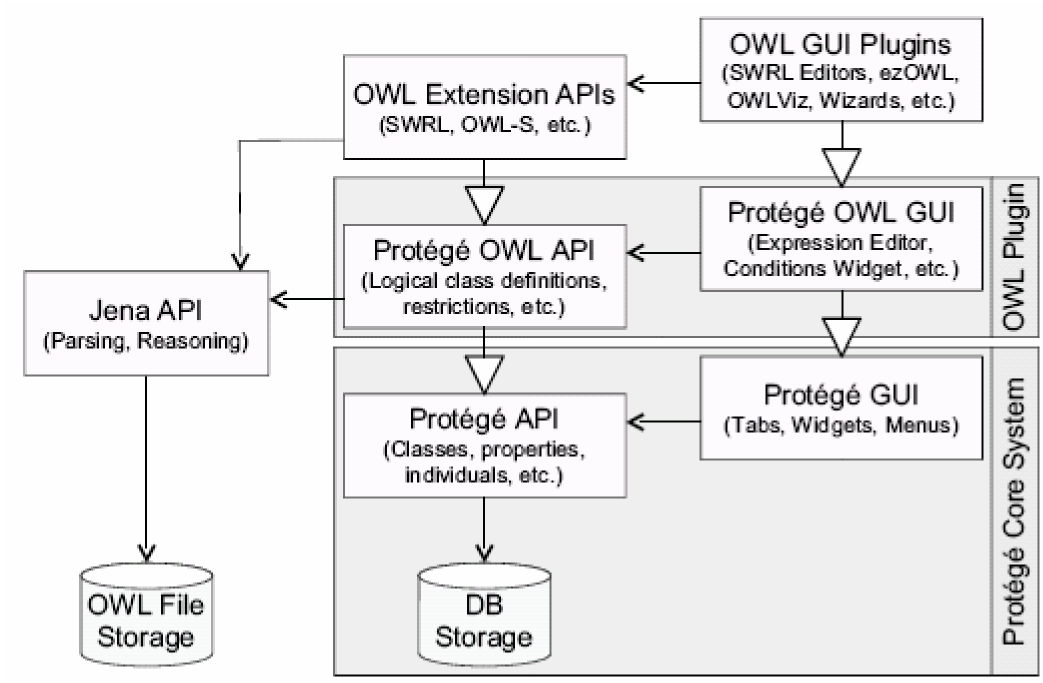

Fig.7.4: The extension mechanism of Protégé [Knublauch et al., 2004].

Protégé can be connected to a DL reasoner through the DIG interface. This interface is a standardized XML interface to Description Logics systems developed by the DL Implementation Group (DIG) [Bechhofer, 2003].

7.5.2 OWLViz plug-in as Visualization of Ontologies

OWLViz [OWLViz, 2004] is designed to be used with Protégé OWL plug-in as tool for OWL ontology inspection and documentation. It has been used extensively in this research due to its possibility to visualize taxonomy of OWL ontologies as well as classes together with individuals and properties and the flexibility of class and individual selection.

The graphical forms of taxonomy of ontologies in the layered-based structure have been produced with OWLViz in this thesis (For instance see Fig. 5.8, Fig.5.9, Fig.5.11, Fig.5.12 and Fig.5.15).

7.5.3 DL Reasoner

A DL reasoner called RacerPro is used to compute degree of matching between requested and provided geo-services (Fig 7.5). This reasoning system performs inference based on Description Logics language.

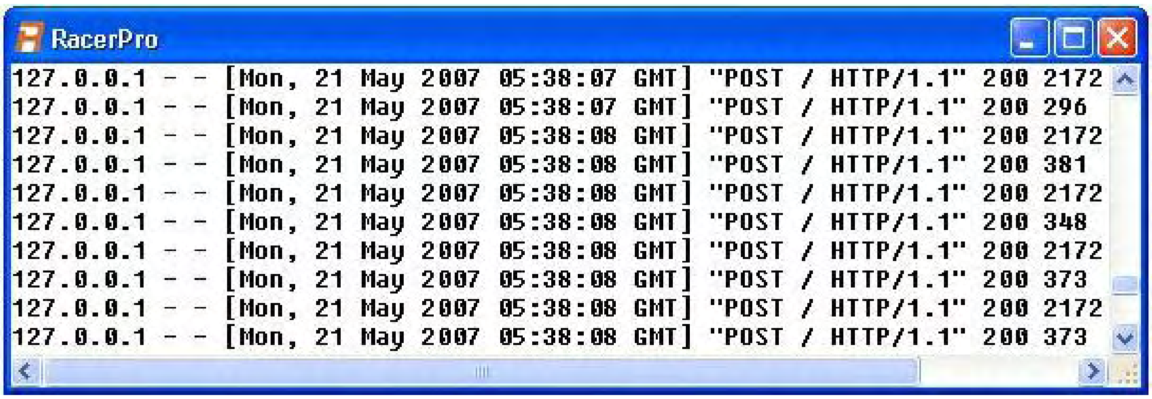

Fig. 7.5: RacerPro inference engine

RacerPro (stands for Renamed ABox and Concept Expression Reasoner Professional) is a knowledge representation system that can be used for reasoning with ontologies. RacerPro implements the description logic $\mathcal{ALCQHI}_{\mathcal{R}^+}$, also known as $\mathcal{SHIQ}$ [Racer, 2005b] (refer to Fig. 6.1). RacerPro can directly read OWL-Lite and OWL-DL documents and represent them as TBoxes and ABoxes in DL knowledge bases [Haarslev and Möller, 2003; Wessel and Möller, 2005]. The only restriction for OWL-DL is that RacerPro does not support nominals (individual names expressed in class descriptions). RacerPro provides numerous functions for managing the knowledge base and reasoning with its TBoxes and ABoxes. Knowledge base management functions, knowledge base declarations, functions for changing the reasoning mode, evaluation functions and retrieval functions are some of these functions [Racer, 2005a].

All these functions can be called through a LISP interface but RacerPro also acts as a server, providing these functions through a TCP interface and an HTTP based standard DIG interface for connecting client programs. This system is free of charge only for researches and non-comercial works. Therefore, I downloaded the RacerPro version 1.8.1 when I was doing my research in the Geoinformatic departments of technical university of Vienna for utilizing in this research.

7.5.4 Connection of Protégé to Reasoner Through the DIG Interface

Protégé can be connected to a DL reasoner through the DIG interface. This interface is a standardized XML interface to Description Logics systems developed by the DL Implementation Group (DIG) [Bechhofer, 2003]. Protégé has several commands that can send to DL reasoner through the DIG interface. It entails the following functions for checking the entire ontology:

- Check consistency: checks the entire ontology for unsatisfiable concepts.
- Classify taxonomy: checks the entire ontology for unsatisfiable concepts and implicit subsumption relationships between concept names. In the Protégé environment two additional windows are opened with respectively the inferred class hierarchy and an overview of concepts, moved by the reasoner.
- Check concept consistency: see similar function in check consistency. The results are displayed in a temporary result window.
- Compute individuals belonging to class: The results are displayed in a temporary result window.
- Get inferred super-classes: computes the classes that subsume this class. The results are displayed in a temporary result window.
- Compute inferred types: computes the inferred types (classes) for the individuals in the ontology. The results are displayed in a temporary result window.

7.6 Implementing a Prototype of the Proposed Methodology

The following sub-sections explain about implementing a prototype of the proposed methodology for geo-service discovery. Implementation has been performed by evaluating sample ontologies of requested and provided geo-services using a software environment which consists of an ontology editor software and reasoning system. Thus in the following sections, first it is briefly explained about building main ontologies of the layered base structure in the Protégé. Then axioms and constrains used to express concepts related to sample geo-services are discussed. The amounts of matching between these geo-services are determined in the last section.

7.6.1 Building Ontologies of the Layered-Base Structure

The ontologies in the layered-base structure provide a semantic framework for geo-service discovery. It consists of four ontologies including the upper ontology, the ontology of measurement theory, the D&S ontology and the core ontology of geo-services. The Protégé ontology editor and its OWL plug-in have been utilized for formalizing these ontologies in OWL language. The Fig 7.6 shows a snapshot of retrieving the axioms and constraints for *geo-operation* concept in the core ontology of geo-services.

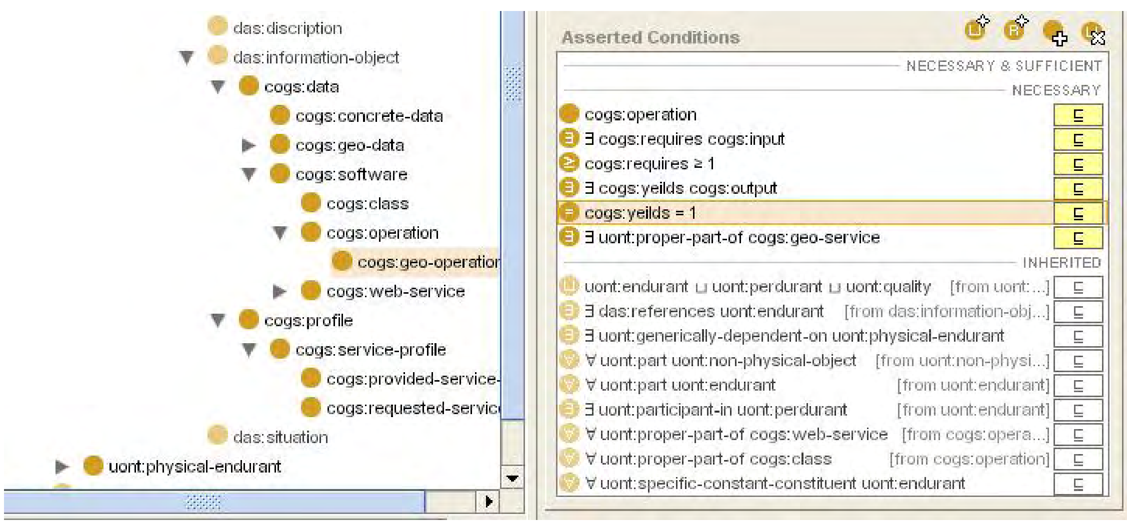

Fig. 7.6: Part of core ontology of geo-services in Protégé with OWL plug-in

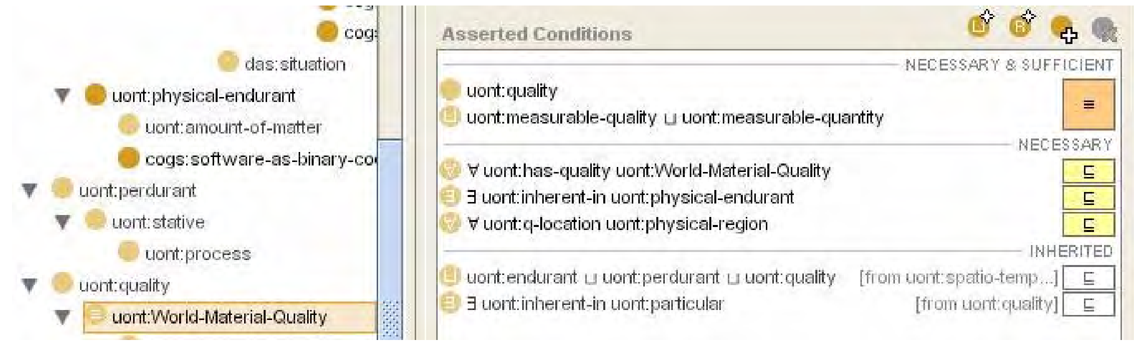

Fig. 7.7: world-material-quality as a defined class with necessary conditions in Protégé with OWL plug-in (see section 7.5 for an explanation about Protégé)

Fig. 7.7 also shows the axioms and constraints for *World-Material-Quality* concept in the upper ontology**.** The details of these ontologies in OWL syntax which is readable by human are presented in the Appendix A.

7.6.2 Building Ontologies for Sample Geo-services

Suppose a modeler needs a service in order to compute runoff rate and there are geo-services for calculating runoff rate. However modeler needs to specify the similarity between his requested geo-service with provided geo-services for discovering a provided geo-service with maximum similarity. The following sub-sections discuss about building the provided and requested ontologies with using concepts in the main ontologies.

7.6.2.1 Building Sample Ontologies for Provided Geo-services

Suppose *profile*, *service* and *operation* concepts for the provided runoff rate services can be described as follow:

$$
\begin{aligned}
& pcr:calculate-runoff-profile \equiv cogs:service-profile \cap \\
& \qquad \exists cogs:describes \cdot pcr:calculate-runoff-service \\
& pcr:calculate-runoff-service \equiv cogs:geo-service \cap \\
& \qquad \exists cogs:part-by \cdot pcr:calculate-runoff-operation \cap \\
& \qquad \forall cogs:part-by \cdot pcr:calculate-runoff-operation \\
& pcr:calculate-runoff-operation \equiv cogs:geo-operation \cap \\
& \qquad \exists cogs:requires \cdot pcr:calculate-runoff-input \cap \\
& \qquad \exists cogs:yields \cdot pcr:calculate-runoff-output
\end{aligned}
$$

where pcr tags is used to uniquely identify concepts in the ontology of provided geo-service. In these cases the intersection of universal $\forall$ and existential $\exists$ restrictions for a given relationship state that "the calculate-runoff-service consists of only one calculate-runoff-operation". The following statements describe the geo-operation concept for the provided runoff rate geo-services:

DEM, land cover value and precipitation rain fall value as field-based data sets are the inputs of calculate-runoff-operation. Thus the calculate-runoff-input concept must be described for this geo-operation by using these input data sets. The runoff rate operation may only need these data sets or it may require other data sets in addition to these data sets. The calculate-runoff-input concept needs additional statement for describing the former case in compare to later case. Since reasoning in OWL (Description Logics) is based on what is known as the open world assumption (OWA). The open world assumption means that it cannot be assumed something doesn’t exist until it is explicitly stated that it does not exist. In other words, because something hasn’t been stated to be true, it cannot be assumed to be false. By applying closure axiom it can be stated that the runoff rate operation only needs DEM, land cover value and precipitation rain fall value as input data sets. Thus the statement for calculate-runoff-input concept along with closure axiom is as follow:

$$
\begin{aligned}
pcr: calculate-runoff-input \equiv cogs: input \cap \exists das: played-by \cdot pcr: DEM \cap \\
\exists das: played-by \cdot pcr: land-cover-value \cap \\
\exists das: played-by \cdot pcr: precipitation-rain-fall-value \cap \\
\forall das: played-by \cdot (pcr: precipitation-rain-fall-value \cup \\
pcr: land-cover-value \cup pcr: DEM)
\end{aligned}
$$

A closure axiom on a property consists of a universal restriction that acts along the property to say that it can only be filled by the specified fillers. The restriction has a filler that is the union of the fillers that occur in the existential restrictions for the property. The statement without closure axiom for calculate-runoff-input concept can be described as follow:

$$pcr: calculate-runoff-input \equiv cogs: input \cap \exists das: played-by \cdot pcr: DEM \cap \\ \exists das: played-by \cdot pcr: land-cover-value \cap \\ \exists das: played-by \cdot pcr: precipitation-rain-fall-value$$

The calculate-runoff-output is described by the following statement:

$$pcr: calculate-runoff-output \equiv cogs: output \cap \\ \exists das: played-by \cdot pcr: runoff-volume \cap \\ \forall das: played-by \cdot pcr: runoff-volume$$

DEM, land cover value, precipitation rain fall value and runoff-volume are formalized as follow by describing their unit of measure and the qualitative or quantitative attribute they represent in the real world:

$$pcr: DEM \equiv cogs: field-data \cap \ni cogs: has-m-uni.mth: Meter \cap \\ \exists cogs: represents.mth: height$$

$$pcr: precipitation-rain-fall-value \equiv cogs: field-data \cap \\ \ni cogs: has-m-uni.mth: Millimeter \\ \cap \exists cogs: represents.mth: precipitation-rain-fall$$

$$pcr: land-cover-value \equiv cogs: field-data \cap \exists cogs: represents.mth: land-cover$$

$$pcr: runoff-volume \equiv cogs: field-data \cap \\ \ni cogs: has-m-uni.mth: LiterPerSquareMeter \\ \cap \exists cogs: represents.mth: runoff$$

where 'pcr', 'cogs', 'mth', 'das', and 'uor' are tags for uniquely identifying the concepts in the ontology of provided runoff geo-service, the core ontology of geo-services, the ontology of measurement, the D&S ontology and the upper ontology respectively.

