Semantic Integration of Geodata with Feature Type Hierarchies

Master Thesis in Computer Science

Bjørn Håkon Horpestad

October 13, 2005
Halden, Norway

Høgskolen i Østfold
Avdeling for Informasjonsteknologi

Abstract

This paper deals with semantic integration of geospatial data, or integration of geodata based on their metadata and attributes. As the data can come from a number of sources one of the main challenges is to manage semantic heterogeneity. Ontologies has emerged as a tool or framework to help in a semantic integration process. As a result ontology development has undergone much research, but as of today no standard methodology exist. This thesis uses scenarios to identify methods and frameworks that can be used in ontology building. By using two existing classifications in three scenarios, different types of semantic heterogeneity are enlightened. To establish the intended usage, scope and level of detail are important steps on the road to determine which existing classifications to use. An approach using existing classifications can provide good simple ontologies, or a starting point for further development towards more complex ontologies.

Acknowledgement

First of all I would like to thank my supervisor Gunnar Misund. He has contributed with numerous suggestions and ideas and been a great support throughout the thesis.

I would like to thank my class mates on the computer lab, Kristian, Mats, Hilde and Linda, for exchanging experiences, frustration and happiness. Your support have been invaluable, and the semester would have been a mountain to climb without you.

Finally, I would like to thank all the persons that in some way or another has helped me during this thesis, and I would especially thank those who have helped with proof-reading the thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Semantic integration of geospatial data is integration of attributes from different geographical sources. The integration can be a cumbersome and comprehensive process, with no simple or correct solution. One of the main challenges with semantic integration is to manage semantic differences, or semantic heterogeneity. To make the integration easier and avoid semantic heterogeneity, some methods and techniques can be utilized.

The purpose of geographic data set integration is to share information between different geographic information sources. There are vast amounts of geospatial data available, but one problem is to merge these data so that they can function in one system. OneMap [64] is a project which main purpose is to provide access to world wide multiscale database of geodata, where data will be provided from several different contibutors ranging from mapping agencies to individual users. This concept is shown in figure 1.1. Geospatial data consists of two parts, on one side is spatial data, on the other side is the metadata or data about the spatial data. To achieve full integration, two different, but equally important aspects have to be addressed. Geometrical integration deals with the merging of spatial segments from different data sets. This type of integration is often referred to as conflation or map conflation [75], [89], [15]. The other important aspect is semantic integration which deals with integration of data sets based on their metadata. An increment approach, like that taken by OneMap, requires strategies on how to handle the contributed data. For the OneMap project it is a goal to keep the contributed data as close as possible to the original, this is solved by the use of encapsulation. Two strategies can be identified by this approach. Firstly there is no loss of geometric and semantic precision. Secondly time and labour are saved since no conversion of the data is needed. The work done in [65] only covers syntactical encapsulation in relation to GML. To make this approach fully work and achieve semantic integration, some kind of classification is needed.

"Making semantics explicit is a communication problem" [84]. Meaning that a language built on a core of shared concepts would provide successful communication [48]. There are many collections of concepts, from simple dictionaries and taxonomies to more advanced collections like thesauri and ontologies. McGuinness [57] states that in its simplest form an ontology may be a controlled vocabulary, a finite list of terms, like a catalog. In its most advanced form an ontology has hierarchies with relationships, classes or concepts and

Figure 1.1: The OneMap increment approach. Submitters contribute with data ranging from local data to world coverage. Taken from [64]

restrictions. It even allows ontologists to state arbitrary logical statements. Consequently the ontologies in this research can be viewed as simple ontologies, equal to taxonomies with extra information.

Ontology is a very old term that originates from the philosophers of ancient Greece. In recent time ontologies have gained interest and acceptance among computational audiences. This has lead to increased availability, which again has resulted in extended development of ontologies. In Geographical Information Systems (GIS) ontologies approaches have been accepted as a very promising approach to solve semantic integration [46]. Nevertheless no de facto standard on how to develop ontologies exists [59]. Even though no standardized methodology exist, some guidelines and helpful methods have been generated.

Development of an ontology always offers several alternatives. The best solution depends on the application in mind and the anticipated extension [23]. Definitions or concepts should be kept as objective as possible and documented with natural language. As a starting point, the ontology should only define concepts that are essential to the communication of knowledge [29], making them easier to understand for a third part. Ontology making is an iterative process, and in each iteration the ontology is extended and refined [27]. These guidelines might help when making decisions about the further development.

Today there exists several different approaches on how to develop and maintain ontologies, and much of this work can be viewed as art rather than science. This situation needs to be changed, and a good methodology can be viewed as an important step in the process.

In this thesis we will provide methods and techniques that can be used in the development of simple ontologies or classifications. This is done by using already existing classifications as a starting point, and build on these incrementally as new data is added. This thesis will not provide a fully built ontology, or any tool to build and maintain ontologies. It will however explore some of the tools and methods that already exist, and how these can aid in the development of ontologies. The data set to be used here are VMAP0, VMAP1 and DNC data.

Chapter 2 introduces important concepts, like taxonomies, thesauri and ontologies, standards like Resource Description Framework (RDF), DAML+OIL, Web Ontology Language (OWL) and tools to use these standards. Finally an overview of existing classifications and hierarchies is given.

In chapter 3 related work is presented. The primarily focus is on ontology development and development methods. There is also a short part about ontology usage in applications.

Chapter 4 deals with ontology development using existing classifications. It starts with a brief discussion concerning important choices and guidelines. Three scenarios are used to reveal semantic heterogeneity between existing classifications and how to solve problems that arise. One of the scenarios is further developed and properties are added. The chapter is finished with a discussion around implementation, ontology languages, and existing tools and APIs.

Chapter 5 summarizes, concludes and outlines the future work.

Chapter 2

Background

The organizing of our environment and the knowledge we have trough aids like classifications, categorization and structure in hierarchies have always been of importance to people. Consider an ordinary day for an individual. First the person gets up at 0700 which is classified as early. Then the person eats a breakfast consisting of cereal and milk, which is classified as healthy. After breakfast it is off to work. Before lunch he is working on boring work, classified as routine. After lunch the person teams up with a group working on the development of a new product, which is classified as interesting or exiting. In the evening the person is reading the newspaper, he just browses the news and sport section, but reads the culture part. When watching TV later, the person is wondering whether the sport on channel 1 is better than the action movie on channel 2.

This example shows us that on one hand, we classify much of the phenomenon or things around us. This is a way to keep control of our surroundings, but it can also be helpful in communication situations. It is this that commercial actors like TV and newspapers take advantage of. They categorize their content in a manner that most people can recognize, and thus people can easily find what they are looking for.

Humans started to categorize their knowledge and information a long time ago. Aristotle is one of the oldest known to categorize and classify. He classified things into categories based on a set of properties, which where shared by all members of a category. For instance he classified animals based on their means of transportation, like air, land or water.

The earliest documented attempt at using hierarchies as means of classifying was by an unknown philosopher in the 5th century, which later has been given the name Pseudo-Dinosysis the Areopagite [47]. The philosopher described the Celestial Hierarchy and the Ecclesiastical Hierarchy. The Celestial Hierarchy describes the intelligent realm and the Ecclesiastical Hierarchy describes the human beings within the church. The combination of Christian religion and hierarchies is not uncommon, as the Roman Catholic and the Eastern Orthodox church were, and still are organized according to hierarchical models. In fact the original meaning of the word hierarchy was "rule by priests".

During the 16th century a new expansion in classification erupted. It started with the Swiss Conrad von Gesner (Conrad Gesner) who made a three-volume work about the Historia Animalum, where he classified the different species or animals according to a set

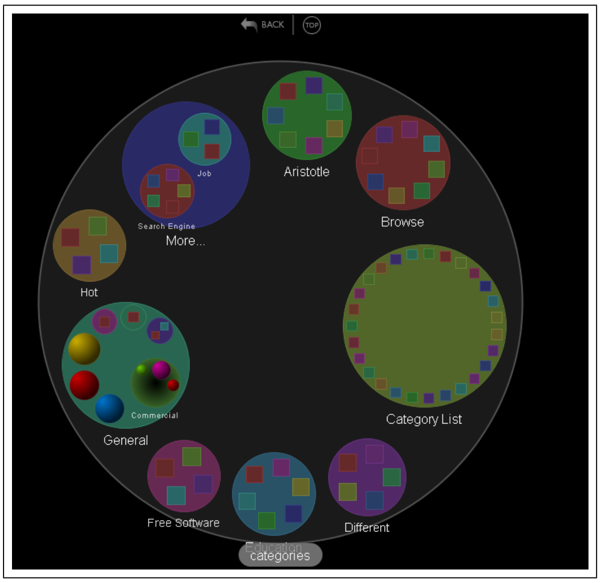

Figure 2.1: Grokker visualization of a search on the word categories

of criterias. Gesner marked the start of modern zoology, a study that culminated some 150 years later with the Swedish scientist Carl von Linnè. The Linnean Taxonomy is as brilliant as it is simple, and it is still used, especially in botanical societies.

In the 19th century, Melvil Dewey developed a library classification system, known as the Dewey Decimal System. The system was finished in 1876, but has undergone several revisions to meet the demands from evolving knowledge and knowledge areas. Along with the Linnean taxonomy, it is maybe one of the most known and used classification in the western world.

One of the foundation thoughts of Communism is to abandon the social hierarchy that exists, and introduce a classless society. This thought was indeed one of the main reasons for the great support that Communism got in some social sets. The Communistic view often came as an opposition to the social systems that had existed in Europe since the middle ages, where a king resided at the top with all the power, and the peasants were at the bottom with little or no power at all. Today the system has changed, but still many human organizations are structured hierarchically. The church has been mentioned earlier, but armies, businesses and political movements also use hierarchies to structure their organizations.

As time has passed new knowledge areas have emerged. New areas that require rethinking and reuse of old organizing principles. One of the last branches to evolve has been computer science. Many Operating Systems (OS) organize the file structure in a hierarchical manner. This makes it easier for humans to understand where to find and retrieve data. The emerge of the Internet has also lead to new areas of hierarchical interest. Internet provides a vast amount of data, and as the number of users continue to grow, even

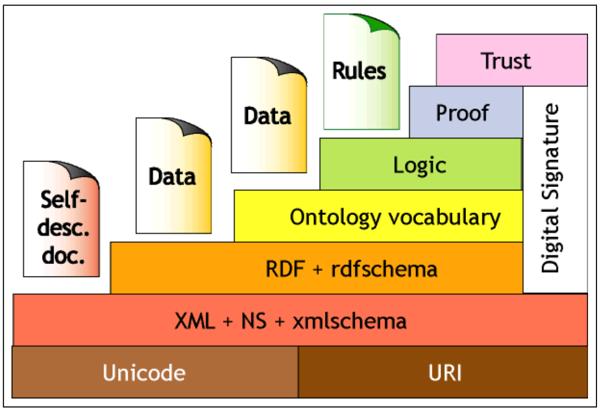

Figure 2.2: Berners-Lee's Architecture (Taken from the Semantic Web presentation [10])

more information is added. People want to get their hands on the right information as fast and easy as possible. As a solution to this several different search engines offer directories that one can browse in order to find the desired information. Examples of such are, Google (directory.google.com), Yahoo (dir.yahoo.com) and the open directory project dmoz (dmoz.org). Some search engines also provide an organization and visualization [73] of the results from your searches, like Grokker (www.grokker.com). Grokker displays your search as a big circle and then divide the big circle into smaller circles with narrower definitions, as shown in figure 2.1. The circles are clickable and a click will zoom in on the selected circle. Another feature with Grokker is that it allows the users to manipulate and make their own organization of data.

Information on the Internet was originally intended for humans to read. The important thing was to get the information out as fast and easy as possible. Today , most of the information on the Internet is still aimed at human consumption. Due to the increasing information available, and the fact that much of the information is not machine readable, means that it can sometimes be difficult to find exact information when searching the Internet [57]. If pages had data that could be understood by programs, then a page could be used more effectively by programs and applications. Many new markup languages have emerged during the last years to address these important issues. The W3C started a collaborative effort to provide a common framework that allows data to be shared and reused across applications. This is known as the Semantic Web [82]. The starting point came from a presentation by Berners-Lee [10] at an XML conference back in 2000 where his vision of the semantic web was presented. The presentation also contained a foundation architecture diagram that is shown in figure 2.2. The figure shows the markup language at the base, just above Unicode. The two next layers, the Resource Description Framework (RDF) and the Ontology vocabulary which are both important assets in the Semantic Web, will be covered later in this chapter.