7.6.2.2 Building Sample Ontologies for Requested Geo-services

Now suppose that an environmental modeler needs a geo-service for deriving runoff rate in order to use it in his model. The model has been introduced as a motivating example for exploring semantic ambiguities in chapter 4 (see section 4.4.1). The following statement describes the profile concept for his ideal geo-service:

$$requested-runoff-profile \equiv cogs: service-profile \cap \\ \exists cogs: describes \cdot requested-runoff-service$$

Accordingly, the geo-service concept is formalized as follow:

$$\begin{aligned} requested-runoff-service \equiv\ & cogs: geo-service \cap \\ & \exists cogs: part-by \cdot requested-runoff-operation \cap \\ & \forall cogs: part-by \cdot requested-runoff-operation \end{aligned}$$

This geo-service is parted by a geo-operation. Sometimes the modeler can describe the input and output of his ideal geo-operation. The following statement describe geo-operation concept:

$$\begin{aligned} requested-runoff-operation \equiv\ & cogs: geo-operation \cap \\ & \exists cogs: requires \cdot requested-runoff-input \cap \\ & \exists cogs: yields \cdot requested-runoff-output \end{aligned}$$

In some cases the modeler can only describe the output of his ideal geo-operation. The geo-operation concept can be formalized as follow:

$$\begin{aligned} requested-runoff-operation \equiv\ & cogs: geo-operation \cap \\ & \exists cogs: yields \cdot requested-runoff-output \end{aligned}$$

The Appendix A shows complete ontology of the provided and requested runoff rate geo-services in OWL syntax which have been created for the purpose of this research.

In order to test the degrees of matching the different ontology are built for requested geo-service by describing the requested-runoff-input and the requested-runoff-output concepts. Then different states of matching are discussed by performing match between provided and requested ontologies describing the runoff rate geo-service.

7.6.3 Matching the Provided and Requested Ontologies

In this regards, profile, geo-service and geo-operation concepts in provided and requested ontologies have been matched. The results of match which are according to degrees of match show the similarity between the requested and provided geo-services.

For instance a modeler describes his ideal runoff rate geo-service by describing its input and output concepts of the requested-runoff-operation as follow:

$$requested-runoff-input \equiv cogs: input \cap \exists das: played-by \cdot DEM \cap$$

$$\exists das : played-by \cdot land-cover-value \cap$$
$$\exists das : played-by \cdot precipitation-rain-fall-value \cap$$
$$\forall das : played-by \cdot (precipitation-rain-fall-value \cup$$
$$land-cover-value \cup DEM)$$

$$requested-runoff-output \equiv cogs : output \cap \exists das : played-by \cdot runoff-volume \cap$$
$$\forall das : played-by \cdot runoff-volume$$

where DEM, land-cover-value and precipitation-rain-fall-value are input data sets and runoff-volume is output data set of requested-runoff-operation. The following statements describe the relation between these field data sets and their unit of measures and qualities they represent.

$$DEM \equiv cogs : field-data \cap \ni cogs : has-m-uni.mth : Meter \cap$$
$$\exists cogs : represents.mth : height$$

$$precipitation-rain-fall-value \equiv cogs : field-data \cap$$
$$\ni cogs : has-m-uni.mth : Millimeter \cap$$
$$\exists cogs : represents.mth : precipitation-rain-fall$$

$$land-cover-value \equiv cogs : field-data \cap \exists cogs : represents.mth : land-cover$$

$$runoff-volume \equiv cogs : field-data \cap \ni cogs : has-m-uni.mth : LiterPerSquareMeter$$
$$\cap \exists cogs : represents.mth : runoff$$

The following statements describe the matching between calculate-runoff-profile concept and requested-runoff-profile concept, calculate-runoff-service concept and requested-runoff-service concept and calculate-runoff-operation concept and requested-runoff-operation concept:

$$requested-runoff-profile \equiv pcr : calculate-runoff-profile$$
$$requested-runoff-service \equiv pcr : calculate-runoff-service$$
$$requested-runoff-operation \equiv pcr : calculate-runoff-operation$$

Fig. 7.8 illustrates the result of matchmaking that have been computed by a DL reasoner and showed in the inferred window of the Protégé ontology editor. In this case, the result is equivalent with "Exact" degree.

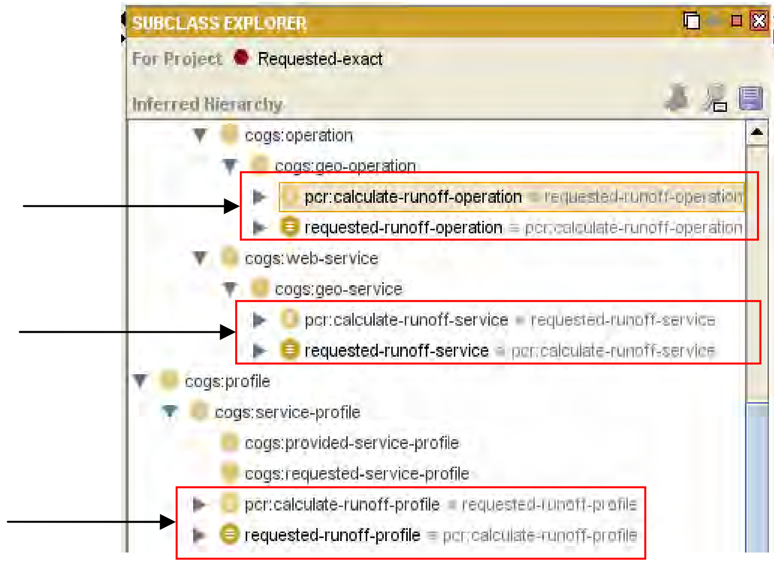

Fig. 7.8: The boxes show "Exact" match between concepts in ontologies of requested runoff geo-service and provided runoff geo-service

This indicates that the requested runoff geo-service is exactly the same as the calculate runoff geo-service. For modelers it means that the input and output of the requested geo-service is the same as the input and output of calculate runoff service. Therefore the calculate runoff service is what the modeler exactly needs.

Now suppose that the calculate runoff service may require other data sets in addition to those mentioned data sets. The statement that describes the calculate-runoff-input concept with no closure axiom has been presented in the section 7.6.2.1.

In this case, the following statements describe the matching between calculate-runoff-profile concept and requested-runoff-profile concept, calculate-runoff-service concept and requested-runoff-service concept and calculate-runoff-operation concept and requested-runoff-operation concept:

$$requested-runoff-profile \subseteq pcr:calculate-runoff-profile$$
$$requested-runoff-service \subseteq pcr:calculate-runoff-service$$
$$requested-runoff-operation \subseteq pcr:calculate-runoff-operation$$

Fig. 7.9 illustrates the result of matchmaking that have been computed by a DL reasoner and showed in the inferred window of the Protégé ontology editor. In this case, the result is equivalent with " Plugin" degree.

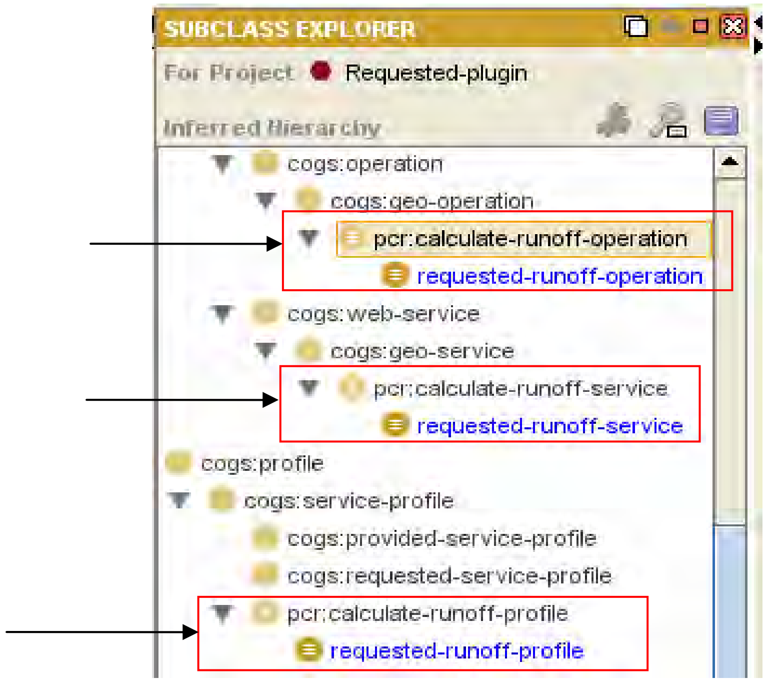

Fig. 7.9: The boxes show "Plugin" match between concepts in ontologies of requested runoff geo-service and provided runoff geo-service

This indicates that the requested runoff geo-service is subclass of the calculate runoff geo-service. For modelers it means that the requested runoff geo-service is a special case of the provided geo-service and, therefore it can satisfy the needs of modeler.

Sometimes modeler can only describe the output of his ideal geo-service. An example for this case is that the pervious requested ontology has no description for the requested-runoff-input concept. Therefore the result of matching calculate-runoff-profile with requested-runoff-profile, calculate-runoff-service with requested-runoff-service and calculate-runoff-operation with requested-runoff-operation, are described as follow:

$$pcr : calculate - runoff - operation \subseteq requested - runoff - operation$$
$$pcr : calculate - runoff - service \subseteq requested - runoff - service$$
$$pcr : calculate - runoff - profile \subseteq requested - runoff - profile$$

Fig. 7.10 illustrates the result of matchmaking that have been computed by a DL reasoner and showed in the inferred window of the Protégé ontology editor. In this case, the result is equivalent with "Subsume" degree.

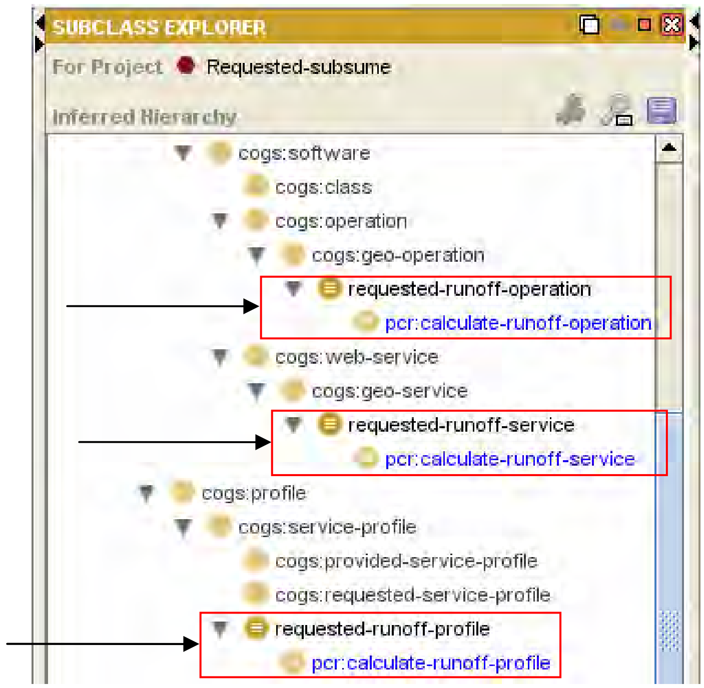

Fig. 7.10: The boxes show "Subsume" match between concepts in ontologies of requested runoff geo-service and provided runoff geo-service

For modelers it means that the requested runoff geo-service is more general than the calculate runoff geo-service and therefore the calculate runoff geo-service may be able to satisfy the needs of modeler.

While there are no "Exact", "Subsume" or "Plugin" match between concepts in the requested and provided ontologies of geo-service, the "Intersection" and "Disjoint" matches should be checked. In this regard the intersection of corresponding concepts i.e. $C_r \cap C_p$ should be created. If the intersection is consistent, then there is an "Intersection" match between concepts otherwise the concepts are disjoint.

In order to demonstrate "Intersection" match the requested runoff service in Subsume match is considered. But its output can be described by the following statement:

$$requested-runoff-output \equiv cogs:output \cap \exists das:played-by \cdot runoff-rate \cap \forall das:played-by \cdot runoff-rate$$

where the runoff-rate can be formalized as follow:

$$runoff-rate \equiv cogs:field-data \cap \ni cogs:has-m-uni.mth:KiloGramPerSquareMeter \cap \exists cogs:represents.mth:runoff \cap \forall cogs:represents.mth:runoff$$

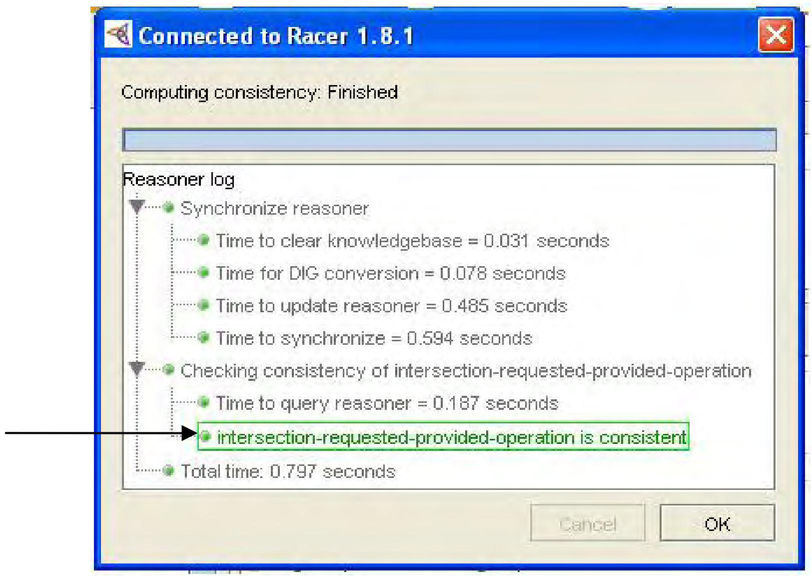

Fig. 7.11: Consistency of intersection of concepts in ontologies of requested runoff geo-service and provided runoff geo-service

The intersection between requested-runoff-operation and calculate-runoff-operation can be described with the following formula:

$$intersection-requested-provided-operation \equiv cogs:geo-operation \cap requested-runoff-operation \cap calculate-runoff-operation$$

Fig 7.11 shows that the intersection is consistent. It means that the intersection of these two concepts is not empty. In this case, the intersection-requested-provided-operation is subclass of the both calculate-runoff-operation and requested-runoff-operation as illustrated in Fig. 7.12. Therefore, there is a little chance that the requested runoff geo-service satisfies the needs of the modeler.

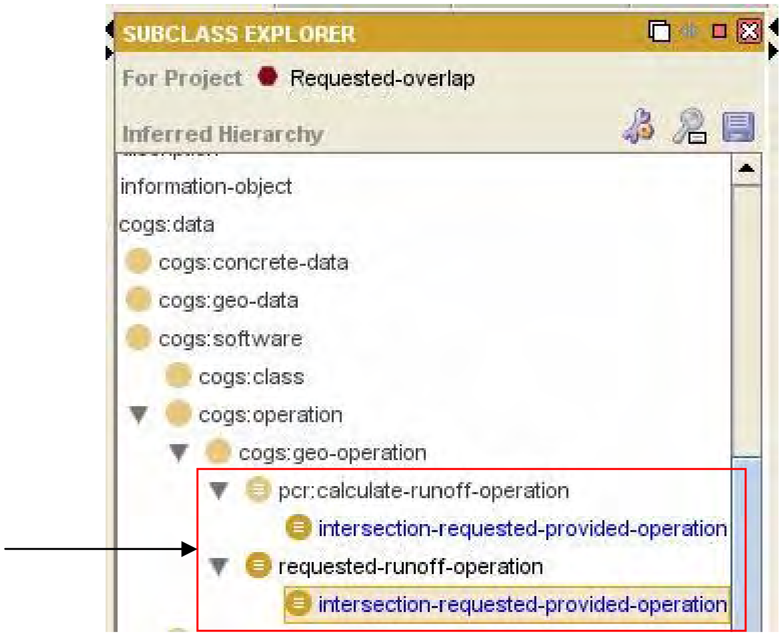

Fig. 7.12: The box shows "Intersection" match between concepts in ontologies of requested runoff geo-service and provided runoff geo-service

As an example for demonstrating the "Disjoint" match, it is supposed that the runoff-rate concept in "Intersection" match is described as follow:

$$runoff - rate \equiv cogs : field - data \cap \neg pcr : runoff - volume$$

In this case the intersection of the requested-runoff-operation and calculate-runoff-operation is inconsistent as it is illustrated in Fig. 7.13. Therefore, there is no chance that the requested runoff geo-service satisfies the needs of modeler.