This remaining of this chapter will first focus on different definitions and areas for classification of data, like taxonomies, thesauruses and ontologies. Then there will be an

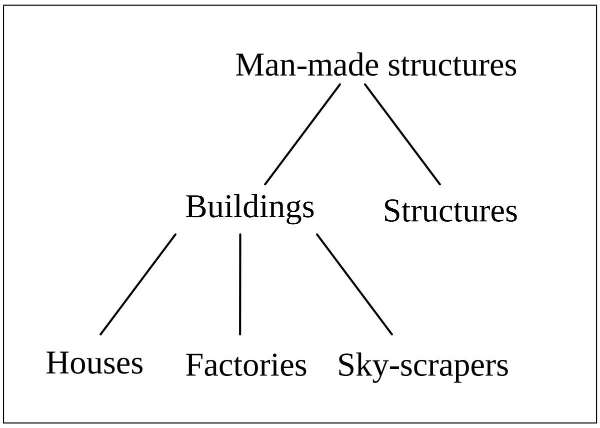

Figure 2.3: A very simple taxonomy. The terms are arranged in a hierarchy with the broadest term at the top

introduction to languages to describe these classification methods. There will also be a short introduction of different tools that are available for building classifications. Finally there will be a brief overview of existing classifications and hierarchies.

2.1 Taxonomy

The word taxonomy can refer to two things. Either a hierarchical classification of things, or the principles underlying the classification. Almost anything can be classified according to some taxonomic scheme. A taxonomy is a tree structure of classifications for a given set of objects. Nodes below the root are more specific classifications that apply to subsets of the total set of classified objects.

In its simplest form a taxonomy is a subject-based classification that arrange the terms into a hierarchy. In this form the taxonomy would contain no information about the relations between the terms. A simple taxonomy is shown in figure 2.3.

The most important pioneer in the work of taxonomy, was the Swedish scientist Carl von Linnè(Carolus Linnaeus). He organized all living organisms in the Linnaean Taxonomy, which is still a widely used taxonomy.

The Linnaean Taxonomy classifies all living things into a hierarchy. It starts with the Kingdom at top, and ends up with Genus and Species at the bottom. The two last classes are often used to uniquely identify species. This is called binomal nomenclature, for humans this is Homo Sapiens, where Homo is the Genus and Sapiens is the specie.

Today taxonomies are most often used in cooperation with biological and botanical areas. Wikipedia has a wiki about species where they try to make an overview and information about species available, this is called Wikispecies [1].

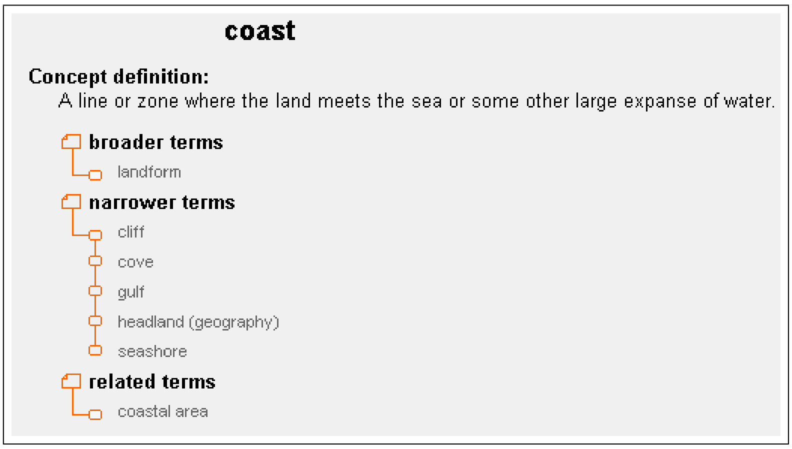

Figure 2.4: Example of a thesaurus. The figure shows an example from the GEMET thesaurus. Notice the Broader, Narrower and Related Terms

2.2 Thesaurus

Thesaurus is basically an extended taxonomy. In addition to describing the world trough hierarchies, thesauruses also allow other statements to be made about the terms.

The term thesaurus is a Latin word, which again comes from the Greek word thesauròs, and the meaning of the word is treasure or repository of words. A name often connected with the word thesaurus is Roget who first published his book *Thesaurus of English Words and Phrases* in 1852. In the introduction Roget described the structure of the thesaurus as a verbal classification [28]. Roget's Thesaurus has been an inspiration for many new thesauri, and many thesauri are currently available on the Internet.

One definition of thesaurus given by several glossaries [1] is as follows, "A list of words showing similarities, differences, dependencies and other relationships to each other". A thesaurus is the most complex type of controlled vocabulary. Although it includes similarities or synonyms, it should not be viewed as a list of synonyms, nor should it be taken as a dictionary since thesauri do not define words. The relationships gives the possibility to navigate and search systems. This will become clearer in the following paragraph.

Making a thesaurus by simply using a simple name list, will easily clutter. Fortunately, by applying three simple rules problems can be avoided [12]. These rules are as follows:

1. **Use a limited list of indexing terms, but plenty of entry terms.**
   In a vocabulary there typical exists many synonyms. For instance Cloaks and Capes, if we have used Cloaks in our thesaurus, a person searching for Capes should be told to search for Cloaks instead. This is done by linking the two terms with the terms USE and USE FOR. USE and USE FOR relationships are used between synonyms that are so close to meaning the same that they do not need to be distinguished.

[1] Buley Library, Southern Connecticut State University. Online Library Learning Center Glossary, Board of Regents of the University System of Georgia. Glossary of Library Terms, St.John University.

2. **Structure terms of the same type into hierarchies.**
   Sometimes terms can be divided into subterms, or some terms can be connected by making a more general term. Dogs can for instance be divided into several sub-species like Border Collie or Greyhound. Dogs then again could be connected to a more general term, for instance mammals or pets. The typical way to link such terms is by using the BROADER TERM (BT) and NARROWER TERM (NT) relationships.

3. **Remind users of other terms to consider.**
   If the hierarchy is restricted to broader and narrower terms, then we need another mechanism to describe the other or related terms that a term can have. Cultivation, which is the practice of growing and nurturing plants outside of their wild habitat, is a part of agriculture. But cultivation will usually take place in a field or garden and therefore it is useful to mention these as related terms. So the RELATED TERM (RT) relationship is used between terms of the same kind, but that are not hierarchical related.

Figure 2.4 shows how the European Environment Information and Observation Network(EIONET) makes use of these terms in their own thesaurus, General Multilingual Environmental Thesaurus (GEMET). The approach taken is to have a description of the term coast, and then the broader, narrower and related terms are listed.

The rich vocabulary to describe terms provided by a thesauri would ease the process of classifying and searching for terms, making a thesaurus a much more powerful tool than ordinary taxonomies. Even though thesauri have a richer vocabulary than taxonomies, the vocabulary is limited compared to the real descriptives; ontologies.

2.3 Ontology

Ontology is a very old term. It originated in early Greece where it occupied Plato and Aristotle, and it is a fundamental branch of metaphysics. A philosophical notion of ontology was also given by Merriam Webster in 1721. According to McGuinness [57] Webster provides two definitions: (1) a branch of metaphysics concerned with the nature and relations of being and (2) a particular theory about the nature of being or the kinds of existens. While ontologies have had a long history, they remained largely the topic of academic interest among philosophers, linguists, librarians and and knowledge representation until recently.

In the recent time ontologies have gained interest and acceptance in computational audiences which have lead to an extended development and availability of ontologies and ontology tools [66] [45] [18].

A frequently quoted, and short definition of ontology, is given by Thomas Gruber [29]: "a specification of a conceptualization". This definition is not very accurate, and there has been some problems defining just what an ontology is. People termed many forms of specifications as ontologies. This diversity is reflected in an overview spectrum, which is shown in figure 2.5. The spectrum was made in preparation for an ontology panel at AAAI in 1999. McGuinness [57] states that in its simplest form an ontology may be a

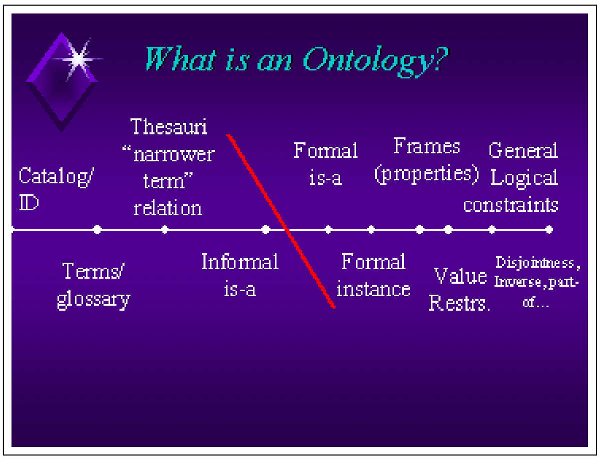

Figure 2.5: An Ontology Spectrum, taken from McGuinness [57]

controlled vocabulary, or a finite list of terms, like a catalog for example. The complex form of an ontology is language that allows ontologists to state arbitrary logical statements, disjoint classes, disjoint coverings, inverse relationships etc. An example of a very expressive ontology language is Ontolingua [2], which will be described in detail in section 2.7.1.

Ontology can be viewed as a collection of shared concepts that a community agrees on. The shared concepts are helpful when solving communication problems within the community, and enables better sharing and reuse of data. According to Kuhn "Any successful communication requires a language that builds on a core of shared concepts" [48]. Studer, Benjamins and Fensel [74], who define ontology as:"an explicit formal specification of a shared conceptualization", means that such a definition makes the ontology a perfect candidate for communicating a shared and common understanding of a domain across people and computers.

It has been determined that ontologies are helpful to achieve a common understanding and communication within a community, but what if to different communities want to share data. Consider two communities that both have defined tall buildings. Community 1 describes a tall building as a building with height over 25 m. Community 2 describes a tall building as a building with more than 10 floors. This mean that the two communities would both be talking about tall buildings but would not realize that they were talking about different concepts. Such semantic differences are sometimes referred to as semantic heterogeneity. Semantic heterogeneity is caused by different conceptualization of real world entities [46]. According to Klien et.al [20] ontologies can be seen as a tool to identify and overcome the problem of semantic heterogeneity. There exists different types of semantic heterogeneity, two of the most common are listed below:

1. Naming heterogeneity (synonyms), the metadata description contains slightly different terminology.

2. Cognitive heterogeneity (homonyms), finding information not relevant to what one need.

One often talks about different kinds of ontologies. Some focus on the domain and application ontolgoies [41] [33] [34]. Catherine Houstis [40] addresses a total of 7 different kinds of ontologies, and Fonseca [24] mentions 4. Some of the different ontology kinds are listed below:

1. General/Common ontologies: vocabulary related to things, events, time, space etc.

2. Meta-ontologies: reusable across different domains

3. Domain ontologies: ontology for a certain discipline, or vocabulary about the concepts in a domain. The domain ontology makes it possible to understand feature definitions between different data sets.

4. Application ontologies: one ontology for each data set, and necessary knowledge for modeling a particular domain.

For instance, geographic data sets have name for mapped or surveyed concepts, such as "road" or "lake", but their precise meaning is not always the same as similar names for concepts in the domain ontology. That is why there must be a distinction between concepts in the domain ontology, and concepts used in the data sets. This is done by constructing an application ontology for every data set involved in the integration process.

Ontologies can be classified along two dimensions, formality and granularity or generality [46].

- Formality
  1. Informal ontology, concept names organized in a hierarchy.
  2. Terminological ontology, concepts defined by natural language definitions and organized in a hierarchy
  3. Formal ontology, further includes axioms and definitions stated in formal language.
- Granularity
  1. Top-level ontology, defines very general concepts which are domain independent.
  2. General ontology, defines concepts that relate to fundamental human knowledge.
  3. Domain ontology, defines concept associated with a specific domain
  4. Task ontology, defines concept related to a particular task or activity.
  5. Application ontology, defines concepts essential for planning a particular application.
  6. Meta-ontology, defines concepts that are common across various domains.

Ontologies can be viewed in levels of depth. The depth of an ontology reflects the complexity of the ontology. At its simplest form an ontology can be viewed as a taxonomy, a hierarchy with relations. Houstis [40] has identified 5 levels that are as follows:

1. Lexicon - a vocabulary with definitions
2. Simple Taxonomy - captures taxonomic relationships
3. Thesaurus - taxonomy plus related terms: captures synonymy, homonymy, etc.
4. Relational Model - Unconstrained use of arbitrary relations
5. Fully Axiomatized Theory- universal, ontologically neutral language; can speicfy/characterize fully a conceptualization.

The key distinction between an ontology and a taxonomy is that ontologies include richer semantic relationships among terms and attributes. These relationships enable the expression of domain-specific knowledge and because ontologies do more than just control a vocabulary they are thought of as knowledge representations.

As an ontology increases in complexity it also uses more of the components available to an ontology. An ontology consists of one or several components. There are 5 main components in an ontology, these are:

1. Classes: are a concrete representation of concepts. The word concept is sometimes used in place of class. Classes are usually organized in taxonomies.
2. Properties: are the attributes of a class. In description logics they are known as roles.
3. Instances: represent specific elements. Referred to as being 'instances of classes'. E.g. Road called E6 is the instance of Road class. Instances are also known as individuals.
4. Relations: a type of interaction between concepts of the domain, e.g. subclass-of, is-a.
5. Axioms: model sentences that are always true, e.g. 1 + 1 is 2

Fundamentally, ontologies provide a shared and common understanding of a domain that can be communicated between people and application systems. Ontologies provides a way to achieve semantic interoperability with the ontology being the interchange format. As shown before, ontologies can assist in communication between humans. Another feature with ontologies is that they can improve the process and/or quality of software engineering processes.

Ontology languages are a good way to present ontologies. Back in figure 2.2 we saw that there was an ontology layer above the markup language layer. At first RDF was used to implement ontologies, but it was soon replaced by the DAML+OIL project. In 2004 the Web Ontology Language(OWL) 2.7 was presented as a W3C Recommendation. Today most ontologies are written in this language.

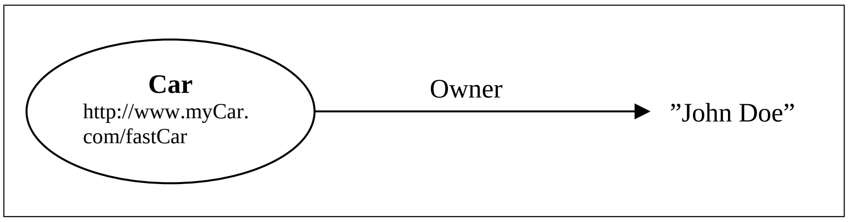

Figure 2.6: A simple RDF statement. The Car is the subject and John Doe is the object

2.4 Metadata

Metadata is simply defined as data about data, or "data which describes attributes of a resource" [17]. It consists of information that characterizes data, like quality and content. Metadata can be helpful to find and characterize data, or make it easier to identify and find data. Consider a worker that needs to organize some articles according to year. To do this he has to read or browse each of the articles. Even this could be insufficient as some articles do not contain this information. But what if there was metadata attached to the article, data about authors, title and year published. The worker could easily find the information needed by simply extracting the metadata.

Metadata is very important for GIS, since it makes data more useful for all types of users. Maps in them selves offer a lot of information, but usually consist of large amounts of data, making operations on the data time consuming. The attributes to maps provide additional information, and the amount of data is small and easy to perform operations and calculations on. "Metadata not only helps find data, but once data has been found, it also tells how to interpret and use data." [70], meaning that metadata encourage data sharing and reuse between organizations and communities.

Standards can improve share and reuse of metadata, as a standard will make work easier for both the user and the cartographer. In the United States the Federal Geographic Data Committee (FGDC) provides a standard for geographical metadata, the National Spatial Data Infrastructure (NSDI) [8]. Among other services the standard provides a way for users to know, what data are available, whether the data meet their specific needs, where to find the data and how to access the data.

Dublin Core [36] is another standardizing project for metadata. The main objective with Dublin Core is to provide metadata that supports a broad range of purposes and business models, and use educational efforts to promote widespread acceptance of metadata standards and practices. The Dublin Core standard includes two levels, a simple and a qualified. The simple level is compromised by fifteen core elements like Title, Subject, Description, and Creator. Dublin Core can be embedded in HTML or XML, or in the machine-parsable Resource Description Framework (RDF) language.

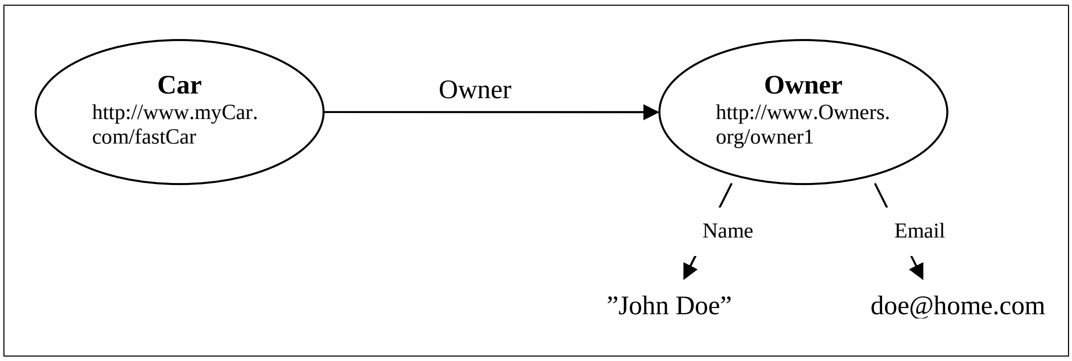

Figure 2.7: A statement with two resources. The Car is still the subject, the object has been changed from John Doe to a resource that represents an owner, in this case it is John Doe.

2.5 Resource Description Framework

It is necessary to say a couple of words about the Semantic Web project [82] [10] before talking about the specific standards that are available. The Semantic Web is a project which aims at giving meaning to information, make information understandable for computers, and make it possible for computers to extract information from the web. Both the Resource Description Framework (RDF) [63] and the Web Ontology Language (OWL) are important parts of this development.

RDF allows multiple metadata schemes to be read by humans as well as parsed by machines. It uses XML to express structure, and thereby allowing metadata communities to define actual semantics. This definition will be further elaborated in the following paragraphs.

RDF provides a model or framework for describing and interchanging metadata [11]. The *Resource* is anything that is uniquely identifiable by a Uniform Resource Identifier (URI). URIs have a number of useful properties including a well developed set of mechanisms for avoiding name collisions, for instance the Domain Name System(DNS). Various Internet protocols make it easy to publish and retrieve information associated with a URI. An example of a resource is a car, and the URI could be 'http://www.myCar.com/fastCar'. A *Property* is a resource that has a name, and can be used as a property to another resource. The "resource" car, mentioned earlier, could have a "property" owner (the owner again being a resource). A *Statement* consists of the combination of subject, predicate, and object. The resource is the subject, that is what is being described. The property is the predicate, or the aspect about the Resource that is being described. This often expresses a relationship between the subject and the object. The object is the object or value of the statement. An example statement is "The Owner of Car is John Doe.". If we replace Car with a resource and use the resource URI, the statement would be as follows, "The Owner of http://www.myCar.com/fastCar is John Doe.". A visualization of the statement is shown in figure 2.6. As shown in the figure and previous statements, the value can just be a string, for example ”John Doe”, but it can also be another resource. ”The Owner of http://www.myCar.com/fastCar is http://www.Owners.org/owner1”. Then a new statement could be ”The Name of http://www.Owners.org/owner1 is John Doe.”. Both of these statements are shown in figure 2.7. It is of course possible to develop the statements further by saying that the owner is a person with a firstname, surname and so on. This shows that the RDF models allows for the creation of resources at multiple levels. The practical and logical limits depends on the domain requirements, and should be addressed and decided by the different communities that use it.

RDF uses XML to define a simple, yet powerful, model for description of resources. As XML is unequalled as an exchange format on the Web, it provides built in distinction between element types and elements. These correspond naturally to the distinction between properties and statements. The reason why XML is not used for metadata exchange is because it falls apart on the scalability design goal [11]. The order in which elements appear in an XML document is significant and very meaningful, whereas the order means nothing in metadata. XML allows for constructions that lead to weird data structures (mix threes, graphs and character strings) in computer memory, which again is time consuming and difficult to handle. Some clear differences can be outlined. Some clear differences can be outlined between RDF and XML. Whereas XML provides interoperability within one application using a given schema, RDF provides interoperability across applications, which again gives greater re-use.

Since RDF provides the ability for resource description communities to define semantics. One of the main purposes is to share these semantics. It is important to distinguish between the semantics made by different communities. The resource owner from figure 2.6 and figure 2.7 is clearly conceptualized different in the two figures. If each figure represented different communities, then the semantics in the two communities would also be different. To detect which of the two approaches to use, RDF use XML-namespaces to unambiguously identify the semantics and conventions provided by the authority of the vocabulary. The Dublin Core [36] initiative provide their own namespace. An example RDF-file using Dublin Core elements to express metadata is shown below.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description>
     <dc:creator>Carl Barks</dc:creator>
     <dc:title>Donald Ducks Fishing adventures</dc:title>
     <dc:description>Donald goes fishing for salmon</dc:description>
     <dc:date>1955-03-22</dc:date>
  </rdf:Description>
</rdf:RDF>
```

RDF can be viewed as an ontology language, but in that case a very week ontology language, and other languages like Web Ontology Language (OWL) and DAML+OIL should be preferred. With RDF it is possible to describe very simple hierarchies, but the limitations become evident when one tries to construct an expression that describes the relationships between classes.

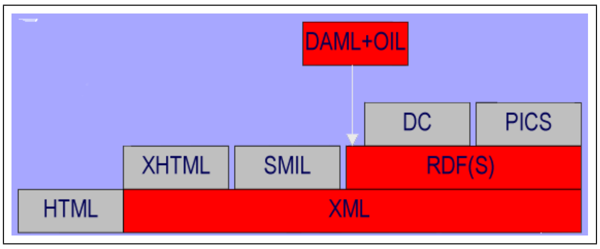

Figure 2.8: DAML+OILs placement (Taken from DAML+OIL an ontology language for the Semantic Web [38])

2.6 DARPA Agent Markup Language and Ontology Interface Layer

DARPA Agent Markup Language (DAML) is a solution to XMLs limited capability to describe the relations between objects. Because of this, DAML is being developed as an extension to XML and RDF. In the latest releases DAML has been expanded with Ontology Interface Layer (OIL). OIL is a proposal for web-based representation and interface layer for ontologies. DAML+OIL [39] [38] provides a rich set of constructs to create ontologies and to make information machine readable and understandable for ordinary users. DAML+OIL builds on RDF and uses the basic ontology primitives, classes and relations. In fact, a DAML+OIL ontology is a set of RDF statements. The placement of DAML+OIL according to other markup languages can be seen in figure 2.8.

DAML+OIL was the starting point for the OWL Ontology Working Group, since DAML+OIL has been gradually replaced by OWL over the later years.

2.7 Web Ontology Language

The Web Ontology Language (OWL) [7] is a markup language for publishing and sharing data on the Internet using ontologies, or as it is said in the OWL guide [88]: "The OWL Web Ontology Language is intended to provide a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications.". OWL was approved in February 2004 as a W3C Recommendation. It is a vocabulary extension of RDF, and it is derived from the DAML+OIL Web Ontology Language [38].

XML is easily readable and is a useful tool when content only needs to be presented to humans. RDF is good for representing information, but OWL has even more facilities for expressing meaning and semantics, and can be used to explicitly represent meaning of terms together with the relationships between terms. In short, OWL can express all than RDF can plus more, this gives OWL greater machine interpretability. OWL allows for more interesting and complex description of classes and properties. Further OWL is intended to be used when the information contained in documents need to be processed by applications, whereas XML is used in situations where the content only needs to be presented to humans.

OWL has three levels of detail, OWL Lite, OWL DL and OWL Full. OWL Lite supports users who primarily needs a classification hierarchy. OWL Lite makes it easier to provide tool support, and a quick migration path for thesauri and other taxonomies. OWL DL and OWL Full are much more expressive than OWL Lite. The main difference between DL and Full is that DL can retain computational completeness and decidability, whereas Full is meant for users who want maximum freedom, but no computational guarantees. There are no reasoning software available for the Full version, and it is unlikely that there will be such software in the near future.

An OWL ontology may include descriptions of classes, properties and their instances. All of these will be described in more detail in the following paragraphs, but the most important feature to notice is that the OWL term for Ontology has been broadened to include instance data.

OWL ontologies consist of classes, properties, individuals and relations between these. The classes and individuals only provide a simple taxonomy. The most basic concept in a domain should correspond to class(es) at the roots of taxonomic trees. For instance in a domain for a country, the country could be the most basic concept. An individual is a member of a class. Countries like France, Russia, and Sweden would be individuals of a country class. The Properties let the user assert general facts about the members of classes and specific facts about individuals. For instance, the class country could have properties like area, population, and capital. The class France, could have properties like 678843 km^2, 63044000, and Paris.

OWL classes are interpreted as sets that contain individuals. In OWL, classes are built from descriptions that specify the conditions that must be satisfied by an individual in order to be a member of the class. A subclass means necessary implication. For example if Man is a Mammal, then ALL instances of Man are instances of Mammal, without exception. If George is a Man, then this implies that George is also a Mammal.

OWL Properties represent relationships between two individuals. There are several different types of properties, but two main types that cover most needs. Object properties and Datatype properties. Object properties link an individual to another individual. Datatype properties link an individual to an XML Schema Datatype value, or a RDF literal. OWL also has a third type of property - Annotation properties. Annotation properties can be used to add information to classes, individuals and object/datatype properties. In OWL, properties may have sub properties, so that it is possible to form hierarchies of properties. Sub properties specialise their super properties. For instance, George hasParent Jerry could be specialised to George hasFather Jerry.

Inverse properties means that if some property links individual a to b, then its inverse property link individual b to individual a. OWL allows the meaning of properties to be enriched through the use of property characteristics.

Domains and Roles in OWL should not be viewed as constrains that has to be checked, instead they are used as axioms in reasoning.

Building a sound and useful reasoning system is not a simple effort, constructing an ontology is much more tractable. Ontology construction will be undertaken by many different organizations and communities as this offers a way to share and formalize meta-data, or data about data.

OWL is written in XML syntax, and the following text shows an example of a class:

```
<owl:Class rdf:ID="Disney_cat">
   <rdfs:subClassOf>
     <owl:Class rdf:about="#Cartoon_cat"/>
   </rdfs:subClassOf>
 </owl:Class>
```

From the text we can deduct the following facts. The class Disney_cat is a subclass of Cartoon_cat. Another thing to notice is that it is in fact the RDF language that has defined the subClassOf tags.

The following is an example of an Object Property, and a Datatype Property.