Fig. 7.13 Inconsistency of intersection of concepts in ontologies of requested and provided runoff geo-services

7.6.4 Geo-service Finder Application

The pervious section discussed about using appropriate commands which are available in the Protégé ontology editor interface in order to match corresponding concepts in the provided and requested ontologies of geo-services. These commands check consistency, classify taxonomy and compute inferred types for all concepts or check concept consistency, compute individuals belonging to a class and get inferred super classes / subclasses for a distinct concept. The result of classification for all concepts is shown in the inferred window and the result of other commands is shown in a modal window. However the match degree is not explicitly computed by these commands. Further for identifying "overlap" and "Disjoint" match of the corresponding concepts, it is needed to build the intersection concept of these concepts and then compute consistency of the intersection concept.

Due to these reasons, a standalone program with a user interface has been developed to compute and illustrate the degree of matchmaking between corresponding concepts. This program is called geo-service finder and written in

Java language by utilizing Java packages of the Protégé in order to access to the model of OWL ontology and reasoner using DIG interface.

Fig 7.14 illustrates a snapshot of the user interface for this program. The source of the program is presented in Appendix B.

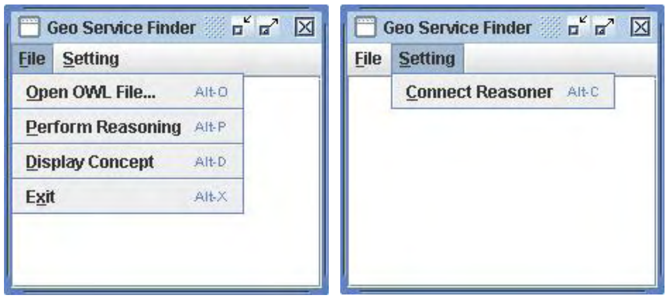

Fig. 7.14: Snapshots of the geo-service finder

The sample ontologies for requested geo-services which have been discussed in pervious sections have been loaded in this program and their similarities with the sample ontologies of the provided geo-services are determined using access to functionalities of the RacerPro inference engine.

For example Fig 7.15 shows the concepts of the sample ontologies where their requested and provided geo-services are of exact similarity.

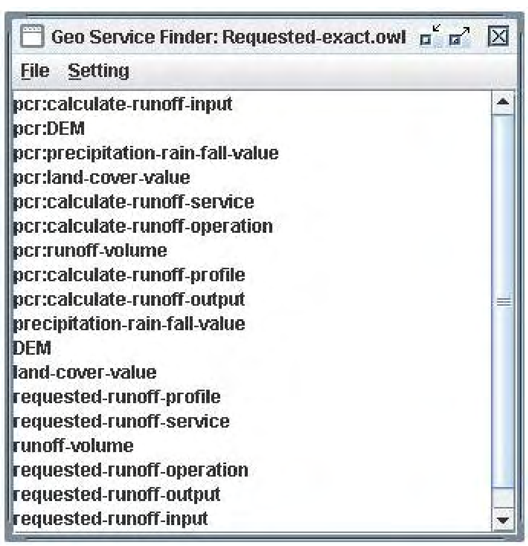

Fig. 7.15: Concepts in the sample ontologies of the requested and provided geo-services when they have exact match

The result of inference between corresponding concepts shows matching degree between requested and provided geo-services. In this case the degree is exact which means that the input and output of the requested geo-service are exactly the same as input and output of the provided geo-services (see Fig 7.16).

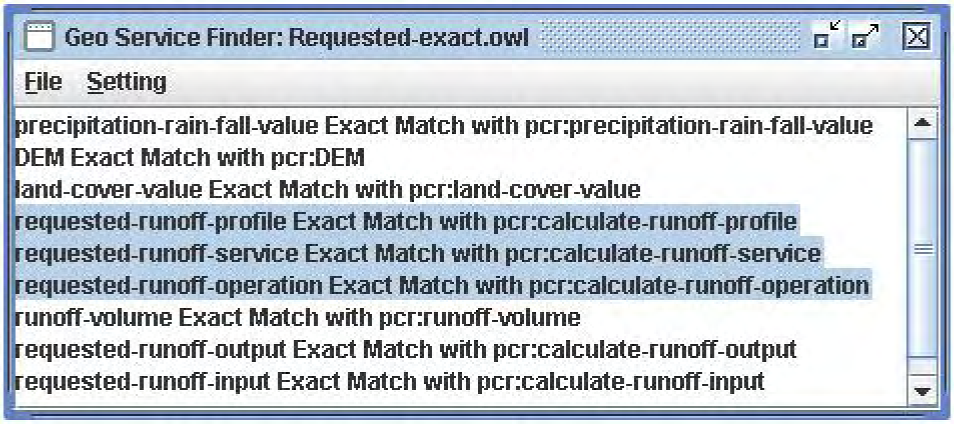

Fig. 7.16: The result of inference between correspondence concepts when they have exact match

Fig 7.17 shows the concepts of the sample ontologies where the concepts related to the requested geo-service are plugged into the concepts of the provided geo-service.

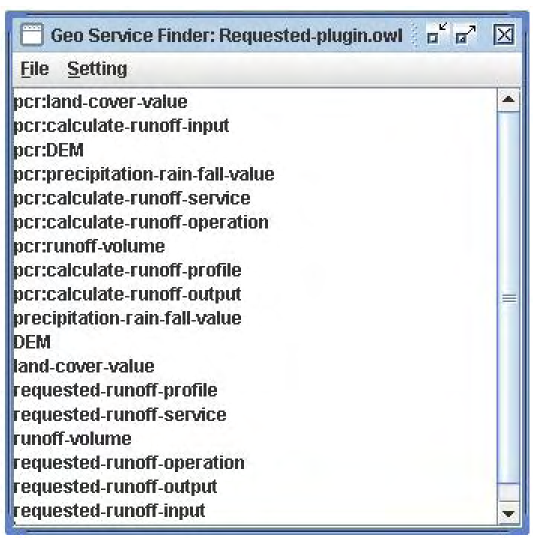

Fig. 7.17: Concepts in the sample ontologies of the requested and provided geo-services when they have plugin match

The result of inference which shows the matching degree between corresponding concepts, in this case, is plug-in which means that the input and output of the requested geo-service are specialized cases of the input and output of the provided geo-service (see Fig 7.18).

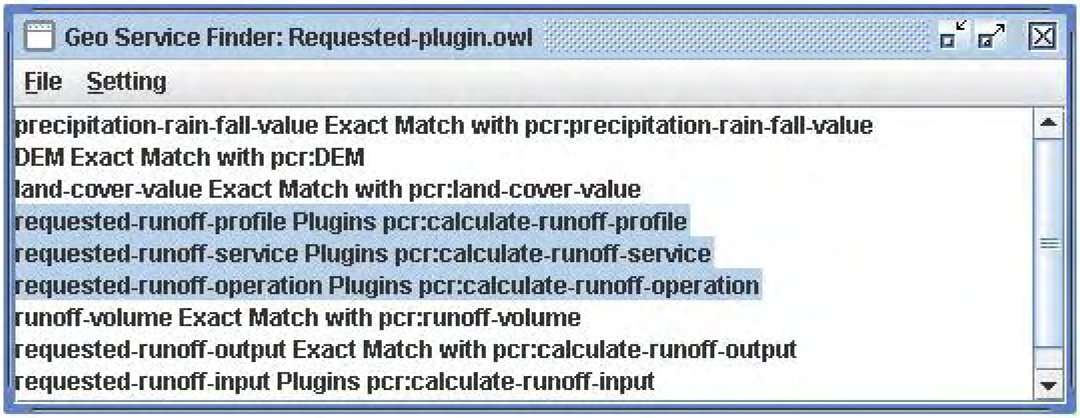

Fig. 7.18: The result of inference between correspondence concepts when they have plugin match

Fig 7.19 shows the concepts of the sample ontologies where the concepts related to the requested geo-service subsume the concepts of the provided geo-service.

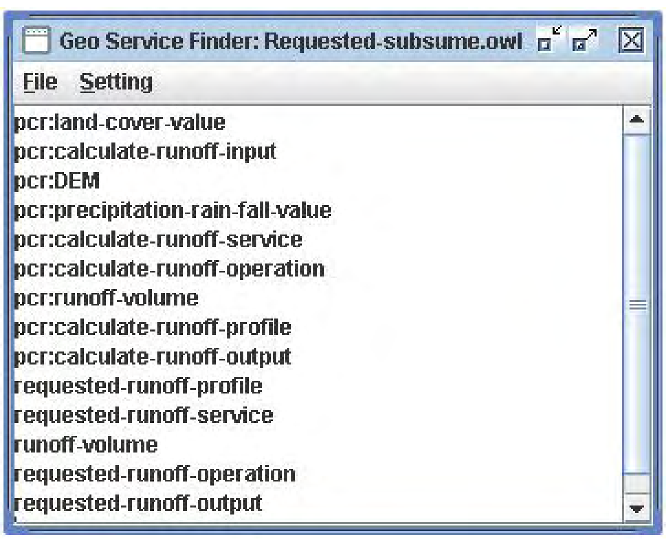

Fig. 7.19: Concepts in the sample ontologies of the requested and provided geo-services when they have subsume match

The result of inference which shows the matching degree between corresponding concepts, in this case, is subsume which means that the input and output of the requested geo-service are generalized cases of the input and output of the provided geo-service (see Fig 7.20).

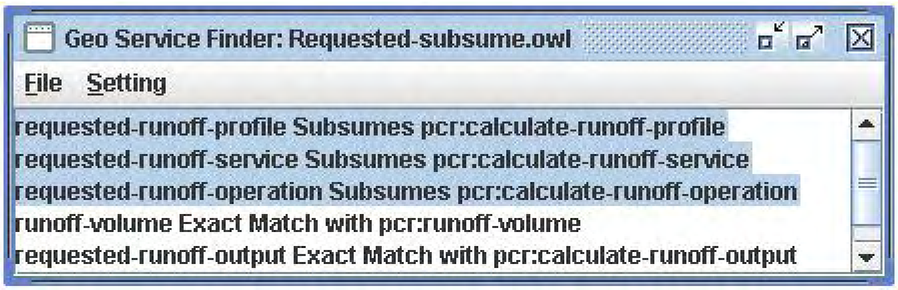

Fig. 7.20: The result of inference between correspondence concepts when they have subsume match

In order to discover the overlap or disjoint similarities between correspondence concepts in the requested and provided ontologies, their intersection are computed and the consistency of the intersection concept are determined automatically by program and then the result of matching is displaced.

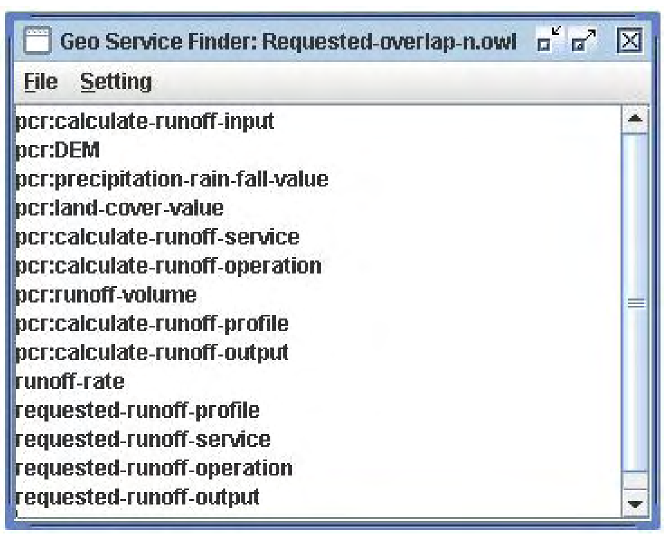

Fig. 7.21: Concepts in the sample ontologies of the requested and provided geo-services when they have intersection match

Fig 7.21 shows the concepts of the sample ontologies where the concepts related to the requested geo-service overlap with the concepts of the provided geo-service.

The result of inference which shows the matching degree between corresponding concepts, in this case, is overlap which means that the input and output of the requested geo-service have common aspects with the input and output of the provided geo-service (see Fig 7.22).

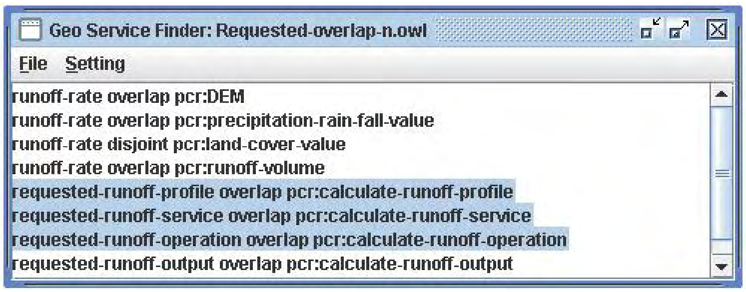

Fig. 7.22: The result of inference between correspondence concepts when they have intersection match

Fig 7.23 shows the concepts of the sample ontologies where the concepts related to the requested geo-service are disjoint from the concepts of the provided geo-service.

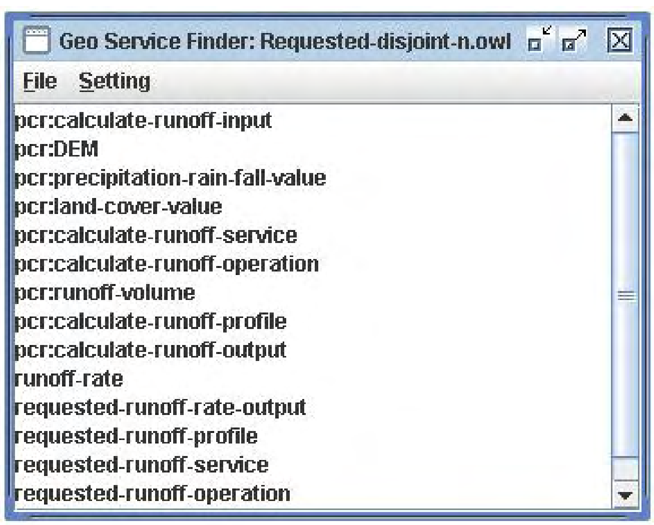

Fig. 7.23: Concepts in the sample ontologies of the requested and provided geo-services when they have disjoint match

The result of inference which shows the matching degree between corresponding concepts, in this case, is disjoint which means that the input and output of the requested geo-service have no common aspects with the input and output of the provided geo-service (see Fig 7.24).