```
<owl:ObjectProperty rdf:ID="Is_friend_of">
   <rdfs:range rdf:resource="#Cartoon_star"/>
   <rdfs:domain rdf:resource="#Cartoon_star"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="In_Year">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Again one can see how OWL and RDF is used together. The Object Property is called Is_friend_of, and has a range of Cartoon_star. This means that the property takes a Cartoon star object as value. The domain shows the class where the property belongs. In this case that is also the Cartoon Star class. So the property Is friend of says that a Cartoon star can have a friend that is another Cartoon star(For instance, Mickey Mouse and Goofy). Finally the rdf:type tag describes that the ObjectProperty is a SymmetricProperty because the domain and range of the property is the same. The Datatype Property can be of one of several different datatypes, for example string, integer, float or boolean. In our example the datatype is a string.

2.7.1 OWL-tools

Since the Recommendation of the Web Ontology Language, numerous different implementations, or tools have been made. These tools vary from KR development inspired tools, like Protege, OilEd and OntoEdit, to more URI based solutions like Swoop. Traditional OWL tools consist of a GUI part to develop and manage ontologies. This GUI part is usually combined with some kind of reasoner that can validate, or reason, the ontologies. Today there are also some APIs available for OWL development.

Protègè

Protègè [67] is a free, open source ontology editor and knowledge-base framework that is based on Java. It is a tool that allows users to construct domain ontologies, customize data entry forms, and enter data. The platform can easily be extended to include other components, such as tables, graphs, sounds, images and storage formats as OWL, RDF and XML through several available plug-ins. The terminology to describe the components

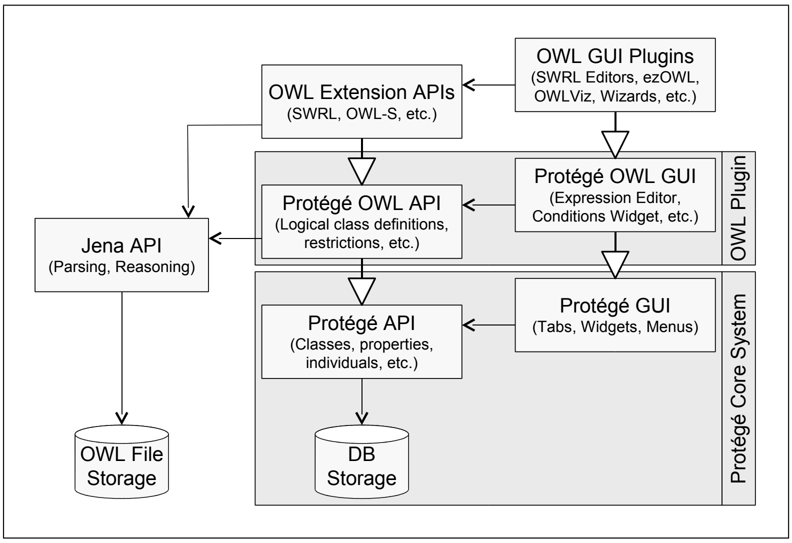

Figure 2.9: The figure shows the position of the Protègè OWL Plugin in proportion to the Protègè core system. The figure is taken from Knublauch et.al [37]

in OWL and Protègè differs. An OWL ontology consists of Individuals, Properties and Classes, whereas the Protègè consists of Instances, Slots and Classes [56]. Classes may be organized into a superclass-subclass hierarchy, which is also known as a taxonomy.

The Protègè OWL Plugin is a complex Protègè extension that can be used to edit OWL files and databases [37]. Figure 2.9 shows the Protègè structure with its core system at the bottom and the OWL Plugin located on top of the core system.

OntoTrack

OntoTrack [50] [51] is a new browsing and editing ontology authoring tool for OWL. The slogan for OntoTrack is Fast Browsing and Easy Editing of Large Ontologies. So the main focus for OntoTrack is on large and complex ontologies, making it easier to efficient navigation and manipulation. The system is implemented in Java2D, and based on a direct acyclic graph presentation of ontologies.

OilEd

OilEd [9] is an ontology editor allowing the user to build ontologies using DAML+OIL. The initial intention behind OilEd was to provide a simple editor that demonstrates the use of, and simulated interest in, the OIL language. OilEd does not provide a full ontology development environment, rather it is the "NotePad" of ontology editors, offering enough functionality to allow users to build ontologies and to demonstrate how we can use the FaCT reasoner to check those ontologies for consistency. It is implemented in Java.

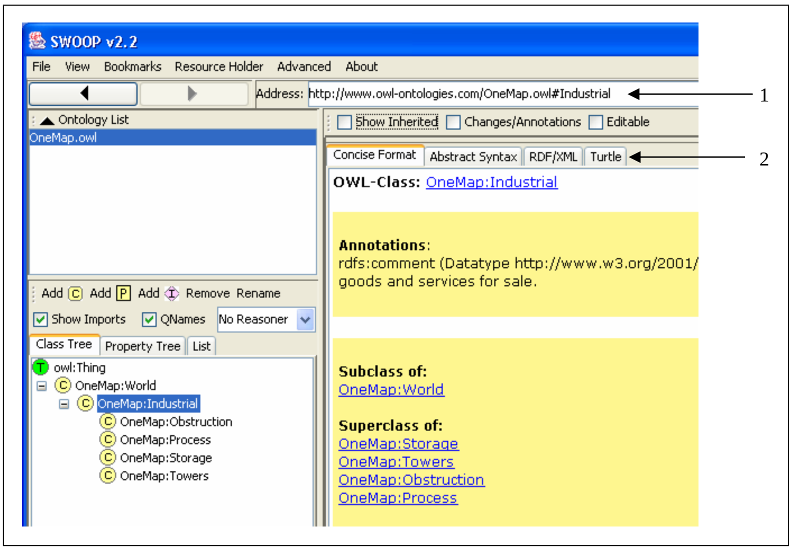

Figure 2.10: A picture of the Swoop program. Notice that the user interface is built up as a browser environment.

Swoop

Swoop [3] is a tool for building OWL ontologies. Unlike many other traditional ontology development tools which are inspired by traditional KR-based paradigms with steep learning curves, Swoop has taken a web like approach to the user interface, meant for rapid and easy browsing and development of OWL ontologies.

Swoop is a simple scalable hypermedia-inspired OWL ontology browser and editor. Hypermedia meaning that it uses URIs to support hypertextesque navigation through and between ontologies. The ontology is presented as a web page, with the classes, properties individuals being analogous to HTML anchors embedded in the page. Swoop provides the user with the option to render the ontological page in several formats. This can be seen in figure 2.10 where the tabs provided are Concise Format, Abstract Syntax, RDF/XML and Turtle as seen by arrow number 2. Arrow number 1 shows the HTML anchors.

Ontology Explorer Tool

Ontology Explorer Tool (OntoXpl) [68] is not an ontology Editor, but an ontology information exploration tool. It helps users quickly understand the ontology domain by going through functions provided by OntoXpl. OntoXpl retrieves the implicit information and reorganizes them in a way such that users can get a global picture of ontology information. It helps users understand the structure and navigate the knowledge-base efficiently.

OntoXpl has been developed based on Tomcat 5.0 Web Server environment. In order to run OntoXpl, users have to download and install Tomcat. It also depends on a connection and communication with RACER, so RACER also has to be run first. OntoXpl is made by the Concordia University in Canada.

Ontolingua

Ontolingua [2] is a knowledge system made by the Stanford University. It is web based and the ontologies are stored on an ontology server. The web-based approach provides the users with the ability to publish, create and share ontologies. A centralized storage of ontologies also enables reuse and browsing of existing ontologies. This tool is useful when parts of communities are assembled at different geographical places.

RACER

RACER [31] is a Semantic Web inference engine for developing ontologies, query answering over RDF documents and wrt specified RDFS/DAML ontologies. It is also a Description Logic reasoning system with support for TBoxes with generalized concept inclusions, ABoxes and concrete domains. Finally it is also a provider for modal logic Km with graded modalities and axioms.

OWL-APIs

Since OWL was approved as a W3C recommendation in February 2004 several different OWL-APIs have emerged. These APIs allows for applications to make and manage ontologies using the OWL language. Today most API s are made in java, this is at least the case for the OilEd [9] tool that provides an own API [76]. The Protègè [67] also provides an API [37]. Some commercial actors are in addition starting to find interest in ontologies, among them are HP that have made their own API [43]. Common for all these APIs is that they are programmed in Java.

2.8 Existing Classifications, and Hierarchies

2.8.1 Wordnet

WordNet [26] is an online lexical reference system where English nouns, verbs, and adjectives are organized into synonym sets, each representing one underlaying lexical concept. WordNet is a combination of a dictionary and a thesaurus, it groups the English language into sets of synonyms and records the various semantic relations between the sets. The development began in 1985, and as of 2005, it contains about 150000 words, organized in over 115000 sets for a total of 203000 word-sense pairs. A typical search result is shown in figure 2.11.

2.8.2 Roget's Thesaurus

Roget's Thesaurus [42] was published in 1852 and is the world's best known thesaurus. The Thesaurus was created by Dr. Peter Mark Roget. It consists of six primary classes, each

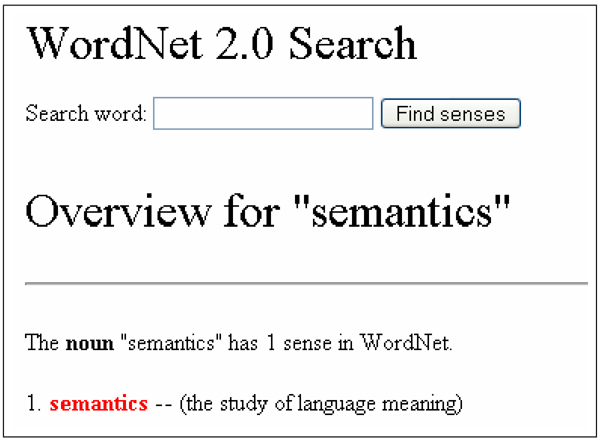

Figure 2.11: WordNet search for semantics

of these classes is composed of multiple divisions and sections. This can be conceptualized as a tree containing over a thousand branches, and based on this Roget's Thesaurus can be viewed as a classification system.

2.8.3 GEneral Multilingual Environmental Thesaurus

Whereas WordNet was merely for the English language, GEMET provides an environmental thesauri for 19 different languages. GEMET [72] has a core terminology of 5,400 generalized environmental terms and their definitions [44]. As mentioned earlier it is multilingual, and translated into 19 languages like, English, French, German and Russian, to mention some. GEMET was developed by the European Environment Agency and the European Topic Centre on Catalogue of Data Sources together with international experts. GEMET is a reference vocabulary of the European Environment Agency(EEA) and its Network (EIONET)

The basic idea for the development of GEMET was to use the best of the currently available multilingual thesauri, in order to save time, energy and funds. GEMET was conceived as a "general" thesaurus, aimed to define a common general language, a core of general terminology for the environment.

The thesaurus has a grouping that can be viewed as a hierarchical system. It is first divided into four main groups, these super-groups do not have any information, they are simply groupings to easier get an overview of the organization of data. The four main groupings are then divided into a numerous new concepts.

2.8.4 Alexandra Digital Library (ADL) Gazetteer

Gazetteers is list of geographic names, together with their geographic locations and other descriptive information. A gazetteer is by some [90] [52], viewed as a kind of geographical thesaurus. ADL Gazetteer [77] is a gazetteer containing an overview of most placenames in the world. The gazetteer is also designed in a hierarchy, so that information is easy to search and locate. As of today there are approximately 4.4 millions entries.

2.8.5 Thesaurus.com

Is an online resource for finding acronyms and synonyms. Here a thesaurus is defined as a support for finding synonyms. A search in the thesaurus returns hits that either describes the term, or terms that are synonyms to the word. The search also provides a definition of the term.

2.8.6 Dewey Decimal Classification

The Dewey Decimal Classification (DDC) is a system of library classification developed by Melvil Dewey in 1876. Dewey wanted to have a hand in the education of the masses and he fulfilled it by developing a system which put related topics in an hierarchical order from general to specific as needed [55]. The system has undergone major modifications on several occasions, the last one being in 2004.

DDC is divided into ten main classes. Each class is represented by a 3 number digit, starting on 000. In this use of numbers also lies the cleverness of DDC, it allows DDC to be purely numerical and infinitely hierarchical. Each of the ten main classes are then divided into 10 divisions, and the divisions are divided again, into sections. The ten main classes are:

- 000 Generalities
- 100 Philosophy and psychology
- 200 Religion
- 300 Social science
- 400 Language
- 500 Natural sciences and mathematics
- 600 Technology (Applied sciences)
- 700 The arts
- 800 Literature and rhetoric
- 900 Geography and history

The numbering makes it easy to identify books. For instance 948.1, 900 shows that this is either Geography or history, 40 shows that it is General history of Europe, and the 8 shows that it is for Northern Europe, or Scandinavia, and finally the .1 shows that it is for Norway. So a book with the number 948.1 would be a book about general history of Norway.

2.8.7 Cyc and OpenCyc

Among all the different knowledge based projects, started in 1984 Cyc [49] was one of the first. The goal of Cyc is to enable applications to perform human like reasoning. To do this Cyc attempts to assemble comprehensive ontology and database of everyday commonsense knowledge. At the present time, Cyc contains nearly two hundred thousand terms. The knowledge base also includes several hand-entered assertions about or involving each term. This combination allows for reasoning about the information using natural-language processing. The fact that the knowledge base is as big as it is, makes it possible for Cyc to reason about natural language that traditional natural-language systems have difficulties solving.

Cyc is proprietary, but a smaller version of the knowledge base, OpenCyc was released under an open source licence. It originally contained 6000 concepts and 60000 assertions about these, but in version 0.9 there are 47000 concepts and 306000 assertions. Among its features is a specification of CycL, the language in which Cyc is written, and the Cyc API for application development. OpenCyc also provides coverage for DAML and the possibility of linking with WordNet.

2.8.8 Geographical Standardizations, Classifications and Gazetteers

Although vast amount of geographical data exists there is a lack of standards and classifications that can help utilize more use and reuse of such data. There is however a lot of work being done by several vendors, open communities and standard organizations. One of the most important contributors is the Open Geospatial Consortium (OGC), who contributes within several different areas to make Geographical information easier to use by providing specifications [58].

There also exists other participants that have made their own standard, like North Atlantic Treaty Organization (NATO), The European Committee for Standardization (CEN), International Organization for Standardization (ISO) and the United States Geological Survey (USGS). An overview of many of these standards are given in [5]. The standard reflect the intended area of usage. For instance, Epicentre that is specialized for the Petrol Industry. Nevertheless some of the standards are of a more general type, and will be described in the following paragraphs.

Digital Geographic Information Exchange Standard (DIGEST) [5], [71] was prepared and issued under the authority of the Digital Geographic Information Working Group (DGIWG) to promote the exchange of geographic information between the defense authorities of Belgium, Canada, Denmark, France, Germany, Italy, The Netherlands, Norway, Spain, The United Kingdom and the United States. DIGEST employs the U.S. Department of Defence's Vector Product Format(VPF). At the conceptual level, it resembles the US Spatial Data Transfer Standard.

The Spatial Data Transfer Standard (SDTS) [6], has the limited scope of being a language for communication spatial information. Since the development of the standard started in the 1980's it has merely evolved as an exchange format. The first truly object-oriented specification of geo-spatial information was done by the Canadians in 1993. It is called the Spatial Archive and Interchange Format (SAIF).

Another way to organize geographical data is through gazetteers which is a geographic dictionary index. The Alexandra Digital Library Gazetteer [52] has been mentioned earlier in this chapter, but there are others as well. USGIS through their Geographic Names Information System (GNIS), National Imagery and Mapping Agency's (NIMA) Geonames Server, the Canadian Geographic Names Service, the Getty Thesaurus of Geographic Names (TGN) and a growing number of new ones [35]. The TGN also provides a hierarchy that the user can browse to find information.

2.9 Summary

In this chapter an overview of different categorization, classification, and structure of information in hierarchies was given. Terms like taxonomies, thesauruses and ontologies were presented. An overview of possible formal description languages like Resource Description Framewok (RDF), DAML+OIL and Web Ontoloy Language (OWL) has been given, and finally an overview of existing classifications and hierarchies was shown. The rest of this thesis will use simple ontologies for information classification, and OWL will be used as a representation language.

Chapter 3

Related Work

The research in semantic integration is motivated by the increasing amount of Geospatial data that is becoming available, and the heterogeneity that exists between different data sets. The complexity and richness of geographic data and the difficulty of their representation raise specific issues for semantic integration. Sharing and integration of geodata between systems is often a very complex process. Poor documentation, obscure semantics, diversity of structure and models are just some of the problems that needs to be solved [78], [80].

Early discussion on integrated GIS, or IGIS as it is often referred to, was about integrating diverse GIS technologies, or to reflect a particular point of view of a community. According to Fonseca et al. [25] this can be traced back to Ehlers et al.(1991) and Davis et al.(1991). The first to point out that the next generation of GIS would need something similar to ontologies, was Nunes in 1991.

The most normal notion is that the ontology should be used as a tool or framework to ease semantic integration [46]. There is even some who mean that it should act as a component on the same terms as databases, and that the whole system should be ontology driven [24], [78]. Nevertheless in order for the ontology to function as a component, it has to be developed in a manner that ensures quality and consistency.

The rest of this chapter will be about different methodology used in ontology development and specific development used in the geospatial domain. The last section will look on how ontologies are used in some applications today.

3.1 Approaches and Methods

Ontologies provide significant benefits for the design and use of geographic information. An ontology define semantics independently of data representation, and reflect the relevance of data without accessing them. Such a high-level description of semantics of geographic information provides more and new means for comparing and integrating spatial data [87].

To achieve a good result, development of ontologies should be based one some guidelines or methods. According to Uschold and Gruninger [59], there are no standard methodologies

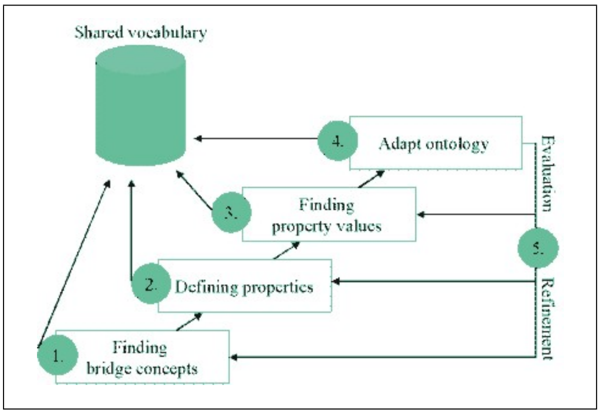

Figure 3.1: Steps of the development process, notice step 5 that can trigger the previous steps. Taken from [27]

for ontology building. This is clearly a situation that needs to be changed, and will only be changed through an understanding of how to construct ontologies [16]. Even though no standardized methodology exist, a number of suggestions for methodologies has emerged.

One of the first to define and provide guidelines on how to develop an ontology was Gruber [29]. In order to make an ontology a set of objective criteria should be followed. There are five different criteria in total, these are:

- Clarity
- Coherence
- Extendibility
- Minimal encoding bias
- Minimal ontological commitment

Clarity means that the ontology is easy to understand, and easy to communicate about for a third part. The definitions should be kept as objective as possible, and documented with natural language. Coherence means that the ontology should sanction inferences that are consistent with the definitions. Since an ontology is expected to be used in shared vocabulary, a extendibility approach is desired. Extendibility, meaning that it should be possible to define new terms based on the existing vocabulary, in a way that does not require revision of the existing definitions. The ontology should consist of as little symbol-level encoding as possible. It should also be defined by specifying the weakest theory and only define terms that are essential to the communication of knowledge. An ontology design, will always be a trade off between these criteria, and the result will depend on the use and purpose of the ontology.

Another approach is taken by Schuster and Stuckenschmidt [27]. They describe a process in five steps, where each step is executed in sequence resulting in a partial specification of a shared ontology. Whereas Gruber is a bit abstract in his steps, Schuster and Stuckensmchidt take a more concrete approach. To ensure that the ontology is extended and refined, the last step is an evaluation step that can trigger the previous steps as shown in figure 3.1. The five steps are:

- Step1: Finding Bridge Concepts.
- Step2: Definition of Properties.
- Step3: Finding property values.
- Step4: Adapt ontology
- Step5: Refine Definitions

The first step makes a concept that encloses all classes from the source and destination systems. It is important to keep this bridge concept as concrete as possible in order to achieve an exact classification. More than one bridge concept can be defined if needed. The next step is to find properties that describe the chosen bridge concept, and then step three fills the attributes with values. Step four brings in supporting ontologies to get a better understanding of the domain, and finally step five makes it possible to redefine step 1 to 4.

This methodology is intended to use when building shared ontologies. For communication to function properly it is important that the concepts in the ontology are defined as closely as possible to a vocabulary, and conceptualization that is widely accepted as a standard. To achieve this, existing information sources like WordNet and GEMET can be used to build an ontology on top of existing classifications. Schuster and Stuckenschmidt enlighten their approach by the use of an example problem. There are two existing classifications and a shared ontology will be built on top of this. First, generalize the problem area and come up with shared concepts. A definition of the concepts are derived from various information sources. Some information sources even provide a full upper ontology that can be used. This way a very general but good definition of the upper ontology is achieved, and make it applicable to several different communities.

Other attempts or suggestions that are similar to the previous approach has emerged. A common starting point is to merge data by making upper ontologies based on existing data. The existing data can either be ontologies, geographical data or other classifications. Timpf and Hakimpour [21] makes an upper ontology by building on existing ontologies from different systems or communities. The upper ontology is created by using a reasoning system that find the similarities between concepts from two systems. A similar approach is taken by Hakimpour and Geppert [32]. In the reasoning system they identify four levels of similarity between two coherent intentional definitions. These are as follows:

1. Disjoint definitions
2. Overlapping definitions
3. Specialized definitions

4. Equal definitions

The reasoning can also be used other ways, Klien et al. [20] uses the reasoning for information retrieval. To perform the retrieval they use a shared vocabulary as an upper ontology. The shared vocabulary consists of concepts that have a common consensus in a community. It is important that the vocabulary is general enough to be used across all information sources, but specific enough to make meaningful definitions possible.

Prolog can also be used as a reasoner, as by Uitermark et al. [33]. The development of ontologies is based on finding object definitions that make data sets semantically transparent to each other. Using two existing data sets over the same geographical area, an ontology is made by finding connections between the two data sets. The connections are found by using Prolog as a reasoner. The process is cyclic and might have to be done over several times.

In [85] a skeletal methodology is described, this is refined in [86]. Before it is finally redefined to consists of 5 steps in [59]. What makes this methodology different from the others is that it addresses the importance of establishing guidelines and best practices for each stage. Then others can benefit from this, and expand the methodology based on their own experiences. The five steps are:

- Identify the Purpose and Scope
- Building the Ontology
  - ontology capture
  - ontology coding
  - integrating existing ontologies
- Evaluation
- Documentation
- Guidelines for each phase

The first step should identify the purpose and intended users of the ontology. Identifying the purpose and scope act as starting point for step two, to build the ontology. This step is divided into three more sub-steps. These are, a capture step that identifies the key concepts, produces a definition of the concepts and identifies terms. A second coding step for representation of the concepts captured in the previous step. Finally an integration of existing ontologies steps. It is difficult to determine what parts of existing ontologies to integrate, and also to achieve an agreement within the community. The third step, evaluation, is to evaluate the ontology. The fourth step is to document the ontology in collaboration with the fifth step. The final step is to include a set of techniques, methods, principles for each of the four steps above. The first, third, fourth and fifth steps can be viewed as general guidelines, while the second step, ontology building, can be viewed as specific guidelines for ontology building.

In the building of an ontology the hierarchy is an essential part. There are three different methods on how to develop hierarchies, top-down, bottom-up, and a combination called middle-out [23]. These different methods can be summarized as follows:

- *Top-down:* starts with the definition of the most general concept in the domain and then specialize the concepts to make subclasses.
- *Bottom-up:* starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.
- *Combination or Middle-out:* starts with defining the core, or most important concepts first, and then generalize or specialize them appropriately.

None of these methods are inherently better than any of the others, they all have their advantages and weaknesses. The approach to take depends strongly on the personal view of the domain, and the intended usage for the ontology.