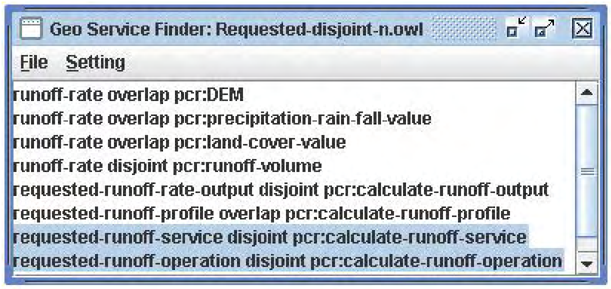

Fig. 7.24: The result of inference between correspondence concepts when they have disjoint match

7.7 Discussion

The aim of the current research has been to propose a methodology based on ontologies for discovering geo-databases to research about main required ontologies for this methodology. In this regard the upper ontology, the core ontology of geo services and the ontology of measurement theory as main required ontologies have been developed by recognizing semantic ambiguities related to geo-services. Generally, development of ontologies is a difficult and consequently expensive task. Major tasks here were extracting axioms and constraints for important concepts and formalizing them. However, it is a price worth paying to avoid semantic conflicts (which in many cases will be even more expensive). Further, ontologies are long-term assets that will remain independent of the application systems. As ontologies are becoming more popular, they can also be used for other purposes. Ontologies for provided and requested geo-services are Built based on a semantic framework supported by the main ontologies.

Note that discovery of geo-services was only considered based on input and output of the geo-services in the current research. In some cases such as commercial services it may be needed to consider preconditions that must be satisfied before executing the geo-services or effects that geo-services have after execution. However field-based geo-services which are the focus of the current research often do not require preconditions and effects.

In order to demonstrate the use of the reasoner in detecting match degree in the ontologies a number of samples ontologies for the provided and requested geo-services have been created. This strategy has been used as a check so that it can be seen that the ontologies have been built correctly.

7.8 Conclusion

This chapter has been introduced architecture of the proposed methodology for discovery of geo services and explained its components. Issues related to architecture its functionalities and specifications which are required to deploy the methodology were explored. Approach of building ontologies and environment which have been used for developing ontologies and performing match were discussed.

As it has been explained in the pervious chapters, semantics ambiguities are obstacles for linking GIS and environmental models based on distributed computing architecture with loosely coupled geo-services. In the proposed methodology for eliminating these ambiguities such as unit of measure or measurement scale, the related concepts in the requested and provided ontologies for sample geo-services are expressed by applying a Description Logics language. Then the sample ontologies are evaluated by using a DL reasoning system as inferential engine of the proposed methodology for discovering amount of matching between them. Capabilities of the methodology depend on expressiveness of the language and the reasoning capabilities of the underlying reasoning system. On the other hand, the availability of a reasoning system to perform desired processes depends on the expressive power of language. Very expressive language is likely to have inference problems of high complexity. Very weak language (with efficient reasoning procedures) may not be sufficiently expressive to represent the important concepts of a given application. OWL-DL language has been selected for describing ontologies developed in the current research. Since it is expressive enough to describe statements related to geo-services and there are also reasoning systems in OWL-DL language in order to perform matching between corresponding concepts.

Further, with eliminating semantic ambiguities in the GIS and environment domains due to the existence of a semantic framework supported by main ontologies, the discovery of geo-services can be performed with more precision.

Chapter 8 **Conclusion and Future Works**

8.1 Introduction

This chapter summarizes the outcomes of the research and the abilities of the proposed methodology for discovery of geo-services and presents its weaknesses. Then the position of this research with respect to other related researches is discussed. The chapter then presents the possible future improvements and conclusions.

8.2 Summary and Conclusion

There are many physical processes such as surface flow, soil erosion or infiltration occurs in natural environment. The natural environments have been directly or indirectly influenced by humans at various locations in time. The environmental sustainability is to minimize environmental degradation that occurs when consuming natural resources faster than the nature can replenish, destroying ecosystems in the process of development caused by human activities or pollution results in irreparable damage done to the environment. Environmental problems have geospatial nature, therefore, the modeler needs GIS to investigate and predict relationship between phenomena.

Integrated or tightly coupled GIS and environmental models can be suitable for certain purposes but the source code is not accessible for user in order to change it for other purposes. In this case, the details of the GIS functionality are coded inside the system so that it is not readable and understandable for user. In order to avoid these drawbacks, the research proposed that a distributed computing architecture based on loosely coupled geo-services can satisfies the needs of modeler to GIS. Modelers conceptualize the environment as quantitative and measurable phenomena and deal with field-based geospatial data to describe the quantitative and qualitative aspects of physical processes. Therefore, this research focused on field-based geo-service to produce new field-based geospatial data for environmental model.

A geo-service according to modeler's request must be discovered before accessing or invoking it. The messages exchanged between requested and provided geo-services are formed according to industrial standard protocols which were briefly discussed in this research. However, these protocols provide syntactic interoperability for terms and terminology used in the exchanged messages rather than semantic interoperability. The semantic ambiguities and implicit details make barriers against geo-services discovery. This research investigated semantic ambiguities and details related to the field-based geo-services.

An outcome of this research is a layered-base structure of ontologies for resolving semantic ambiguities and describing details in geo-services. The general concepts which are frequently used in the field of geo-services and environmental modeling have been recognized by investigating existing ontologies. In this regard, the taxonomy of Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) was identified as framework for upper ontology. The ontology of measurement theory has included concepts related to unit of measure and measurement scale and the core ontology of geo-services has included concepts related to software and geo-services. These ontologies form three layers of the structure which have been developed by this research. The existence of Descriptions and Situations (D&S) ontology in the structure is essential to fill conceptual gaps between concepts in the upper ontology and the core ontology of geo-services.

The layered-base structure of ontologies contains knowledge-base for the methodology of geo-service discovery proposed by this research. The ontologies of the structure are used to approximate the subtle distinction of concepts and their relationships as well as explicitly describe the details related to field-based geo-services. The subtle distinction in the meaning may be emerged between two agents, let say geo-services provider and requester due to subtle distinction in the meaning of general terms such as process and operation.

Building ontologies from the scratch is time consuming and labor intensive and may be extremely hard. Therefore, major effort has been taken in this research for developing these ontologies.

The degree of formality employed in capturing descriptions of concepts and relationships can be quite variable, ranging from natural language to logical formalisms, but increased formality and regularity clearly facilitates machine understanding. Description Logics (DL) provide the formal foundation for modern ontology languages. DL also allows accomplishment of reasoning on concepts and relation expression represented by first-ordered predicates. In this regard the OWL (Web Ontology Language) as a DL based ontology and central standard for ontologies on the Semantic Web has been investigated. The OWL language is expressive enough in order to formalize the concepts and relationships covered by ontologies of the layered-base structure and the sample ontologies of requested and provided geo-services. Therefore, these ontologies have been formalized by using OWL. The sample ontologies of requested and provided geo-services can provide semantically richer geo-service descriptions and facilitate machine-based communication between requester (modeler) and provider (GIS) and can more flexibly interpret by intelligent agents.

Another outcome of this research was a methodology for geo-service discovery based on ontology composed of layered-based structure of ontologies as knowledge-base and a DL reasoner as inference engine for evaluating the requested and provided ontologies of sample geo-services. The sample ontologies describe the details of the input and output of requested and provided geo-services. Therefore, matching between the requested and provided ontologies shows the similarity between input and output of the requested and provided geo-services. In this regard a set of matching degrees has been identified which can be used as measurement scale in order to determine the amount of similarity.

The evaluation of the research was performed by building ontologies at the top and domain levels and developing sample ontologies for requested and provided geo-services which were represented in this research. Matchmaking between requested and provided geo-services was performed by using Racer as DL reasoner. In this regards the ontologies were created and managed in java based software called Protégé [Protégé, 2003]. Protégé was connected to these reasoners through the DIG interface and performed match between requested and provided ontologies.

8.3 Position of the Research with Respect to State of the Art

In the current efforts for linking environmental models and GIS, integrated or tightly coupled systems are produced for specific tasks [Fedra, 1993]. The common drawback of current methods is that they can not make a solution for removing the terminological and conceptual ambiguities due to the fact that the details of GIS functions are coded inside the systems. Different efforts have been established to design interface that the user can select the operation but current approach could not completely solve it [Bruns and Egenhofer, 1997]. This research investigated that distributed computing architecture based on loosely coupled geo services avoids these drawbacks. In this case, the user can access various geo-services and customized them for the desired situation.

One of the weaknesses of this architecture is parameter meaning and for transferring parameter values from one model to another it is needed to use a common language as an essential component of any communications [Hutchings et al, 2002]. Aim of ARION (2003) project is to develop digital library based on the coupling of ontologies with metadata and workflows for the domain of ocean and meteorology. Feng et al (2004) have been approached to model the semantics of hydrologic processes in surface hydrology in terms of endurance, perdurance, and granularity notions. However, the state of art this research is to propose a layered-base structure of ontologies as knowledge-base for approximating the intended meaning of concepts and relations for environmental modeling and GIS domain. These ontologies and their usage for discovering field-base geo-service is novel.

Probst, and Lutz, (2004) proposed the semantic reference system based on image schemata for top level ontology. They used domain or conceptual ontology for representing the vocabulary humans which is used to communicate about domain such as meteorology.

However, from ontological point of view, an ontological structure was developed in which the upper ontology, the ontology of measurement theory and the core ontology of geo-services were built so that the provider and requester of geo-services commit to them. In one hand the upper ontology covers general notions related to GIS and environmental models. On the other hand specific domain notions related to measurement theory and geo-services can be plugged in to the concepts of upper ontology. Ontological structure as knowledge-base contains concepts which could be communicated between experts or scientists.

Ontological structure proposed in this research is also different from EngMath [Gruber, 1993] in two aspects. In one hand the research followed the top, domain and application hierarchy [Guarino, 1997] for developing ontology in order to establish precise agreements between modeler and GIS while EngMath inherits from KIF which is used as logical language for representing ontology. On the other hand, the ontology of measurement theory and the core ontology of geo-services have been developed for describing concepts at domain level such as measurement scale and unit of measure. These ontologies are missing in EngMath ontology.

This research put the ontologies in practices for discovering geo-services by approximating the intended meaning of concepts in environmental modeling and GIS as well as explicitly describing details of geo-services.

Formalized ontologies provide the modeler with the advantage of independence from GIS's background knowledge. That is, the modelers and GIS experts specify their knowledge about the vocabulary related to environmental models and geo-services. Also modeler specifies description for his/her required geo-service. Afterwards, discovering the desired geo-service can be performed by matching ontologies of the requested and provided geo-services, with no dependency to background knowledge of modeler about geo-services.

8.4 Direction for Future Works

Environmental models consist of physical processes which are depending on time. Quantities or qualities such as magnitude and direction of wind which represent different aspects of environment in physical process often change with time. They can be described with field-based geospatial data. In other words the value of fields may change with time. The role of time in physical processes depends on how to model these processes. For examples, a river modeling will, naturally, commit to the existence of “river” entities. However, whether the model is designed for the flows over (relatively) short periods or of river morphology over longer ones will decide whether the river geometry is considered to be fixed or changing. The changes may be observed in some models over a long period such as geomorphology and in some others over shorter one like plant growing. Therefore, these changes and processes are depending on type of the model and time is a very basic concept to describe the processes. In addition to this research there is a need to investigate the concepts of time in linking GIS and environmental models, since specific knowledge of model is needed. Therefore, research work is required to study the concept of time in different environmental models and GIS.

This research considered GIS as a repository of geo spatial data and geo-services library for interpreting and analyzing the geo spatial data. GIS can assist environmental modelers to apply geospatial data and geo-services to reach the aim of sustainable development. However the geospatial data and geo-services may be distributed across different physical locations. In this regard, Spatial Data Infrastructure (SDI) provides facilities to share discover and access geospatial data and geo-services. Due to the fact that the modelers, producer of geospatial data, or developer of geo-services may have different background knowledge about geospatial data and geo-services, the semantic ambiguities and implicit details make barriers against discovery, sharing and accessing across SDI. Therefore mediation is needed in order to connect the requirement of the modelers to geo-services and geospatial data which are offered by developers and producers. In this regard, solving differences in conceptualization between users, developers and producers which is an extensive task should be managed by mediator. Therefore, an ontological infrastructure is needed in order to install on top of SDI platform to enhance discovery, sharing and accessing across SDI. The methodology proposed by this research can be used as a part of solution; however, research is needed to study architecture and components of ontological infrastructure.

Aim of this research was linking environmental modeling and GIS by using ontology. This research focused on field-based geo-services which are applicable in environmental models. However, in some cases, the object-based geo-services may be applicable in environmental problems. Thus a research objective can be defined to investigate link of environmental models with object-based geo-services.

Appendix A: Main Ontologies

The following ontology describes the upper ontology which has been developed based on DOLCE. The part of ontology which has been edited or created are illustrated here.

```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/UpperOnt.owl">
  <owl:Ontology rdf:about="">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The UpperOnt ontology. OWL engineering by G.R. Fallahi.</rdfs:comment>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
  </owl:Ontology>
  <owl:Class rdf:ID="spatio-temporal-particular">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:ID="endurant"/>
              <owl:Class rdf:ID="perdurant"/>
              <owl:Class rdf:ID="quality"/>
            </owl:unionOf>
          </owl:Class>
          <owl:Class rdf:ID="particular"/>
```

```
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="measurable-quality">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >A physical quality inherent in a physical endurant.</rdfs:comment>
    <owl:disjointWith>
      <owl:Class rdf:ID="measurable-quantity"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="World-Material-Quality"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="physical-region">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >A region at which only physical attributes can be  directly located. It assumes
some metrics for physical properties.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#region"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#part"/>
        </owl:onProperty>
        <owl:allValuesFrom rdf:resource="#physical-region"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="q-location-of"/>
        </owl:onProperty>
        <owl:allValuesFrom>
          <owl:Class rdf:about="#World-Material-Quality"/>
        </owl:allValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#measurable-quantity">
    <owl:disjointWith rdf:resource="#measurable-quality"/>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#World-Material-Quality"/>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >A physical quantity inherent in a physical endurant.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#World-Material-Quality">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom rdf:resource="#World-Material-Quality"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#has-quality"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
```

```xml
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="q-location"/>
        </owl:onProperty>
        <owl:allValuesFrom rdf:resource="#physical-region"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#measurable-quality"/>
              <owl:Class rdf:about="#measurable-quantity"/>
            </owl:unionOf>
          </owl:Class>
          <owl:Class rdf:about="#quality"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom rdf:resource="#physical-endurant"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#inherent-in"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >A quality inherent in a physical endurant.</rdfs:comment>
  </owl:Class>
</rdf:RDF>
```

The following ontology describes the measurement theory which has been developed during this research. Measurement scale and unit of measure concepts have been described in this ontology.


```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/MeasureTheory.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The Measurement Theory ontology. OWL engineering by G.R.Fallahi.</rdfs:comment>
  </owl:Ontology>
  <owl:Class rdf:ID="unit-of-measure">
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#physical-region"/>
  </owl:Class>
  <owl:Class rdf:ID="system-international-unit">
    <rdfs:subClassOf rdf:resource="#unit-of-measure"/>
  </owl:Class>
  <owl:Class rdf:ID="height">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="ratio-quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="runoff">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#ratio-quantity"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="interval-quantity">
    <owl:disjointWith>
      <owl:Class rdf:about="#ratio-quantity"/>
    </owl:disjointWith>
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#measurable-quantity"/>
  </owl:Class>
  <owl:Class rdf:ID="ordinal-quality">
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#measurable-quality"/>
    <owl:disjointWith>
```

```xml
      <owl:Class rdf:ID="nominal-quality"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:about="#ratio-quantity">
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#measurable-quantity"/>
    <owl:disjointWith rdf:resource="#interval-quantity"/>
  </owl:Class>
  <owl:Class rdf:ID="precipitation-rain-fall">
    <rdfs:subClassOf rdf:resource="#ratio-quantity"/>
  </owl:Class>
  <owl:Class rdf:about="#nominal-quality">
    <owl:disjointWith rdf:resource="#ordinal-quality"/>
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#measurable-quality"/>
  </owl:Class>
  <owl:Class rdf:ID="land-cover">
    <rdfs:subClassOf rdf:resource="#nominal-quality"/>
  </owl:Class>
  <unit-of-measure rdf:ID="LitrPerSquareMeter"/>
  <unit-of-measure rdf:ID="KiloGramPerSquareMeter"/>
  <unit-of-measure rdf:ID="Millimeter"/>
  <unit-of-measure rdf:ID="Meter"/>
</rdf:RDF>
```