Methondology is another way to develop ontologies which relies heavily upon Knowledge Based Engineering. Methontology have identified the following activities that are involved in the development of an ontology [22] [69].

1. Specification
2. Knowledge acquisition
3. Conceptualization
4. Integration
5. Implementation
6. Evaluation
7. Documentation

Identifying the purpose and scope of the ontology is called specification. The output of this phase is a natural-language ontology specification document. Phase 2 largely occurs in parallel with phase 1. Any type of knowledge source and any elicitation method can be used. The conceptualization phase identifies terms as concepts, instances, verbs relations or properties and each are represented using an applicable informal representation. Integration of definitions from other ontologies helps to obtain uniformity across the ontologies. The implementation phase represents the ontology formally in a language. Much emphasis is placed on the evaluation stage. The techniques are largely based on methods from Knowledge Based Systems. Finally a collation of documents that result from the activities.

The last methodology to be described here states that the development of an ontology includes these practical terms: defining classes in the ontology, arranging the classes in a taxonomic hierarchy, defining slots[1] and describing allowed values for these slots, and filling

[1]The properties of classes is sometimes referred to as slots [23], but with the W3C OWL language Recommendation in February 2003, properties have been established as a standard.

in the instance values into slots [23]. Before the ontology development is started, some fundamental rules that can be applied when making design decisions. These are as follow:

1. There is no correct way to model a domain - there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.

2. Ontology development is necessarily an iterative process

3. Concepts in the ontology should be close to objects(physical or logical) and relationships in your domain of interest. These are most likely to be nouns(objects) or verbs(relationships) in sentences that describe your domain.

Based on these basis sentences Noy and McGuinness have identified seven steps for developing an ontology. These are as follows:

- Step 1. Determine the domain and scope of the ontology
- Step 2. Consider reusing existing ontologies
- Step 3. Enumerate important terms in the ontology
- Step 4. Define the classes and the class hierarchy
- Step 5. Define the properties of classes - slots
- Step 6. Define the facets of the slots
- Step 7. Create instances

The first step is done by answering several basic questions about the domain of the ontology, the use of the ontology, possible information retrieval scenarios, and how the users are. The second step is self-explanatory, and it is important to remember that the use of existing ontologies can save a lot of time and work. The next three steps depend a bit on each other. First the user should write down terms and arrange these according to importance. When this is done, all the nouns in the terms would probably be classes and the remaining terms would become slots, or properties, for the classes. The sixth step adds further information about the properties. Facets describe value type, allowed values and cardinality to mention some. The last step is also straight forward, simply creating individual instances of classes.

Summary

Upper ontologies seems to be the most common way to use ontologies in a geographical relation. The upper ontology is used as an connection between two or more existing data sets. Few of these proposals offers a solution on how to develop ontologies from bottom and up, for instance basing the start on existing classifications or the like.

The methodologies presented in the previous section can be divided into two main parts. First are the methodologies that provide general guidelines, like Gruber [29], Uschold and

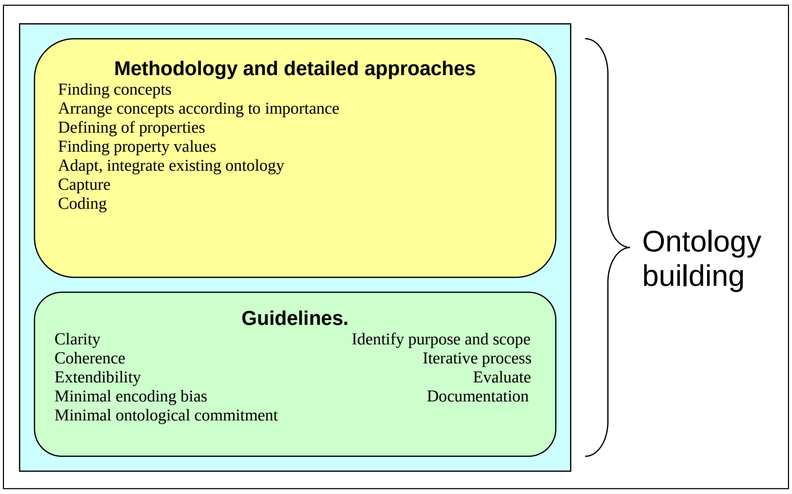

Figure 3.2: Suggestion to ontology building. First determine the guidelines, then start the development of the ontology and use the methods that are available

Gruninger [59] and the first steps described by Noy and McGuinness [23]. The other main approach consists of more concrete, or handfast steps, in ontology development, as Schuster and Stuckenschmidt [27], and the last steps mentioned by Noy and McGuinness [23].

How to choose what method to use depends on the ontology to be built. The best might be to develop ontologies based on a merge of the two approaches. First the general guidelines could be used to determine a reference framework. The reference framework could help clarify and allow the developer to get a better overview of the domain, thus making the next phase of specific development easier. Figure 3.2 shows this concept in a model.

3.2 Different ways to use ontologies

Last in this chapter we will look at examples where ontology is used. In most cases it is used as a tool for information retrieval. This is here shown through the BUSTER project and the Pizza Finder Application uses ontologies.

Bremen University Semantic Translator for Enhanced Retrieval (BUSTER) [81] is a project undertaken by the Bremen University. To put it short, BUSTER is an application available from the Internet. The application consists of two parts, BUSTER/Q which is a tool for intelligent information retrieval, and BUSTER/SI, which is a tool for the semantic integration of heterogeneous data sources. BUSTER/Q is available as a prototype on the web-page, while BUSTER/SI is still under development (as of may 2005). BUSTER/Q needs access to detailed description of the data sources and services. This metadata is available in the form of Comprehensive Source Descriptions (CSD), which is formalized in

XML/RDF format. After the retrieval of information sources, semantic integration may be needed. During this integration some challenges arises, the data has to be reclassified so that they fulfill each others catalogue structure, the way BUSTER deals with this problem is through something similar to an upper level ontology.

The Manchester Pizza Finder is a simple application which uses a Pizza Ontology, the one developed in the Protègè-OWL tutorial [56], and a remote Description Logic (DL) reasoner to dynamically generate pizza toppings and pizza toppings categories. The user can include and exclude toppings and the description logic reasoner is used to determine if the choices contradict each other. After the user has made his topping choices the DL reasoner perform a query and returns all results that satisfy the query.

Chapter 4

Development of Ontology using Existing Classifications

In this chapter an ontology based on existing classifications will be developed. A set of scenarios will be used to enlighten different semantic problems, or semantic heterogeneities, that might occur in a semantic integration process. Possible solutions to the problems will be given, and one of the scenarios will be used to make a full development of an ontology.

4.1 Establishing working conditions

A set of working conditions have to be established before the development can start.

- Determine type of hierarchy to be used
- Define the context
- Decide on the development methodology

4.1.1 Determine hierarchy type

In order to avoid geographic heterogeneity a framework is needed to seamlessly integrate new data [78]. In this thesis the main focus is on the development of a framework or tool that can be used as a classification facility, and eventual help users to easier integrate new data.

Common to many classification methods is the use of some kind of hierarchy to classify concepts in a domain. The three most common classification methods, Thesaurus, Taxonomy and Ontology have taken this approach. These three ways to classify data have their strengths and weaknesses. Taxonomies arrange the concepts, or definitions into hierarchies from a broad to narrower term, but little or nothing descriptive information is given about the concepts. Thesauri also use a hierarchy to classify the information within a domain, but in addition they have descriptive information about the concepts in the hierarchy as well. Ontologies are similar to the two previous methods in many ways, but ontologies give the possibility to add additional information about the concepts and their relations. This means that ontologies provide a way to describe how concepts relate to each other, giving users an understanding of, and facilities to communicate between each other about concepts in a domain. It can be difficult to separate different classification methods, and in many ways all the classification methods mentioned here can be viewed as forms of ontologies [57].

In the work that is done here it is not necessary to use descriptive language for describing the relationship between concepts. The desired model here is to have a hierarchy with classes and properties. Both classes and properties should be given descriptive definitions. This is done to easier identify which features and attributes that fall within the definitions. Based on these facts the hierarchies here should be considered as simple ontologies.

4.1.2 Define the context

The context contains the setting in which the ontology will be developed and used. Three factors that influence the development process in one way or another can be identified.

- Domain, the domain that the ontology will be used in. All classification are usually valid for a domain of knowledge. The domain will dictate the arrangement and set-up of the ontology.

- Users, what is the knowledge level to the users that will interact or use the ontology (classification). If the users are domain experts the information can be highly detailed, but if the users are on an ordinary level, the level of detail should be kept at a general one.

- Usage, how and what will the ontology be used for. An ontology within a domain can have different usages. Is it used for browsing, retrieving data, or to determine the classification of new species.

In this paper the domain will be geographical information. Geography can be a location and spatial variation in both physical and human phenomena on the Earth. As this can be a very broad and diverse field, communities will require different ontologies to map their needs. For instance, both the mapping authority in a country, and a local hiking group that use GPS to make their hiking routes, create information within the geographical domain, but with clearly different interests and needs.

OneMap [64] among many things encourages users to submit new data. Some kind of classification is needed before submitting new data. The data will be classified against an ontology and then subjected to a peer-review. The usage will be in connection with a world wide coverage of different data, and the users range from mapping agencies to individuals. According to these demands it is important that the ontology is as general and intuitive as possible.

For an ontology to be as intuitive as possible some measures have to be taken. Concepts or class names should be as descriptive as possible. A class with the name Ve23 tells the user absolutely nothing, but a class with the name Vegetation would give the user a better understanding of the concept. The broad diversity of users will lead to different interpretations of concepts. Consider a captain of a tanker and a captain of a leisure boat. The captain of the tanker would refer to a holme as an obstacle, whereas the captain of the leisure boat would refer to it as a possible sunbathing spot. Taking this into consideration, the ontology should clearly be expanded with descriptive information about the classes, hence more like a thesaurus or vocabulary.

The ontology should be as general as possible, but taking all the different users into consideration this is not a simple task. New classes will also emerge as new developments and technologies emerge. Insertion of new data might lead to refinement or reclassification of the ontology. For example, at first it is decided that the class road is adequate to cover the needs, but as new data is inserted, it becomes clear that this is a too wide concept that has to be split up into more descriptive sub-classes.

The generality of an ontology is never easy, but the ontology in question will be subjected to different usages. It will be browsed, used to classify new data, and retrieval of stored data. Having this in mind, it could be wise to start with the ontology as general as possible, and then make it more specialized as it evolves.

4.1.3 Development approach

As mentioned in chapter 3, no de facto standard for ontology development exists. This has resulted in a number of different approaches emerging from different sources.

The approach taken here is to use existing classifications from already existing data sources. These classifications will be taken from the three data sets, Vector Smart Map Level 0 (VMAP0), Vector Smart Map Level 1 (VMAP1) and Digital Nautical Chart (DNC). The data will be used to uncover problems and uncertainties that might occur during an ontology development process.

4.2 Data sets

4.2.1 VMAP0 Data

VMAP0 [79], [62] is an updated and improved version of the Digital Chart of the World. It provides worldwide coverage of vector-based geospatial data. VMAP0 includes major road and rail networks, hydrologic drainage systems, utility networks, major airports, elevation contours, coastlines, international boundaries and populated places. Figure 4.1 shows some of the different coverages in VMAP0. The area shown is New York City, blue is water, orange is urban areas. In figure 4.2 we can see the railroad and road network in the New York area. The data is available from the mapability [53] web page. The data in VMAP0 is derived from either Operational Navigation Chart or Jet Navigation Chart. The ONC have an horizontal accuracy of 2040 meters, and the JTC have a horizontal accuracy of 4270 meters. The vertical accuracy on contours are +- 152.4 meters and +- 30 meters on spot elevations.