The following ontology developed during this research, describes concepts related to geo-services. The concepts such as Software, geo-service or geo-operation have been described by this ontology.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl">
  <owl:Ontology rdf:about="">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The Core Ontology of geo services. OWL engineering by G.R.Fallahi.</rdfs:comment>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="geo-operation">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class rdf:ID="geo-service"/>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:TransitiveProperty rdf:ID="part-of"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="yeilds"/>
        </owl:onProperty>
        <owl:allValuesFrom>
          <owl:Class rdf:ID="output"/>
        </owl:allValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="requires"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
```

```
      >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#yeilds"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#requires"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="input"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="operation"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#yeilds"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#output"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="requested-service-profile">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#geo-service"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="describes"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="service-profile"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#geo-service">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
```

```
        <owl:TransitiveProperty rdf:ID="part-by"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#geo-operation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:TransitiveProperty rdf:about="#part-by"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#geo-operation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="web-service"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="concrete-data">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="data"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://www.ncc.org.ir/ontologies/UpperOnt.owl#physical-
endurant"/>
<owl:Class rdf:ID="geo-data">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#data"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="provided-service-profile">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#service-profile"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#describes"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#geo-service"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#data">
  <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#information-object"/>
</owl:Class>
<owl:Class rdf:about="#web-service">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="software"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="class">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#software"/>
  </rdfs:subClassOf>
```

```
  </owl:Class>
  <owl:Class rdf:about="#operation">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#proper-part-of"/>
        <owl:allValuesFrom rdf:resource="#class"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom rdf:resource="#web-service"/>
        <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#proper-part-of"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#software"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#output"/>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#yeilds"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#requires"/>
        </owl:onProperty>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#input"/>
        </owl:someValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://www.ncc.org.ir/ontologies/DandS.owl#role"/>
  <owl:Class rdf:about="#service-profile">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom rdf:resource="#geo-service"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#describes"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="profile"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="object-data">
```

```
    <rdfs:subClassOf rdf:resource="#geo-data"/>
  </owl:Class>
  <owl:Class rdf:about="#input">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
        <owl:someValuesFrom>
          <owl:Class rdf:ID="field-data"/>
        </owl:someValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#role"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class rdf:ID="computational-task"/>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="input-for"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="software-as-binary-code">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="instrument-of"/>
        </owl:onProperty>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#computational-task"/>
        </owl:someValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#physical-endurant"/>
  </owl:Class>
  <owl:Class rdf:about="#profile">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom>
          <owl:Class rdf:about="#software"/>
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#describes"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#information-object"/>
  </owl:Class>
  <owl:Class rdf:about="#field-data">
    <rdfs:subClassOf>
      <owl:Restriction>
```

```
        <owl:someValuesFrom>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#unit-of-measure">
            <rdfs:subClassOf>
              <owl:Restriction>
                <owl:allValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#World-Material-Quality"/>
                <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#q-location-of"/>
              </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
              <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#q-location-of"/>
                <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#World-Material-Quality"/>
              </owl:Restriction>
            </rdfs:subClassOf>
            <rdfs:subClassOf>
              <owl:Restriction>
                <owl:someValuesFrom rdf:resource="#field-data"/>
                <owl:onProperty>
                  <owl:ObjectProperty rdf:ID="m-unit-of"/>
                </owl:onProperty>
              </owl:Restriction>
            </rdfs:subClassOf>
          </rdf:Description>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="has_m-uni"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/UpperOnt.owl#World-Material-Quality"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="represents"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#geo-data"/>
  </owl:Class>
  <owl:Class rdf:ID="task">
    <rdfs:subClassOf rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#course"/>
  </owl:Class>
  <owl:Class rdf:about="#output">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
        <owl:someValuesFrom rdf:resource="#field-data"/>
      </owl:Restriction>
```

```
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="output-for"/>
        </owl:onProperty>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#computational-task"/>
        </owl:someValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#role"/>
  </owl:Class>
  <owl:Class rdf:about="#software">
    <rdfs:subClassOf rdf:resource="#data"/>
  </owl:Class>
  <owl:Class rdf:about="#computational-task">
    <rdfs:subClassOf rdf:resource="#task"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="described-by">
    <rdfs:domain rdf:resource="#software"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#aboutness-of"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#describes"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#yeilds">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >a relation between operation and its input and output parameters can be perform by
opertionRequires and operationYeilds</rdfs:comment>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="is-yield-by"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#has_m-uni">
    <rdfs:range rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#unit-
of-measure"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#m-unit-of"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#input-for">
    <rdfs:range rdf:resource="#computational-task"/>
    <rdfs:domain rdf:resource="#input"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >a relation between operation and its input and output parameters can be perform by
opertionRequires and operationYeilds</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#represents">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="is-represented-by"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#describes">
```

```xml
    <owl:inverseOf rdf:resource="#described-by"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#about"/>
    <rdfs:range rdf:resource="#software"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#is-represented-by">
    <owl:inverseOf rdf:resource="#represents"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#requires">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="is-required-by"/>
    </owl:inverseOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >a relation between operation and its input and output parameters can be perform by
opertionRequires and operationYeilds</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="profiles">
    <rdfs:subPropertyOf
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#about"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="profiled-by"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#m-unit-of">
    <rdfs:domain rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#unit-
of-measure"/>
    <owl:inverseOf rdf:resource="#has_m-uni"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#profiled-by">
    <owl:inverseOf rdf:resource="#profiles"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#aboutness-of"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#output-for">
    <rdfs:range rdf:resource="#computational-task"/>
    <rdfs:domain rdf:resource="#output"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >a relation between operation and its input and output parameters can be perform by
opertionRequires and operationYeilds</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#instrument-of">
    <rdfs:range rdf:resource="#computational-task"/>
    <rdfs:domain rdf:resource="#software-as-binary-code"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >a relation between operation and its input and output parameters can be perform by
opertionRequires and operationYeilds</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#is-required-by">
    <owl:inverseOf rdf:resource="#requires"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="data-scale">
    <rdfs:domain rdf:resource="#concrete-data"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#is-yield-by">
    <owl:inverseOf rdf:resource="#yeilds"/>
```

```xml
  </owl:ObjectProperty>
  <owl:TransitiveProperty rdf:about="#part-by">
    <owl:inverseOf>
      <owl:TransitiveProperty rdf:about="#part-of"/>
    </owl:inverseOf>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </owl:TransitiveProperty>
  <owl:TransitiveProperty rdf:about="#part-of">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <owl:inverseOf rdf:resource="#part-by"/>
  </owl:TransitiveProperty>
</rdf:RDF>
```

Appendix B: Provided and Requested Ontologies of Sample Geo-Services

These ontologies have been developed for describing provided and requested geo-services. The sample geo-services have been used for implementing the proposed methodology of this research.

The following ontology describes provided geo-service which computes runoff rate (with closure axiom)


```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.ncc.org.ir/ontologies/Provided-Calculate-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Provided-Calculate-runoff.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl"/>
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="calculate-runoff-input">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#input"/>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="DEM"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="precipitation-rain-fall-value"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="land-cover-value"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:allValuesFrom>
              <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                  <owl:Class rdf:about="#precipitation-rain-fall-value"/>
                  <owl:Class rdf:about="#land-cover-value"/>
                  <owl:Class rdf:about="#DEM"/>
                </owl:unionOf>
              </owl:Class>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="calculate-runoff-service">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
```

```
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="calculate-runoff-operation"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:about="#calculate-runoff-operation"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#DEM">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Meter"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#height"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#precipitation-rain-fall-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#precipitation-rain-
fall"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
```

```
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Millimeter"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="runoff-volume">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#LitrPerSquareMeter"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="calculate-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#calculate-runoff-service"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#calculate-runoff-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#calculate-runoff-input"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#requires"/>
          </owl:Restriction>
```

```xml
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="calculate-runoff-output"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#land-cover-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#land-cover"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#calculate-runoff-output">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#runoff-volume"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:allValuesFrom rdf:resource="#runoff-volume"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <rdf:List>
    <rdf:first
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#input"/>
</rdf:RDF>
```

The sample ontology for describing provided geo-service which computes runoff rate. The inputs of this geo-service has not exactly been identified (with no closure axiom)


```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.ncc.org.ir/ontologies/Provided-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Provided-runoff.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl"/>
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
  </owl:Ontology>
      <owl:Class rdf:ID="land-cover-value">
            <owl:equivalentClass>
                  <owl:Class>
                        <owl:intersectionOf rdf:parseType="Collection">
                              <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
                              <owl:Restriction>
                                    <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#land-cover"/>
                                    <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
                              </owl:Restriction>
                        </owl:intersectionOf>
                  </owl:Class>
            </owl:equivalentClass>
      </owl:Class>
      <owl:Class rdf:ID="calculate-runoff-input">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#input"/>
```

```
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="DEM"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="precipitation-rain-fall-value"/>
            </owl:someValuesFrom>
          </owl:Restriction>
              <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
                  <owl:someValuesFrom rdf:resource="#land-cover-value"/>
            </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="calculate-runoff-service">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="calculate-runoff-operation"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:about="#calculate-runoff-operation"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#DEM">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
```

```
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Meter"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#height"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#precipitation-rain-fall-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#precipitation-rain-
fall"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Millimeter"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="runoff-volume">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#LitrPerSquareMeter"/>
          </owl:Restriction>
```

```xml
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="calculate-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#calculate-runoff-service"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#calculate-runoff-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#calculate-runoff-input"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#requires"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="calculate-runoff-output"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#calculate-runoff-output">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#runoff-volume"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
```

```
            <owl:allValuesFrom rdf:resource="#runoff-volume"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

The following ontology describes a sample of requested geo-service which computes runoff rate. This ontology has been used to show the "Exact" match between this ontology and the ontology describing provided geo-service with closure axiom.


```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:pcr="http://www.ncc.org.ir/ontologies/Provided-Calculate-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.ncc.org.ir/ontologies/Requested-runoff.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Requested-runoff.owl">
  <owl:Ontology rdf:about="">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/Provided-Calculate-
runoff.owl"/>
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="precipitation-rain-fall-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Millimeter"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#precipitation-rain-
fall"/>
          </owl:Restriction>
```

```
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="DEM">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#height"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Meter"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="land-cover-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#land-cover"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-service"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
```

```
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="runoff-volume">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#LitrPerSquareMeter"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-service">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-operation"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:allValuesFrom>
              <owl:Class rdf:about="#requested-runoff-operation"/>
            </owl:allValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-output">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
```

```
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#runoff-volume"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:allValuesFrom rdf:resource="#runoff-volume"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-input">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#input"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#land-cover-value"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#DEM"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#precipitation-rain-fall-value"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:allValuesFrom>
              <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                  <owl:Class rdf:about="#precipitation-rain-fall-value"/>
                  <owl:Class rdf:about="#land-cover-value"/>
                  <owl:Class rdf:about="#DEM"/>
                </owl:unionOf>
              </owl:Class>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-operation">
```

```
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#requested-runoff-input"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#requires"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
            <owl:someValuesFrom rdf:resource="#requested-runoff-output"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

The following ontology describes a sample of requested geo-service which computes runoff rate. This ontology has the "Subsume" match with the ontology describing provided geo-service with closure axiom.

```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:pr="http://www.ncc.org.ir/ontologies/Provided-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.ncc.org.ir/ontologies/Requested-runoff.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Requested-runoff.owl">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/Provided-runoff.owl"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
  </owl:Ontology>
  <owl:Class rdf:ID="precipitation-rain-fall-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Millimeter"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#precipitation-rain-
fall"/>
          </owl:Restriction>
        </owl:intersectionOf>
```

```
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="DEM">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#height"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#Meter"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="land-cover-value">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#land-cover"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-service"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
        </owl:intersectionOf>
```

```
      &lt;/owl:Class&gt;
    &lt;/owl:equivalentClass&gt;
  &lt;/owl:Class&gt;
  &lt;owl:Class rdf:ID="runoff-volume"&gt;
    &lt;owl:equivalentClass&gt;
      &lt;owl:Class&gt;
        &lt;owl:intersectionOf rdf:parseType="Collection"&gt;
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          &lt;owl:Restriction&gt;
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#LitrPerSquareMeter"/>
          &lt;/owl:Restriction&gt;
          &lt;owl:Restriction&gt;
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
          &lt;/owl:Restriction&gt;
        &lt;/owl:intersectionOf&gt;
      &lt;/owl:Class&gt;
    &lt;/owl:equivalentClass&gt;
  &lt;/owl:Class&gt;
  &lt;owl:Class rdf:about="#requested-runoff-service"&gt;
    &lt;owl:equivalentClass&gt;
      &lt;owl:Class&gt;
        &lt;owl:intersectionOf rdf:parseType="Collection"&gt;
          &lt;owl:Restriction&gt;
            &lt;owl:someValuesFrom&gt;
              &lt;owl:Class rdf:ID="requested-runoff-operation"/&gt;
            &lt;/owl:someValuesFrom&gt;
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          &lt;/owl:Restriction&gt;
          &lt;owl:Restriction&gt;
            &lt;owl:allValuesFrom&gt;
              &lt;owl:Class rdf:about="#requested-runoff-operation"/&gt;
            &lt;/owl:allValuesFrom&gt;
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          &lt;/owl:Restriction&gt;
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        &lt;/owl:intersectionOf&gt;
      &lt;/owl:Class&gt;
    &lt;/owl:equivalentClass&gt;
  &lt;/owl:Class&gt;
  &lt;owl:Class rdf:ID="requested-runoff-output"&gt;
    &lt;owl:equivalentClass&gt;
      &lt;owl:Class&gt;
        &lt;owl:intersectionOf rdf:parseType="Collection"&gt;
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
```

```
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#runoff-volume"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:allValuesFrom rdf:resource="#runoff-volume"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-input">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#input"/>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#land-cover-value"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#DEM"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#precipitation-rain-fall-value"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:allValuesFrom>
              <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                  <owl:Class rdf:about="#precipitation-rain-fall-value"/>
                  <owl:Class rdf:about="#land-cover-value"/>
                  <owl:Class rdf:about="#DEM"/>
                </owl:unionOf>
              </owl:Class>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-operation">
    <owl:equivalentClass>
      <owl:Class>
```

```
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#requested-runoff-input"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#requires"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#requested-runoff-output"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

The following ontology describes a sample of requested geo-service which computes runoff rate with no closure axiom. This ontology has the "Plugin" match with the ontology describing provided geo-service.

آنتولوژی درخواست سرویس مکانی محاسبه کننده نرخ سیلاب برای نمایش تطابق "Plugin"

```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:pcr="http://www.ncc.org.ir/ontologies/Provided-Calculate-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.ncc.org.ir/ontologies/Requested-runoff.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Requested-runoff.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/Provided-Calculate-
runoff.owl"/>
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
  </owl:Ontology>
  <owl:Class rdf:ID="requested-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-service"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

```xml
  <owl:Class rdf:ID="runoff-volume">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#LitrPerSquareMeter"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-service">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-operation"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:allValuesFrom>
              <owl:Class rdf:about="#requested-runoff-operation"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-output">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#runoff-volume"/>
```

```xml
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:allValuesFrom rdf:resource="#runoff-volume"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
            <owl:someValuesFrom rdf:resource="#requested-runoff-output"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

The following ontology describes a sample of requested geo-service which computes runoff rate with no closure axiom. This ontology has the "Intersection" match with the ontology describing provided geo-service.

آنتولوژی درخواست سرویس مکانی محاسبه کننده نرخ سیلاب برای نمایش تطابق "Intersection"

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:pcr="http://www.ncc.org.ir/ontologies/Provided-Calculate-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.ncc.org.ir/ontologies/Requested-runoff.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Requested-runoff.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl"/>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/Provided-Calculate-
runoff.owl"/>
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
  </owl:Ontology>
  <owl:Class rdf:ID="runoff-rate">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
          <owl:Restriction>
            <owl:hasValue
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#KiloGramPerSquareMeter
"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#has_m-uni"/>
          </owl:Restriction>
          <owl:Restriction>
```

```
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
            <owl:someValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:allValuesFrom
rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#runoff"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#represents"/>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-service"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="intersection-requested-provided-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
              <rdf:Description rdf:about="http://www.ncc.org.ir/ontologies/Provided-
Calculate-runoff.owl#calculate-runoff-operation"/>
              <owl:Class rdf:ID="requested-runoff-operation"/>
            </owl:intersectionOf>
          </owl:Class>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-service">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
```

```
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:someValuesFrom>
              <owl:Class rdf:about="#requested-runoff-operation"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:allValuesFrom>
              <owl:Class rdf:about="#requested-runoff-operation"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-output">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#runoff-rate"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:allValuesFrom rdf:resource="#runoff-rate"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
            <owl:someValuesFrom rdf:resource="#requested-runoff-output"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

The following ontology has the "Disjoint" match with the ontology describing provided geo-service.


```xml
<?xml version="1.0"?>
<rdf:RDF
    xmlns:pcr="http://www.ncc.org.ir/ontologies/Provided-Calculate-runoff.owl#"
    xmlns:das="http://www.ncc.org.ir/ontologies/DandS.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:cogs="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.ncc.org.ir/ontologies/Requested-runoff.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:mth="http://www.ncc.org.ir/ontologies/MeasureTheory.owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:uont="http://www.ncc.org.ir/ontologies/UpperOnt.owl#"
  xml:base="http://www.ncc.org.ir/ontologies/Requested-runoff.owl">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl"/>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/Provided-Calculate-
runoff.owl"/>
    <owl:imports rdf:resource="http://www.ncc.org.ir/ontologies/MeasureTheory.owl"/>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >classified
    </owl:versionInfo>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100
    </owl:versionInfo>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >ontology of a sample of geo service. OWL engineering by
G.R.Fallahi.</rdfs:comment>
  </owl:Ontology>
  <owl:Class rdf:ID="runoff-rate">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class>
            <owl:complementOf rdf:resource="http://www.ncc.org.ir/ontologies/Provided-
Calculate-runoff.owl#runoff-volume"/>
          </owl:Class>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#field-data"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-rate-output">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
```

```
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
            <owl:someValuesFrom rdf:resource="#runoff-rate"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:allValuesFrom rdf:resource="#runoff-rate"/>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/DandS.owl#played-by"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#output"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="requested-runoff-profile">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#describes"/>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-service"/>
            </owl:someValuesFrom>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#service-profile"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-service">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="requested-runoff-operation"/>
            </owl:someValuesFrom>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#part-by"/>
            <owl:allValuesFrom>
              <owl:Class rdf:about="#requested-runoff-operation"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-service"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
```

```
  </owl:Class>
  <owl:Class rdf:ID="intersection-requested-provided-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#requested-runoff-operation"/>
              <rdf:Description rdf:about="http://www.ncc.org.ir/ontologies/Provided-
Calculate-runoff.owl#calculate-runoff-operation"/>
            </owl:intersectionOf>
          </owl:Class>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#requested-runoff-operation">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#yeilds"/>
            <owl:someValuesFrom rdf:resource="#requested-runoff-rate-output"/>
          </owl:Restriction>
          <rdf:Description
rdf:about="http://www.ncc.org.ir/ontologies/CoreOntGeoService.owl#geo-operation"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
</rdf:RDF>
```

Appendix C: Geo-Service Discovery Program

The application has been written in Java language and developed and used for discovering geo-services in this research. This application has been used OWL model by accessing to packages of Protégé software.

```
/*##################################################
 * ReasonerExamp.java
 *
 * Created on February 26, 2006, 12:08 PM
 *
*/###################################################

package reasoner;

import edu.stanford.smi.protegex.owl.model.OWLModel;
import edu.stanford.smi.protegex.owl.model.OWLNamedClass;
import edu.stanford.smi.protegex.owl.ProtegeOWL;
import edu.stanford.smi.protegex.owl.model.OWLIntersectionClass;
import edu.stanford.smi.protegex.owl.inference.dig.exception.DIGReasonerException;
import edu.stanford.smi.protegex.owl.inference.protegeowl.ReasonerManager;
import edu.stanford.smi.protegex.owl.inference.protegeowl.ProtegeOWLReasoner;
import edu.stanford.smi.protegex.owl.inference.protegeowl.log.ReasonerLogger;
import edu.stanford.smi.protegex.owl.inference.protegeowl.log.ReasonerLoggerListener;
import edu.stanford.smi.protegex.owl.inference.protegeowl.log.ReasonerLogRecord;
import edu.stanford.smi.protegex.owl.inference.protegeowl.log.MessageLogRecord;
import edu.stanford.smi.protegex.owl.inference.dig.reasoner.DIGReasonerIdentity;
import edu.stanford.smi.protegex.owl.model.RDFResource;
```

```
import edu.stanford.smi.protegex.owl.model.RDFSNamedClass;
import edu.stanford.smi.protegex.owl.ui.ProtegeUI;
import org.apache.commons.logging.LogFactory;

import java.awt.*;
import java.awt.event.*;
import java.awt.event.ActionEvent;
import java.io.*;
import java.util.Iterator;
import java.util.Collection;
import java.util.*;
import java.util.Map;

import javax.swing.*;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JRadioButtonMenuItem;
import javax.swing.ButtonGroup;
import javax.swing.JMenuBar;
import javax.swing.KeyStroke;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JScrollPane;
import javax.swing.JFrame;
import java.lang.*;



public class ReasonerExamp implements ActionListener, ItemListener{

    public static JTextArea output;
    public String selectedClassName;
    JScrollPane scrollPane;
    public static String newline = "\n";
    public static ProtegeOWLReasoner reasoner = null;
    public static JFrame frame;
    public static OWLModel model;
    //Collection OWLclassCollection;

    public JMenuBar createMenuBar() {
        JMenuBar menuBar;
        JMenu menu, submenu;
        JMenuItem menuItem;
```

```
JRadioButtonMenuItem rbMenuItem;
JCheckBoxMenuItem cbMenuItem;

//#####Create the menu bar.
menuBar = new JMenuBar();

//#####Build the first menu.
menu = new JMenu("File");
menu.setMnemonic(KeyEvent.VK_F);
menu.getAccessibleContext().setAccessibleDescription(
        "The file menu has menu items");
menuBar.add(menu);

//######a group of JMenuItems
menuItem = new JMenuItem("Open OWL File...",
        KeyEvent.VK_O);
menuItem.setAccelerator(KeyStroke.getKeyStroke(
        KeyEvent.VK_O, ActionEvent.ALT_MASK));
menuItem.getAccessibleContext().setAccessibleDescription(
        "This open an OWL file");
menuItem.addActionListener(this);
menu.add(menuItem);
menu.addSeparator();

//#########a group of JMenuItems
menuItem = new JMenuItem("Perform Reasoning",
        KeyEvent.VK_P);
menuItem.setAccelerator(KeyStroke.getKeyStroke(
        KeyEvent.VK_P, ActionEvent.ALT_MASK));
menuItem.getAccessibleContext().setAccessibleDescription(
        "This perform reasoning");
menuItem.addActionListener(this);
menu.add(menuItem);
menu.addSeparator();

//##########a group of JMenuItems
menuItem = new JMenuItem("Display Concept",
        KeyEvent.VK_D);
menuItem.setAccelerator(KeyStroke.getKeyStroke(
        KeyEvent.VK_D, ActionEvent.ALT_MASK));
menuItem.getAccessibleContext().setAccessibleDescription(
        "This perform reasoning");
menuItem.addActionListener(this);
menu.add(menuItem);
menu.addSeparator();
```

```
    //###########a group of JMenuItems
    menuItem = new JMenuItem("Exit",
            KeyEvent.VK_X);
    menuItem.setAccelerator(KeyStroke.getKeyStroke(
            KeyEvent.VK_X, ActionEvent.ALT_MASK));
    menuItem.getAccessibleContext().setAccessibleDescription(
            "This exit the program");
    menuItem.addActionListener(this);
    menu.add(menuItem);
    menu = new JMenu("Setting");
    menu.setMnemonic(KeyEvent.VK_S);
    menu.getAccessibleContext().setAccessibleDescription(
            "This menu does setting");
    menuBar.add(menu);

    //###############a group of JMenuItems
    menuItem = new JMenuItem("Connect Reasoner",
            KeyEvent.VK_C);
    menuItem.setAccelerator(KeyStroke.getKeyStroke(
            KeyEvent.VK_C, ActionEvent.ALT_MASK));
    menuItem.getAccessibleContext().setAccessibleDescription(
            "This connect to a reasoner");
    menuItem.addActionListener(this);
    menu.add(menuItem);
    return menuBar;
}

public Container createContentPane() {


    //######Create the content-pane-to-be.
    JPanel contentPane = new JPanel(new BorderLayout());
    contentPane.setOpaque(true);

    //#######Create a scrolled text area.
    output = new JTextArea(5, 30);
    output.setEditable(false);
    output.setFont(new Font("Plain", Font.BOLD, 12));
    output.setLineWrap(true);
    output.setWrapStyleWord(true);
    scrollPane = new JScrollPane(output);

    //#########Add the text area to the content pane.
    contentPane.add(scrollPane, BorderLayout.CENTER);
```

```
    return contentPane;
}

public void actionPerformed(ActionEvent e) {

    JMenuItem source = (JMenuItem)(e.getSource());
    if (source.getText().equals("Open OWL File...")){
        output.setEditable(true);
        output.selectAll();
        output.cut();
        output.setCaretPosition(output.getDocument().getLength());
        openOWLFile();
        loadOWLRequestedGeoService();
    } else if (source.getText().equals("Perform Reasoning")){
        performReasoner();
    } else if (source.getText().equals("Exit")){
        System.exit(0);
    } else if (source.getText().equals("Connect Reasoner")){
        connectReasoner();
    }
}

public void itemStateChanged(ItemEvent e) {
    JMenuItem source = (JMenuItem)(e.getSource());
    String s = "Item event detected."
            + newline
            + "    Event source: " + source.getText()
            + " (an instance of " + getClassName(source) + ")"
            + newline
            + "    New state: "
            + ((e.getStateChange() == ItemEvent.SELECTED) ?
                "selected":"unselected");
    output.append(s + newline);
    output.setCaretPosition(output.getDocument().getLength());
}
// Returns just the class name -- no package info.
protected String getClassName(Object o) {
    String classString = o.getClass().getName();
    int dotIndex = classString.lastIndexOf(".");
    return classString.substring(dotIndex+1);
}

private void loadOWLRequestedGeoService() {
    try {
        // ########## Get the OWLNamedClass from the OWLModel
```

```
        Collection OWLclassCollection = model.getUserDefinedOWLNamedClasses();
        if(OWLclassCollection != null) {
                for(Iterator it = OWLclassCollection.iterator(); it.hasNext();) {
                OWLNamedClass curClass = (OWLNamedClass) it.next();
                if (curClass.getName().contains("mth:")==false &&
                        curClass.getName().contains("cogs:")==false &&
                        curClass.getName().contains("das:")==false &&
                        curClass.getName().contains("uont:")==false){
                    output.append(curClass.getName() + newline);
                    output.setCaretPosition(output.getDocument().getLength());
                }
            }
        } else {
            output.append("OWLModel is empty" + newline);
            output.setCaretPosition(output.getDocument().getLength());
        }
    } catch(Exception e1) {
        e1.printStackTrace();
    }
}

private static void openOWLWindow() {
    JFrame.setDefaultLookAndFeelDecorated(true);
    frame = new JFrame("Geo Service Finder");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //########Create and set up the content pane.
    ReasonerExamp demo = new ReasonerExamp();
    frame.setJMenuBar(demo.createMenuBar());
    frame.setContentPane(demo.createContentPane());

    //#######Display the window.
    frame.setSize(450, 260);
    frame.setVisible(true);
}

private static void openOWLFile() {
    UseFileDialog ufd = new UseFileDialog();
    String fileName = ufd.loadFile(new Frame(), "Open OWL file");
    int dot = fileName.lastIndexOf('.');
    int sep = fileName.lastIndexOf('\\');
    System.out.println(fileName);
    frame.setTitle("Geo Service Finder: " + fileName.substring(sep + 1, fileName.length()));
    model=null;
    FileInputStream fis = null;
```

```
    try {
        File file = new File(fileName);
        fis = new FileInputStream(file);
    } catch (FileNotFoundException ex) {
        ex.printStackTrace();
    }
    try {
        model = ProtegeOWL.createJenaOWLModelFromInputStream(fis);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

private static void connectReasoner() {
    output.setEditable(true);
    output.selectAll();
    output.cut();
    output.setCaretPosition(output.getDocument().getLength());
    try {
        final String REASONER_URL = "http://localhost:8080";
        ReasonerManager reasonerManager = ReasonerManager.getInstance();
        if (model != null){
            reasoner = reasonerManager.getReasoner(model);

            // ###########Set the reasoner URL and test the connection
            reasoner.setURL(REASONER_URL);
            if(reasoner.isConnected()) {

                /*=============================================*/
                // Get the reasoner identity - this contains information
                // about the reasoner, such as its name and version,
                // and the tell and ask operations that it supports.
                /*=============================================*/
                DIGReasonerIdentity reasonerIdentity = reasoner.getIdentity();
                System.out.println("Connected to " + reasonerIdentity.getName());
                output.append("Connected to " + reasonerIdentity.getName() + newline);
                output.setCaretPosition(output.getDocument().getLength());
            } else {
                System.out.println("Reasoner not connected!");
                output.append("Reasoner not connected!" + newline);
                output.setCaretPosition(output.getDocument().getLength());
            }
        } else {
            System.out.println("An OWL file must be opened!");
```

```
            output.append("An OWL file must be opened!" + newline);
            output.setCaretPosition(output.getDocument().getLength());
        }
    } catch(Exception e) {
        e.printStackTrace();
    }

    ReasonerLoggerListener lsnr = new ReasonerLoggerListener() {
        public void logRecordPosted(ReasonerLogRecord reasonerLogRecord) {
            if(reasonerLogRecord instanceof MessageLogRecord) {
                MessageLogRecord msgLog = (MessageLogRecord) reasonerLogRecord;
                System.out.println(msgLog.getMessage());
            }
        }
    };
    ReasonerLogger.getInstance().addListener(lsnr);
}

private static void performReasoner() {
    Collection OWLserviceCollection = null;
    int flag;
    OWLNamedClass intersectionClass[];
    output.setEditable(true);
    output.selectAll();
    output.cut();
    output.setCaretPosition(output.getDocument().getLength());
    try {
        // ###############Get OWLNamedClass from the OWLModel
        OWLserviceCollection = model.getUserDefinedOWLNamedClasses();
        if(OWLserviceCollection != null) {
            for(Iterator its = OWLserviceCollection.iterator(); its.hasNext();) {
                OWLNamedClass curClass = (OWLNamedClass) its.next();
                flag = 0;
                if(curClass != null){
                    if(curClass.getName().contains(":")==false){
                        // Now get the inferred equivalent classes
                        Collection inferredEquiclasses = reasoner.getEquivalentClasses(curClass, null);
                        if(inferredEquiclasses.size() > 1) {
                            for(Iterator it = inferredEquiclasses.iterator(); it.hasNext();) {
                                OWLNamedClass curClassE = (OWLNamedClass) it.next();
                                if (curClass.equals((OWLNamedClass) curClassE)==false &&
                                        curClassE.getName().contains("mth:")==false &&
                                        curClassE.getName().contains("cogs:")==false &&
                                        curClassE.getName().contains("das:")==false &&
                                        curClassE.getName().contains("uont:")==false){
```

```
            output.append(curClass.getName() + " Exact Match with " + curClassE.getName()
                                                                    + newline);
            output.setCaretPosition(output.getDocument().getLength());
        }
    }
    flag = 1;
}
if(flag==0){
    Collection inferredSubclasses = reasoner.getSubclasses(curClass, null);
    if(inferredSubclasses.size() > 0) {
        for(Iterator it = inferredSubclasses.iterator(); it.hasNext();) {
            OWLNamedClass curClassSb = (OWLNamedClass) it.next();
            if (curClass.equals((OWLNamedClass) curClassSb)==false &&
                    curClassSb.getName().contains("mth:")==false &&
                    curClassSb.getName().contains("cogs:")==false &&
                    curClassSb.getName().contains("das:")==false &&
                    curClassSb.getName().contains("uont:")==false) {
                output.append(curClass.getName() + " Subsumes " + curClassSb.getName() +
                                                                  newline);
                output.setCaretPosition(output.getDocument().getLength());
            }
        }
        flag = 1;
    }
}
if(flag==0){
    Collection inferredSuperclasses = reasoner.getSuperclasses(curClass, null);
    if(inferredSuperclasses.size() > 0) {
        for(Iterator it = inferredSuperclasses.iterator(); it.hasNext();) {
            OWLNamedClass curClassSp = (OWLNamedClass) it.next();
            if(curClass.equals((OWLNamedClass) curClassSp)==false &&
                    curClassSp.getName().contains("mth:")==false &&
                    curClassSp.getName().contains("cogs:")==false &&
                    curClassSp.getName().contains("das:")==false &&
                    curClassSp.getName().contains("uont:")==false){
                output.append(curClass.getName() + " Plugins " + curClassSp.getName() +
                                                                newline);
                output.setCaretPosition(output.getDocument().getLength());
                flag = 1;
            }
        }
    }
}

if(flag==0){
```

```
                    intersectionClass = new OWLNamedClass[2];
                    Collection inferredSuperclasses = reasoner.getSuperclasses(curClass,null);
                    for(Iterator itp = inferredSuperclasses.iterator(); itp.hasNext();) {
                        OWLNamedClass curClassSp = (OWLNamedClass) itp.next();
                        Collection inferredSubclasses = curClassSp.getNamedSubclasses();
                        for(Iterator itb = inferredSubclasses.iterator(); itb.hasNext();) {
                            OWLNamedClass curClassSb = (OWLNamedClass) itb.next();
                            if (curClass.equals((OWLNamedClass) curClassSb)==false &&
                                    curClassSb.getName().contains("mth:")==false &&
                                    curClassSb.getName().contains("cogs:")==false &&
                                    curClassSb.getName().contains("das:")==false &&
                                    curClassSb.getName().contains("uont:")==false){
                                intersectionClass[0] = curClass;
                                intersectionClass[1] = curClassSb;
                                //intersectionClass.addOperand(curClass);
                                //intersectionClass.addOperand(curClassSb);
                                boolean result = reasoner.isIntersectionSatisfiable(intersectionClass,null);
                                if (result == true){
                                    output.append(curClass.getName() + " overlap " + curClassSb.getName() +
                                                                                newline);
                                    output.setCaretPosition(output.getDocument().getLength());
                                    //flag = 1;
                                }else{
                                    output.append(curClass.getName() + " disjoint " + curClassSb.getName() +
                                                                                newline);
                                    output.setCaretPosition(output.getDocument().getLength());
                                }
                                flag = 1;
                            }
                        }
                    }
                }
            }
        }
    }
  } else {
      output.append("OWLModel is empty" + newline);
      output.setCaretPosition(output.getDocument().getLength());
  }
 } catch(Exception e1) {
     e1.printStackTrace();
 }
}
```

```
    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                openOWLWindow();
            }
        });
    }
}
```