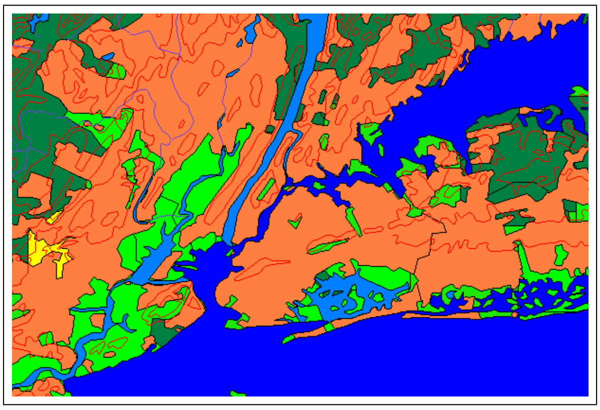

Figure 4.1: VMAP0 coverages, urban, trees, swamp, grass and cropland

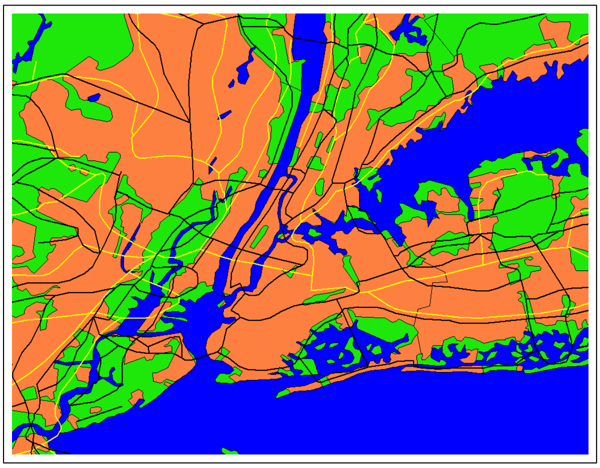

Figure 4.2: VMAP0 transport network, displaying the roads and railroads

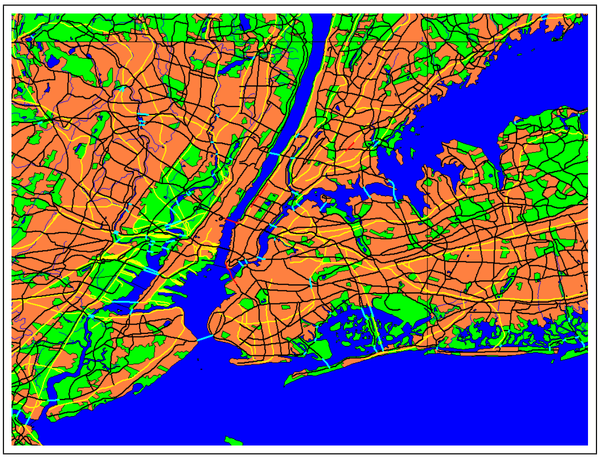

Figure 4.3: VMAP1 transport network, displaying roads and railroads

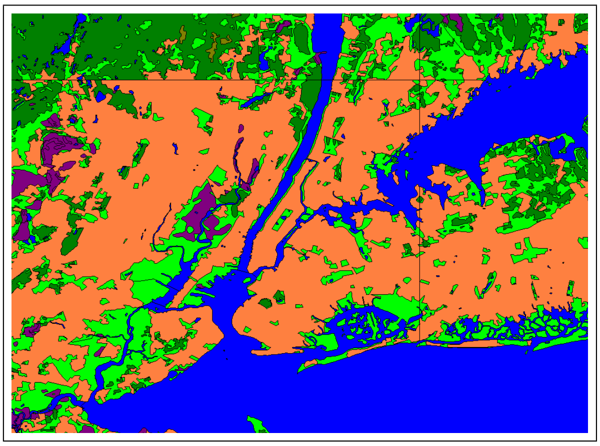

Figure 4.4: VMAP1 coverages, urban, trees, swamp, grass and cropland

4.2.2 VMAP1 Data

VMAP1 [83], [61] is divided into 234 geographic zones. At the present time only 55 selected areas are available. VMAP1 is structural similar to VMAP0, and contains all the standard topographic vector data types familiar to GIS users. The VMAP1 data content includes 10 thematic layers, boundaries, coastlines, road, rail and hydrography to mention a few. Figure 4.4 shows the different coverages, and figure 4.3 shows the road and railroad network in VMAP1. The VMAP1 data is also available from the mapability [54] web page. VMAP1 accuracy can be divided into horizontal and vertical accuracy. VMAP1 product resolution is based on 1:250000 map scale, and the data is also divided into four different classes. The horizontal errors are as follows:

1. 125 m
2. 250 m
3. 500 m
4. bigger than 500 m

While the vertical errors for the four classes are as follows:

1. 0.5
2. 1.0
3. 2.0
4. bigger than 2.0

4.2.3 DNC data

The Digital Nautical Chart (DNC) [60] is a vector based product designed to provide an up-to-date seamless database of the world. It is produced in the standard Vector Product Format (VPF). The features are thematically organized into 12 layers or coverages including: Cultural landmarks, Earth Cover, Environmental, Hydrography, Inland Waterways, Land Cover, Limits, Aids to Navigation, Obstructions, Port Facilities, Relief and Data Quality. The main focus of DNC is on coastline, harbour and near coastline/harbour related information. Figure 4.5 shows some of the information that is available through DNC. There is a light beacon, that typical represents oceanic features, near the lower left corner. DNC data consists of 4 types of data sets, each set having different accuracy. These four data sets are Harbour, Approach, Coastal and General. The Harbour data set is the most accurate, and the General data set is the least accurate. It is worthwhile to note that the DNC Coastal data set is less accurate than the VMAP1 data set. DNC data is available from the NGA page [4]. In this paper we will use DNC Harbour and Approach data. Harbour data has the following horizontal accuracy:

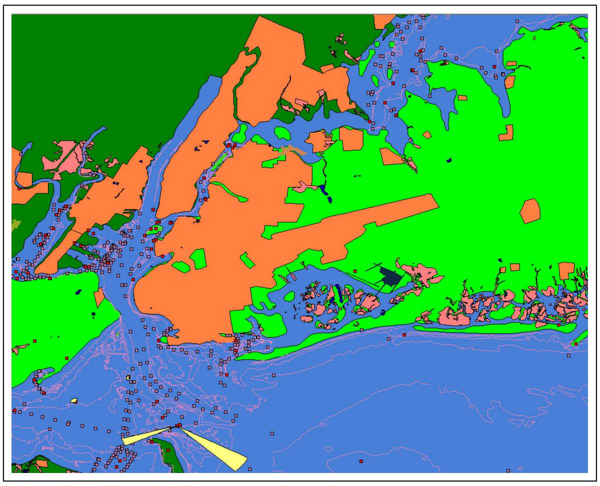

Figure 4.5: DNC coverages, depth contours, danger points and one light beacon(lower left)

1. 25 m
2. 50 m
3. 100 m
4. bigger than 100 m

Approach data is a bit less accurate and has the following horizontal accuracy:

1. 50 m
2. 100 m
3. 200 m
4. bigger than 200 m

4.2.4 Area of interest

New York City is one of the biggest and most renown cities in the world. Our interest of this city comes from the fact that it contains many features typical for a city, like roads, airports, industry and power grids. The geographical placement of the city adds interesting features like islands, rivers and coastlines. The third and most important reason is that there are much data available about the city. VMAP0 covers the whole world and provides data everywhere, whereas VMAP1 and DNC have available data for certain areas. New York City is one of the places covered by all three data sets. All data sets are based on the Vector Product Format (VPF) [14].

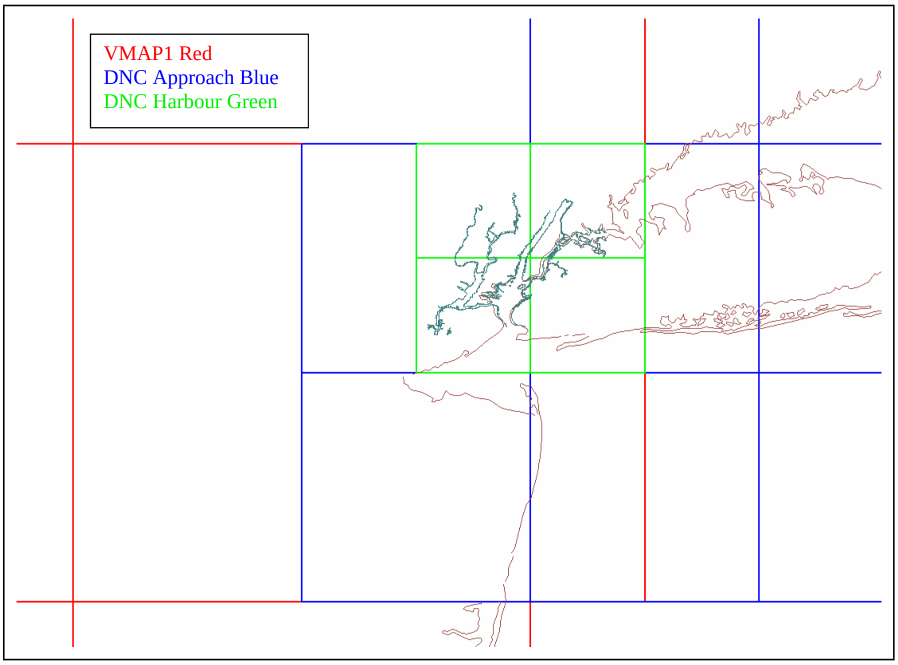

Figure 4.6: The different tile sizes

As earlier mentioned the focus will be on New York City, but it is desirable to geographically define an area of interest. Since each original data set is divided into tiles it might be a possibility to use one of these. VMAP0 tiles are out of the questions since these tiles are humongous in our context. The VMAP1 tiles are smaller, but not small enough to be taken into consideration. The Coastal and General tiles for DNC are also too big, this leaves either the harbour tiles or the approach tiles as an option. The size of the different tiles is shown in figure 4.6. The problems with these tiles are that the harbour tiles are to small, and the approach tiles do not fit the area of interest. This means that it is not possible to use any of the tiles from the data sets, and a new one has to be made. The tile gets the coordinates, Upper left: -74.3, 41, Lower Right: -73.7, 40.4. The tile is shown in figure 4.7, along with the harbour and approach tiles.

4.3 The VMAP1 and DNC Hierarchies

The VMAP1 and DNC data sets are both vector data sets, based on the VPF standard. Due to the fact that VMAP0 and VMAP1 hierarchies are completely equal, VMAP1 has been chosen to represent both. VMAP1 is a general data set describing all aspects of the geography surrounding us, opposed to DNC which is an oceanic data set. The main focus of DNC is on describing phenomenons that are connected with the ocean, and means to make navigation easier. These two completely different areas of intended usage, lead to differences

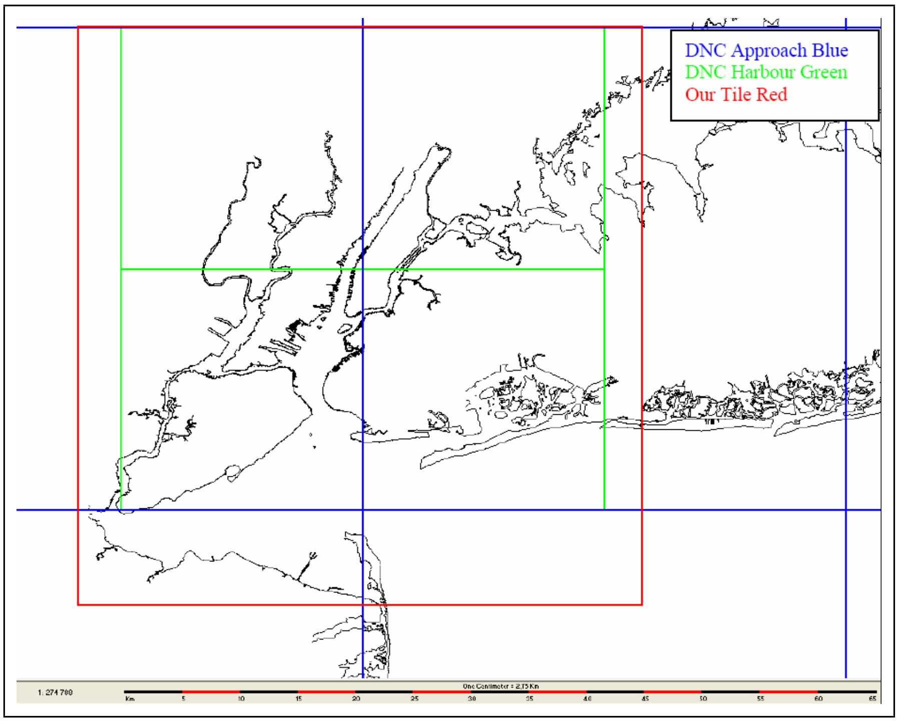

Figure 4.7: The harbour, approach tile, and the tile over the area of interest

in the data set, and differences in how they are organized. This is shown in figure 4.8 and figure 4.9. The VMAP1 hierarchy consists of 9 different coverages. Each Coverage consists of a number of files that contains a number of features. For instance the VMAP1 coverage Hydrography consists of several files, among them Dangers and Lakes/Reservoirs. These files contain one or several types of features. The danger file for instance consists of Rock and Wreck features.

The VMAP1 coverages cover the different aspects of ordinary geography. On one hand there is natural made features like Boundaries, Elevations, Hydrography, Physiography and Vegetation, and on the other hand there is clear man made features like Industry, Population, Transportation and Utilities.