References

[ARION, 2003], Advanced Lightweight Architecture for Accessing Scientific Collections, EU FP5 project, IST Project No 2000-25289, Copyright 2003 The ARION Consortium.

[Alameh, 2003], N. Alameh, SEPTEMBER • OCTOBER 2003 Published by the IEEE Computer Society, 1089-7801/03/$17.00©2003 IEEE, IEEE INTERNET COMPUTING

[ArcGIS, 2006], Analysis operators and the output attribute table, http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=3040&pid=3035&topicname=Analysis%20operators%20and%20the%20output%20attribute%20table, last modified 15 November 2006

[Baader et al., 2002] Edited by F. Baader, D. McGuinness, D. Nardi, and P. F. Patel-Schneider (2002), Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press.

[Bechhofer, 2003] Bechhofer, S., (2003), the DIG Description Logic Interface: DIG/1.1, In Proceedings of DL2003 Workshop, Rome, Italy.

[Bechhofer and Horrocks, 2003], Sean Bechhofer and Ian Horrocks, (2003) "The WonderWeb Ontology Language Layer ", WonderWeb Deliverable Del 1, http://wonderweb.semanticweb.org/deliverables/D1.shtml

[Bechhofer et al. 2001] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens, (2001), OilEd: a reason-able ontology editor for the semantic web. In Proceedings of the 2001 Description Logic Workshop (DL 2001), pages 1–9. CEUR (http://ceur-ws.org/).

[Benjamins and Fensel, 1998] Benjamins, V. R. and Fensel, D. (1998), The ontological engineering initiative (KA)2, In Guarino, N., editor, Formal Ontology in Information Systems, pages 287–301, IOS Press, Trento-Italy.

[Bergamaschi and Nebel, 1994], Bergamaschi S.and Nebel B., (1994), Acquisition and validation of complex object database schemata supporting multiple inheritance, Applied intelligence, 4(2), 185-203.

[Bernard and Krüger, 2000] Lars Bernard, Thomas Krüger, 2000, Integration of GIS and Spatio-temporal Simulation Models:Interoperable Components for Different Simulation Strategies, "(3+1)D-GIS" project, funded by the German Science Foundation grant no. STR 172/8-1

[Bernard et al., 1998] Bernard L, Schmidt B, and Streit U 1998, AtmoGIS - Integration of Atmospheric Models and GIS, In Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, International Geographical Union: 267-276.

[Berners-Lee, 1998] T. Berners-Lee, (1998), Semantic web road map, http://www.w3.org/DesignIssues/Semantic.html

[Berners-Lee et al. 2001] T. Berners-Lee, J. Hendler, & O. Lassila, (2001), "The semantic web", Scientific American ,284(5).

[Bishr, 1998], Yaser Bishr, 1998, "Overcoming the semantic and other barriers to GIS interoperability", int. j. geographical information science, 12(4), 299-314

[Blackburn and Seligman, 1995] Blackburn, P. and Seligman, J., (1995), Hybrid languages. J. of Logic, Language and Information 4, 251–272.

[Borgo et al., 2005] Stefano Borgo, Nicola Guarino, Laure Vieu, "Formal Ontology for Semanticists", Research Institute for Computer Science of Toulouse – CNRS, Laboratory for applied ontology, www.loa-cnr.it

[Borgida, 1996] Borgida, A. (1996), On the relative expressiveness of description logics and predicate logics, Artificial Intelligence, 82(1-2):353–367.

[Brachman and Levesque, 1984] Ronald J. Brachman and Hector J. Levesque, (1984), "The Tractability of Subsumption in Frame-Based Description Languages", In the Proceedings of the 4th National Conference of the American Association for Artificial Intelligence, pages 34-37, Austin, TX.

[Brodie, 1992], Brodie, M. L., 1992, "The promise of distributed computing and the challenge of legacy lnformation systems", In Proceedings of the IFIP WG2.6 Database Semantics Conference on Interoperable Database Systems (DS-5), (Amsterdam: North Holland), 1-31

[Brickley et al, 2000] D. Brickley and R. V. Guha (editors), Resource description framework (rdf) schema speci¯cation 1.0, http://www.w3.org/TR/2000/CR-rdf-schema-20000327/, March, 2000.

[Bruns and Egenhofer, 1997] H. Thomas Bruns, Max Egenhofer, 1997, User Interfaces for Map Algebra, Journal of the Urban and Regional Information Systems Association Vol.9, No.1, pp. 44-54.

[Buehler and McKee, 1996] Buehler, K. and McKee, L. 1996, The OpenGIS Guide - Introduction to Interoperable Geoprocessing, Technical report, Open Geodata Interoperability Specification (OGIS), Open GIS Consortium Inc.

[Burners-Lee et al. 2001] Burners-Lee, T., Hendler, J. & Lassila, O. (2001), "The semantic web", Scientific American ,284(5).

[Capeclear, 2005] Cape Clear Software Inc., 2005, "An introduction to service-oriented architecture", Available at: http://www.capeclear.com/technology/architecture/. Accessed 4 Jan 2005

[Casti, 1989], Casti, John L, 1989, Alternate Realities: Mathematical Models of Nature and Man, John Wiley & Sons

[CCSP, 2003] "Strategic Plan for the Climate Change Science Program", Final Report, July 2003, http://www.climatescience.gov/Library/stratplan2003/final/annexc-figure5-1.htm

[Cookbook, 2003] Editor: Kris Kolodziej, 2003, "OpenGIS Web Map Server Cookbook", April 28, 2003, OGC Document Number: 03-050, Version: 1.0.0

[CO-ODE-R], Co-ode, Protégé OWL Guides, http://www.co-ode.org/resources/reference/, last visit: 28/07/2007

[Cornet 2003] R. Cornet (2003), Rice ontology editor, Homepage at http://www.b1g-systems.com/ronald/rice/, last visit: 28/07/2007

[Couclelis, 1992] Couclelis, H. 1992, "People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS", Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, A. U. Frank, I. Campari and U. Formentini, Springer-Verlag. 639: 65-77

[Dangermond, 2004] Jack Dangermond, Fall 2004, “Speaking the Language of Geography-GIS”, ArcNews, Vol. 26 No.3 pp. 1-7.

[Davis, 1990] Ernest David, 1990 “Representations of Commonsense Knowledge” by Morgan Kaufmann Publishers, Inc. ISBN 1-55860-033-7.

[Davies et al, 2004] Davies N.J., Fencel D., and Richardson M., 2004, "The future of web services", BT Technology Journal, 22(1), Jan 2004.

[Decker et al., 2000] Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Broekstra, J., Erdman, M., and Horrocks, I. (2000). The semantic web: The role of XML and RDF. IEEE Internet Computing.

[Decker et al., 1999] Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1999), ONTOBRKER: Ontology based access to distributed and semi-structured information. In Meersman, R., Tari, Z., and Scott, editors, Semantic Issues in Multimedia System, Proceedings of Eighth Working Conference on Database Semantics (DS-8), IFIP

TC2/WG2.6, pages 351–369. Kluwer Academic Publisher. ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/rdf3.ps.

[Decker et al., 1998] Decker, S., Brickley, D., Saarela, J., and Angele, J. (1998),A query and interface service for rdf, http://www.w3.org/TandS/QL/QL98/pp/queryservice.html.

[Djokic, 1996] D. Djokic 1996, “Toward a general-purpose decision support system using existing technologies”. In GIS and Environmental Modeling: Progress and Research Issues, (Fort Collins GISWorld Books), 353-356

[Drummond, 2005] Compiled by Nick Drummond, (2005), A Practical Introduction to Ontologies & OWL, The University of Manchester, www.co-ode.org/resources/tutorials/intro/, last visit: 28/07/2007

[Egenhofer, 2002] Egenhofer, M., 2002, Toward the semantic geospatial web, Tenth ACM International Symposium on Advances in Geographic Information Systems. ACM Press, New York, NY, USA, McLean, Virginia, USA, pp. 1-4.

[ESRI, 2004], What is ArcGIS? Copyright © 2001–2004 ESRI, All rights reserved, Printed in the United States of America.

[Estrem, 2003] ESTREM, W. A. An evaluation framework for deploying web services in the next generation manufacturing enterprise. Robotics and Computer Integrated Manufacturing 19 (December 2003), 509-519.

[Fabrikant et.al., 2001] Fabrikant, S.I. and Buttenfield, B.P., 2001, Formalizing Semantic Spaces for Information Access. Annals of the Association of American Geographers, 91(2): 263-280.

[Farrar and Bateman, 2005], Scott Farrar and John Bateman (2005), "General Ontology Baseline", Deliverable D1, I1-[OntoSpace], Workpackage 1, http://www.sfbtr8.uni-bremen.de/I1

[Fedra 1996] Fedra K 1996, “Distributed Models and Embedded GIS: Integration Strategies and Case Studies”. In M F Goodchild, L T Steyaertand B O Parks (eds) GIS and Environmental Modeling: Progress and Research Issues, (Fort Collins, GIS World Books), 413-418

[Fedra, 1993] Fedra, 1993, GIS and environmental modeling, in M.F. Goodchild, B.O. Parks and L.T. Steyaert, eds, Environmental Modeling and GIS, Oxford University Press, Chapter 5.

[Feng, 2005], Chen-Chieh Feng, 2005, Identification of Semantic Barriers for Exchanging Concepts between Surface Hydrology Domain and Spatial Domain, Annual meeting of AAG, Denver, CO

[Feng, 2000], Chen-Chieh Feng, Open Hydrologic Model for Facilitating GIS and Hydrologic Model Interoperability, 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs, Banff, Alberta, Canada, September 2 - 8, 2000

[Feng et al, 2004],Chen-Chieh Feng, Thomas Bittner, Douglas M. Flewelling: Modeling Surface Hydrology Concepts with Endurance and Perdurance. GIScience 2004: 67-80

[Feng and Sorokine, 2001], Chen-Chieh Feng, Alexandre Sorokine: Incorporating hydrologic semantic information for interoperable GIS with hydrologic model. ACM-GIS 2001: 59-63

[Fensel, 2001] Fensel, D. (2001), Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag.

[Fensel et al., 1999] Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.-P., Staab, S., Studer, R., and Witt, A. (1999), On2Broker: Semantic-based access to information sources at the WWW, In Proceedings of the World Conference on the WWW and Internet (WebNet 99), pages 366–371. ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/webnet.pdf.

[Fischer, 1994] Fischer, M. M. 1994, From conventional to knowledge-based geographic information systems, Computing, Environment and Urban Systems 18: 233-242.

[Frank, 2005] Andrew U. Frank, “PRACTICAL GEOMETERY” “THE MATHEMATICS FOR GEOGRAPHIC INFORMATION SYSTEMS”, Draft manuscript, March 2005 v14.

[Frank and Kuhn, 1998] Frank, A. and Kuhn, W. (1998), A specification language for interoperable GIS, In Goodchild, M. F., Egenhofer, M. J., Fegeas, R., and Kottman, K., editors, Interoperating Geographic Information Systems, pages 123–132. Kluwer.

[Gangemi and Mika 2003] Aldo Gangemi and Peter Mika (2003), "Understanding the semantic web through descriptions and situations", In DOA/CoopIS/ODBASE, 2003 Confederated International Conferences DOA, CoopIS and ODBASE, Proceedings, LNCS, Springer.

[Galton, 2001] Galton, A. C. 2001, "Space, time, and the representation of geographical reality" Topoi, 20, 173-187

[Gärdenfors, 2000], Peter Gärdenfors, 2000, Conceptual Spaces: The Geometry of Thought, A Bradford Book, MIT Press, Cambridge (Massachusetts), 2000.

[Genesereth and Fikes, 1992] Genesereth, M. R., & Fikes, R. E. (1992), Knowledge Interchange Format, Version 3.0 Reference Manual, Technical Report Logic-92-1, Computer Science Department, Stanford University.

[Genesereth and Nilsson 1987] Genesereth M. R. and N. J. Nilsson (1987), Logical Foundations of Artificial Inteligence, Morgan Kaufmann Publishers.

[GEOLEM, 2006], Geospatial Object Library for Environmental Modeling, http://www.iscmem.org/WG01_Geolem.htm (last view, 29 Dec. 2006)

[GFO, 2006] Herre Heinrich Herre, Barbara Heller, Patryk Burek, Robert Hoehndorf, Frank Loebe and Hannes Michalek (2006), "General Formal Ontology (GFO) Part I: Basic Principles", Version 1.0, No. 8, http://www.onto-med.de/.

[Gibson, 1986] Gibson J J, 1986, the Ecological Approach to Visual Perception (Lawrence Erlbaum Associates, Hillsdale, NJ)

[Gijsbers, 2003], Peter Gijsbers, 2003, Review OpenMI architecture v.0.6, http://www.water21.re.kr/nimage/down/review_openmi_v06.pdf

[Goodchild, 2000], Michael F. Goodchild, 2000" SPATIAL ANALYSIS and GIS" ESRI USER CONFERENCE, Pre-Conference Seminar

[Goodchild et. al., 1996] Goodchild M F, Steyaert L T, and Parks B O 1996, GIS and Environmental Modeling: Progress and Research Issues. Fort Collins, GIS World Books.

[Groot, 1997] Richard Groot, (1997), Spatial data infrastructure (SDI) for sustainable land management, ITC Journal, 1997-3/4.

[Gruber, 1992] Gruber, T. R. (1992), Ontolingua: A mechanism to support portable ontologies, Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory. Revision.

[Gruber, 1993] Thomas R. Gruber, 1993, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.

[Gruber and Olsen, 1994], Thomas R. Gruber and Gregory R. Olsen, 1994, An Ontology for Engineering Mathematics, In Jon Doyle, Piero Torasso, & Erik Sandewall, Eds., Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institut, Bonn, Germany, Morgan Kaufmann.

[Gruber, 1993] Thomas R. Gruber, 1993, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University.

[Guarino, 1997] Guarino N., (1997) Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In M. T. Pazienza (ed.) Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. Springer Verlag: 139-170.

[Guarino and Welty, 2002] Guarino, N. and Welty, C. (2002), Evaluating ontological decisions with Onto-Clean, Communications of the ACM, 45(2):61–65.

[Haarslev and Möller, 2003] Haarslev, V., and Möller, R., (2003), Racer: An OWL Reasoning Agent for the SemanticWeb, In Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with the 2003 IEEE/WIC International Conference on Web Intelligence, Halifax, Canada. (October 2003), p. 9195.

[Hakimpour, 2003] Hakimpour, Farshad (2003), "Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata", Dissertation zur Erlangung der naturwissenschaftlichen Doktorwürde (Dr. sc. nat.).

[Harvey, 2003a], H. Harvey, 2003, "some thoughts on modeling tools of the future" http://www.floodrisknet.org.uk/newsletters/2003-1/future_modelling/metadata_rdf

[Harvey, 2003b] Harvey 2003, "A Generic Modeling Framework Component for Hydroinformatics Systems.", Ph.D. Thesis, University of Bristol http://www.cen.bris.ac.uk/pgra/dph/publications.html.

[Harvey and Han, 2002] Harvey and Han, 2002, The relevance of Open Source to Hydroinformatics. J. Hydroinformatics 4(4):219-234.

[Harvey et al, 2004] H. Harvey, Jim Hall, Jonathan Evans, 2004, "Defining Models Using The Resource Description Framework", 6th International Conference on Hydroinformatics, (Liong, Phoon & Babovic (eds) © 2004 World Scientific Publishing Company), ISBN 981-238-787-0

[Harvey et al, 2005] H. Harvey, J. Hall, P.D. Bates, G. Pender, N.G. Wright and D. Mason, 2005, Next Generation Computer Modeling for the Prediction of Flood Level and Inundation Extent: Part 3 Modeling Framework, Technical report.

[Hitchcock, et. al., 1996] Pundt H, Hitchcock A, Bluhm M, and Streit U 1996, A GIS supported Freshwater Information System including a Pen-Computer Component for digital field data recording, In Proceedings of the HydroGIS 96 - Application of GIS in Hydrology and Water Resources Management. Vienna, IAHS: 703-711.

[Horridge et al., 2004] Horridge, M., Knublauch, H., Rector, A., Stevens, R., andWroe, C (2004), A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0.

[Horrocks and Sattler, 2001] Horrocks, I. and Sattler, U., (2001), Ontology reasoning in the SHOQ(D) description logic. In: Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001). (2001)199–204

[Horrocks et al., 2000] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta, (2000), The ontology inference layer OIL. http://www.cs.vu.nl/ dieter/oil/Tr/oil.pdf.

[Horrocks et al., 1999] I. Horrocks, U. Sattler, and S. Tobies , (1999), Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99), number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer-Verlag.

[Hudak, 2000] Hudak, P. (2000), The Haskell School of Expression: Learning Functional Programming through Multimedia. Cambridge University Press.

[Hutchings et al, 2002] Hutchings C., Struve J., Westen S., Millard K., Fortune D., 2002, "State of the Art Review", Work Package 1, IT Frameworks (HarmonIT), http://www.harmonit.org/docs/repu_01_09_soa_review_approved.pdf.

[IBM's tutorial] IBM's tutorial www-4.ibm.com/software/solutions/webservices/, last visit: 28/07/2007

[ISO, 2001], ISO 19119, "Geographic Information –Services" http://www.isotc211.org, last visit: 28/07/2007.

[Jeffers, 1982], Jeffers, J.N.R., 1982, Modeling, Chapman and Hall, New York

[Karp et al., 1999] Karp, P. D., Chaudhri, V. K., and Thomere, J. (1999), XOL: An XML-based ontology exchange language, http://xml.coverpages.org/xol-03.html.

[Kemp, 1993] Kemp, K. K. (1993). Environmental Modeling with GIS: A strategy for dealing with spatial continuity, National Center for Geographic Information and Analysis, Department of Geography, University of California, Santa Barbara.

[Knublauch et al., 2004] Knublauch, H., Fergerson, R. W., Noy, N. F., and Musen, M. A. (2004), The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications, In Proceedings Third International Semantic Web Conference, Hiroshima, Japan.

[Kuhn, 2005] Werner Kuhn, 2005, "Geospatial Semantics: Why, of What, and How?" In: Journal on Data Semantics III, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, ISBN: 978-3-540-26225-1, Computer Science, Volume 3534/2005, ISSN: 0302-9743, pp. 1-24.

[Kuhn, 2002] Kuhn, W., 2002, Modeling the Semantics of Geographic Categories through Conceptual Integration, In: M.J. Egenhofer and D.M. Mark (Editors), GIScience 2002. Lecture Notes in Computer Science. Springer-Verlag.

[Kuhn, 2001] Kuhn, W. (2001), Ontologies in support of activities in geographic space, International Journal of Geographic Information Science, 15(7):613–631.

[Lam, 1983] Lam, N.S-N., 1983. Spatial interpolation methods: a review, The American Cartographer, 10(2):129-149.

[Lassila et al, 1999] O. Lassila and R. R. Swick (editors), Resource description framework (rdf) model and syntax specification, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, Feb, 1999.

[Lenat, 1995] Lenat, D. B. (1995), CYC: A large-scale investment in knowledge infrastructure, Communications of the ACM, 38(11):33–38. http://www.cs.umbc.edu/471/papers/cyc95.pdf.

[Li and Horrocks 2003] Lei Li and Ian Horrocks, (2003), "A Software Framework For Matchmaking Based on Semantic Web Technology", WWW 2003, May 20–24, Budapest, Hungary, ACM 1581136803/03/0005.xxx.

[Mark and Smith, 2001], David M. Mark and Barry Smith (2001), "A Science of Topography: Bridging the Qualitative-Quantitative Divide", supported by the National Science Foundation, Geography and Regional Science program, under Grant No. BCS-9975557

[Masolo et al, 2003] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, Stefano Borgo, "Ontology Infrastructure for the Semantic Web", WonderWeb Deliverable D18, IST WonderWeb Project 2001-33052

[McGuinness and Harmelen, 2004], McGuinness D. L., van Harmelen F., 2004, OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[McIlraith et. al., 2001], Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng, Stanford University 2001, "Semantic Web Services", IEEE Intelligent Systems, vol. 16, no. 2, Mar./Apr. 2001, pp 46-53

[Microsoft, 2003], Microsoft Corporation, December 2003, "Interoperability Technologies: Point to Point", MSDN Library, ms-help://MS.MSDNQTR.2004APR.1033/dnpag/html/jdni_ch04.htm

[Mika et al, 2004], Peter Mika, Daniel Oberle, Aldo Gangemi, Marta Sabou, "Foundations for Service Ontologies: Aligning OWL-S to DOLCE", WWW2004, May 17–22, 2004, New York, NY USA. ACM 1-58113-844-X/04/0005.

[Moore et al., 1993] Moore, I.D., A.K. Turner, J.P. Wilson, S.K. Jenson, L.E. Band, (1993), GIS and land surface subsurface modeling, in M.F. Goodchild, B.O. Parks and L.T. Steyaert, eds, Environmental Modeling and GIS, Oxford University Press, Chapter 19.

[Natis, 2003] Natis, Y. V., 2003, "Service-oriented architecture scenario" Gartner, Inc.

[Navratil 2002] Navratil, G., (2002), "Formalisierung von Gesetzen", Ph. D. thesis, Institute for Geoinformation, TU Vienna, Austria

[NCGIA, 1998] Mitasova H. and Mitas L. (1998) Process Modeling and Simulations, NCGIA Core Curriculum in GIScience, http://www.ncgia.ucsb.edu/giscc/units/u130/u130.html, posted December 2, 1998.

[Newcomer, 2002] Eric Newcomer, 2002, "Understanding Web Services- XML, WSDL, SOAP and UDDI", Copyright © 2002, Boston: Addison-Wesley,

[Nichols and Terry, 2003] Deborah Nichols and Allan Terry (2003), "User's Guide to Teknowledge Ontologies", Teknowledge Corp, Sponsored by the Air Force Research Laboratory under contract F30602-02-C-0045.

[Nyerges, 1992] Nyerges T L 1992, Coupling GIS and Spatial Analytic Models, In Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, USA, IGU Comission on GIS: 524-533.

[OGC, 2003] Open GIS Consortium, Inc. (OGC) 2003, Observations and Measurements (Version: 0.9.2), OpenGIS® Discussion Paper, Reference number: OGC 03-022r3, URL: http://www.opengis.org/, last visit: 28/07/2007

[OGC, 2000], Open GIS Consortium, Inc. (OGC) 2000, OpenGIS Web Map Server Interface Implementation Specification (Revision 1.0.0). Wayland, Massachusetts: Open GIS Consortium, Inc., URL: http://www.opengis.org/techno/specs.htm, last visit: 28/07/2007

[Ogden et al, 1923] Ogden, C. and Richards, I. 1923, "The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism", (London : Routledge & Kegan Paul Ltd., 10 edition)

[Ontologua], Stanford KSL Network Services, http://ontolingua.stanford.edu, last visit: 28/07/2007

[OWL, 2004a] OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-owl-features-20040210/, last visit: 28/07/2007

[OWL, 2004b], OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-owl-semantics-20040210/, last visit: 28/07/2007

[OWL1.1], OWL 1.1 Web Ontology Language, Model-Theoretic Semantics, http://www.webont.org/owl/1.1/semantics.html#ref-owl-1.1-specification, last visit: 28/07/2007

[OWL-S, 2004], OWL-S: Semantic Markup for Web Services, http://www.daml.org/services/owl-s/1.1/overview/, last visit: 28/07/2007

[OWLViz, 2004], Co-code, http://www.co-ode.org/downloads/owlviz/OWLVizGuide.pdf, last visit: 28/07/2007

[Paolucci et al. 2002] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, (2002), "Semantic matching of web services capabilities", In Proceeding of the 1st International Semantic Web Conference (ISWC)

[Peuquet et al, 1999] Peuquet, D. J., Smith, B., & Brogaard, B. 1999, "The ontology of fields (Report of a Specialist Meeting, National Center for Geographic Information and Analysis)" Retrieved from: http://www.ncgia.ucsb.edu/Publications/Varenius_Reports/Ontology_of_Fields.pdf.

[Probst, and Lutz, 2004] Florian Probst, and Michael Lutz, (2004), Giving Meaning to GI Web Service Descriptions, supported by :the European Commission through the ACE-GIS project (grant number IST-2002-37724) and the German Fed-eral Ministry for Education and Research as part of the GEOTECHNOLOGIEN program (grant number 03F0369A), publication no.: GEOTECH-52

[Protégé 2003] Protégé, (2003), Homepage at, http://protege.stanford.edu/, last visit: 28/07/2007

[Pundt et al., 1996] Pundt H, Hitchcock A, Bluhm M, and Streit U 1996, A GIS supported Freshwater Information System including a Pen-Computer Component for digital field data recording, In Proceedings of the HydroGIS 96 - Application of GIS in Hydrology and Water Resources Management, Vienna,IAHS: 703-711.

[Racer, 2005a] Racer, (2005), RacerPro Reference Manual Version 1.8., Racer Systems GmbH & Co. KG, April.

[Racer, 2005b] Racer, (2005), RacerPro User’s Guide Version 1.8. Racer Systems GmbH & Co. KG, April.

[Rajabifard and Williamson, 2001] Abbas Rajabifard, Ian P. Williamson, (2001), Spatial Data Infrastructure: Concept, SDI Hierarchy AND Future Direction, Geomatics’80 Conference, Tehran, Iran.

[Smith, 2002], Barry Smith, Ontology and Information Systems, the National Science Foundation under Grant No.: BCS-9975557 (“Ontology and Geographic Categories”), http://ontology.buffalo.edu/ontology(PIC).pdf

[Smith and Mark, 2003], Barry Smith and David M Mark 2003, "Do mountains exist? Towards an ontology of landforms"' Environment & Planning B (Planning and Design), 30, 411-427.

[Sowa, 2000] Sowa, J. F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole - Thomson Learning.

[Stevens 1946] S.S. Stevens, (1946), "On the theory of scales of measurement", Science, 103(2684):677-680.

[Sure et al. 2002] Y. Sure, S. Staab, and J. Angele, (2002), OntoEdit: Guiding ontology development by methodology and inferencing, In Proceedings of the Confederated International Conferences DOA, CoopIS and ODBASE 2002, vol. 2519 of Lecture Notes in Computer Science. Springer-Verlag, 2002.

[Sydelko, et al, 2000] Pamela J. Sydelko, Jayne E. Dolph, Kimberly A. Majerus, Thomas N. Taxon, 2000, “An advanced object-based software framework for complex ecosystem modeling and simulation”, 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs, Banff, Alberta, Canada, September 2 - 8, 2000.

[Tessaris, 2001] S. Tessaris, (2001), Questions and answers: reasoning and querying in description logic, PhD thesis, University of Manchester, 2001.

[Tomlin, 1990] Tomlin C. D. 1990, "Geographic Information Systems and Cartographic Modeling", (Englewood Cliff, NJ: Prentice-Hall).

[Uschold and Grüninger, 1996] Uschold, M. and Grüninger, M. 1996, Ontologies: Principles, Methods and Applications, The Knowledge Engineering Review, (in press).

[Uschold and King, 1995] Uschold, M. and King, M. (1995), Towards a methodology for building ontologies, In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95, ftp://ftp.aiai.ed.ac.uk/pub/documents/1995/95-ontijcai95-ont-method.ps.gz.

[Uschold et al, 1995] M. Uschold, M. King, S. Moralee, and Y. Zorgios (1995), The enterprise ontology, Enterprise Project Deliverable: MID 3.1, Version 1.1.

[Visser et. al., 2002] Visser, U., Stuckenschmidt, H., Schuster, G. and Vogele, T., 2002, Ontologies for geographic information processing. Computers & Geosciences, 28: 103-117.

[W3] OWL Web Ontology Language, Overview, http://www.w3.org/TR/owl-features/, last visit: 28/07/2007

[W3C, 2004] Web Services Architecture, http://www.w3.org/TR/ws-arch/, last visit: 28/07/2007

[Waris, 2005], Mikko Waris, (2005), Adding Semantics into WSDL and UDDI, http://www.cs.helsinki.fi/u/chande/courses/cs/MWS/seminar/Articles/9.pdf

[Welty and Guarino, 2001] Welty, C. and Guarino, N. (2001). Supporting ontological analysis of taxonomic relationships, Data and Knowledge Engineering, 39(1):51–74.

[Wessel and Möller, 2005] Wessel, M., and Möller, R., (2005), A High Performance Semantic Web Query Answering Engine, In Proceedings International Workshop on Description Logics (2005), I. Horrocks, U. Sattler, and F. Wolter, Eds.

[Worboys and Duckham, 2004], M. F. Worboys and M. Duckham, 2004, "GIS: A Computing Perspective", Boca Raton, FL: CRC Press, 2nd edition, ISBN 0-415-28375-2 (alk. Paper)

[XML], XML in 10 points, http://www.w3.org/XML/1999/XML-in-10-points.html, last visit: 28/07/2007