DNC consists of 10 coverages, but with a significant difference from VMAP1. DNC has focused on ocean, or sea related features, and this is reflected in the coverages. Like VMAP1 covers oceanic features, DNC covers on land information as well(though the information is mostly related to near ocean features). Coverages like Cultural Landmarks, Earth Cover and Land Cover have information about on land features, but the majority of coverages are related to the ocean, like Environment, Hydrography, Inland Waterways, Limits, Aids to Navigation, Obstructions and Port Facilities.

As these two data sets describe two different areas of geographic information they have also classified some of the data different. VMAP1 is an overall general data set, but it is more specialized than the DNC data set on land based features like population, industry, physiography, utilities and transportation. While DNC describes most of the hydrography data in a more detailed manner. DNC has taken a general approach on on land features, collecting them in one coverage, for instance like Cultural Landmark. When it comes to oceanic feature, DNC has taken a more specialized approach.

The scenarios in section 4.4 looks into how different problems can be solved when considering making one common hierarchy for both data sets. Small parts of the hierarchy will be used to enlighten different problems and solving techniques.

4.3.1 VMAP1 Hierarchy

The VMAP1 Hierarchy is wide and short. Wide because it has 9 coverages directly under the top node. Short because it usually only takes three steps to reach the bottom of it. An overview of the first two levels is shown in figure 4.8.

First a short note to the VPF organization of data. All the data is collected in a library. Here the library is represented as a "World" node. The data in the Library is then divided into coverages, each coverage consisting of a number of files, and each file consisting of a number of features.

The top node, World, is a collection of all thinkable features in the world. The coverages on the next level are specializations of different real-world phenomena.

- Boundaries: different boundaries, man made and natural made. Examples are political boundaries and coast lines.

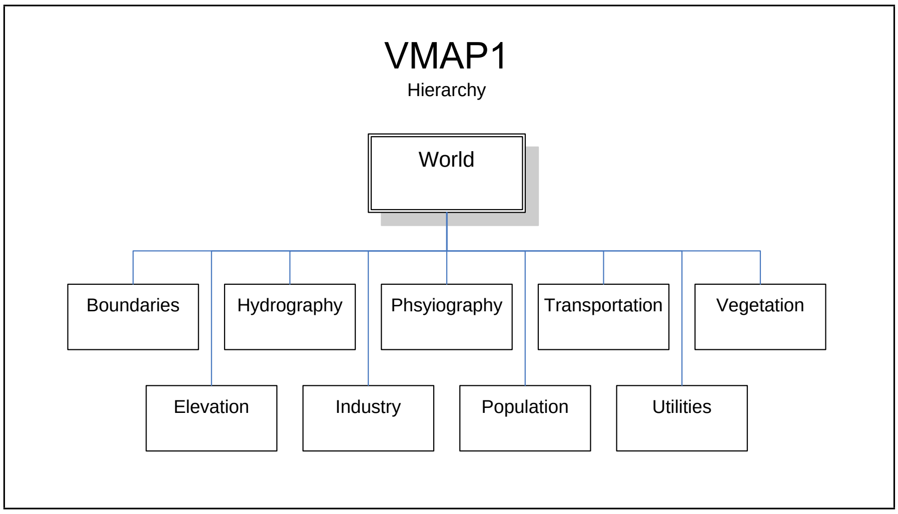

Figure 4.8: The top of the VMAP1 hierarchy

- Elevation: height information about land and depth information about ocean. Examples are contourlines and elevation points.
- Hydrography: hydro information. Examples are danger points, aqueducts, canals, rivers and lakes.
- Industry: industrial information. Examples: extraction points, nuclear installations, storage points, towers and processing installations.
- Physiography: about physical phenomenons like caves, cliffs, embarkment and surface conditions.
- Population: about populations and populated areas. Examples: buildings, built up areas, landmarks, forts and ruins.
- Transportation: about transportation and transportation networks. Examples: airports, railroads, bridges, ferries, tunnels harbours and roads.
- Utilities: about utility systems. Examples: phone network, power network and pipelines.
- Vegetation: about land coverage. Examples: trees, crops, grass, swamps and tundra.

4.3.2 DNC Hierarchy

The DNC hierarchy is, as all VPF hierarchies, wide and short. Wide because it has 10 coverages directly under the top node. Short because it usually only takes three steps to reach the bottom of it. The two upper levels of the DNC hierarchy are shown in figure 4.9.

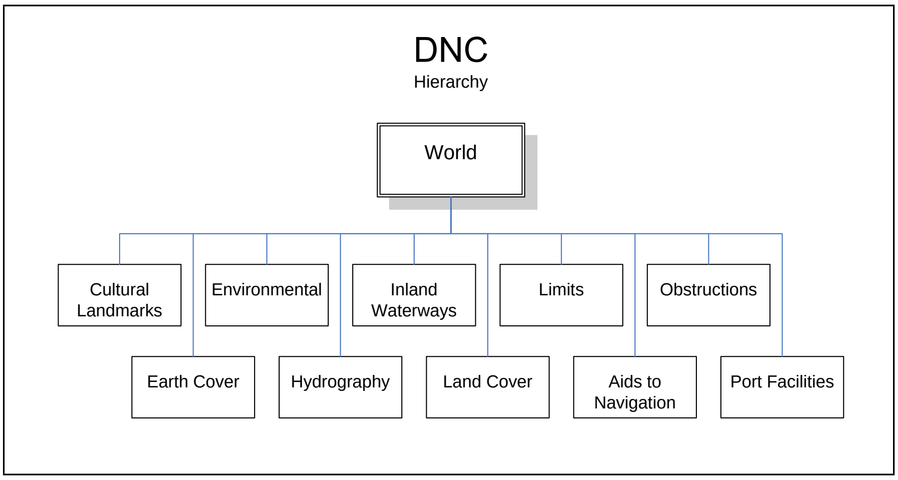

Figure 4.9: The top of the DNC hierarchy

World is the top node and a replacement for the Library definition used in the DNC specification. The 10 coverages on the second level are shortly described below.

- Cultural Landmarks: contains primarily land features of human origin that are significant to marine navigation. Examples: buildings, industrial installations, landmarks, power network, railroads, roads and airports.

- Earth Cover: contains topographic and hydrographic features. Examples: shorelines, islands and boundaries of significance to marine navigation.

- Environmental: contains environmental characteristics of anomalies of significance to marine navigation. Examples: currents and tides.

- Hydrography: contains hydrographic features of significance to marine navigation. Examples: depths and bottom characteristics.

- Inland Waterways: contains inland hydrographic features. Examples: lakes, rivers, canals and dams.

- Land Cover: contains topographic features. Examples: glaciers, trees, swamps and marshes.

- Limits: contains limits. Examples: ferrylines, routes and other separators.

- Aids to Navigation: contains navigational aids. Examples: boys, beacons, and light sectors.

- Obstructions: contains obstructions of significance to marine navigation. Examples: hazards, bridges, tunnels, wrecks and reefs.

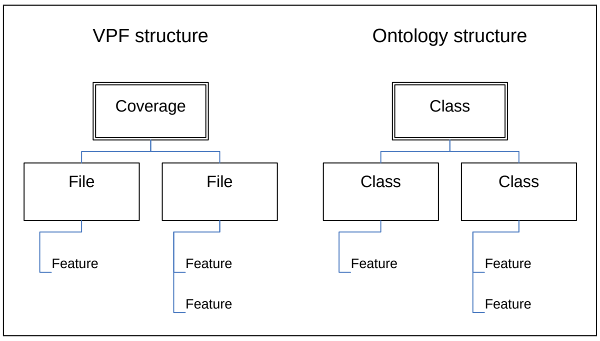

Figure 4.10: The VPF hierarchy structure to the left. A coverage contains a number of files that again contains a number of features. To the right is the ontology conceptualization of the same hierarchy, all coverages and files are classes.

- Port Facilities: contains hydrographic related to ports. Examples: piers, seawalls and breakwaters.

4.3.3 General about Semantic Integration of VMAP1 and DNC Data

Roughly there are three main solutions to the ontology development problem. The first solution is to use the VMAP1 hierarchy, and integrate the DNC hierarchy where it is appropriate. The second solution is to keep the DNC hierarchy and integrate the VMAP1 hierarchy. The third solution is to integrate the two hierarchies, making one single new hierarchy. How and what that should be integrated from the two hierarchies depend on the application and usage of the hierarchy. Based on this one could either integrate single parts, subsets or the whole of the different hierarchies.

The level of detail of the hierarchy is also an important question, but the generality versus specialization also depends on the intended usage and area of application to the ontology.

In the following work the Coverage and File levels will be considered as classes, whereas features will be considered as features, see figure 4.10.

4.4 Scenarios

Three scenarios will enlighten problems and solutions for an integration of classifications from two different sources. The hierarchies from VMAP1 and DNC will be used. (VMAP0 and VMAP1 have similar hierarchies, but VMAP1 contains more features).

The main problem of semantic integration is semantic heterogeneity, which is usually caused by [46]:

1. Different coverage (level of detail) due to different scope - user needs

Table 4.1: Different combinations of term (T) and definition (D) cases. Taken from [46]

	$T_1 = T_2$	$T_1 \neq T_2$
$D_1 = D_2$	equivalence	synonymy
$D_1 > D_2$	additional	IS-A
$D_1 \cap D_2$	overlap	overlap
$D_1 \neq D_2$	homonymy	disjointness

2. Different relations often due to imposition of single inheritance or due to different classification perspectives

3. Different semantics due to different conceptualizations - classification aspects

The first cause can be a problem when dealing with information from different sources and the sources are of different scope. For instance an ontology describing the world would usually classify all buildings as buildings, while an ontology for a city would need to classify the different buildings into narrower terms like hospital, school, fire station and church. This kind of heterogeneity usually deals with the complementary views of the same domain, and therefore does not obstruct the integration process. The second cause of semantic heterogeneity can be overcome by the permission of multiple inheritance in the integrated ontology. A canal can for instance be both a waterway and a man-made feature. The third cause of semantic heterogeneity is the most difficult to identify and tackle. Assume that a real world entity consists of a term T and a definition D, then a concept C is presented by the combination of term and definition, C=(T,D). An overview of possbile cases is shown in table 4.18. The two clearest cases that can be identified are when the term and definition are the same, or when they are completely different. Synonyms occur when concepts are represented by different terms with the same definition. For example the terms "forest" and "wood", both with the definition "A dense growth of trees". Likewise homonymy occurs when concepts are represented by the same terms but with different definitions. In addition to these basic cases there are four more specialized cases. When one of the definition is more descriptive than the other it is either additional information, or an IS-A relation. Overlap occurs when definition overlaps, regardless of the terms.

There will be three scenarios, one for each of the semantic heterogeneity problems. The first problem will be discussed in the Danger point, hazard point scenario. This scenario will present a number of ways to solve a generalization versus a specialization problem. The bridge duality scenario presents a number of possible solutions to the single inheritance, classification problem. Finally the conceptualization problem will be presented through the Monument feature scenario.

4.4.1 Danger points, Hazard points

Danger and Hazard points are classes that refer to dangers in connection with the ocean, lakes or rivers. The Danger, Hazard point scenario shows how one data set consider it a

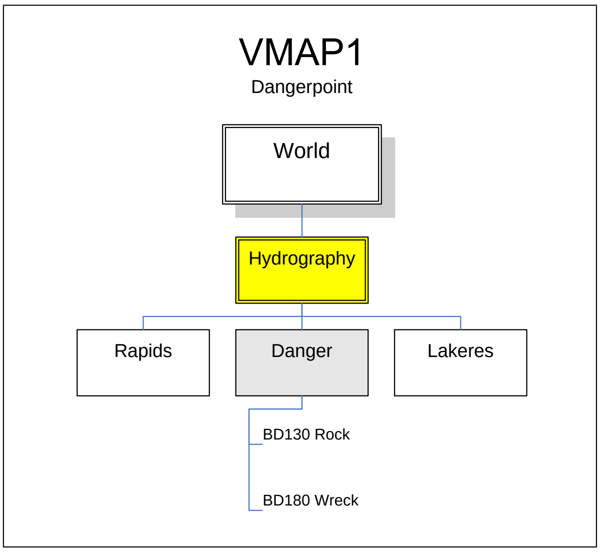

Figure 4.11: The VMAP1 danger point hierarchy

general area, while the other data set take a specialized approach.

In VMAP1 all danger points are collected in one single class, containing two types of dangers, rocks and wrecks. The danger class is a subclass to hydrography that includes all water related classes in the VMAP1 hierarchy, which is shown in figure 4.11.

DNC on the other hand has a more detailed description of danger points, including 8 different danger types. The DNC hierarchy reflects this amount, here the different danger points have been separated into 3 sub classes, as shown in figure 4.12.

Comparing the two hierarchies one notices that the class Danger is included in both, but a further examination of the hierarchy will reveal that the features in the two classes are different. The Danger class in VMAP1 consists of rock and wreck features, while the Danger class in DNC consists of Foul Ground and Pile/Piling/Post. In DNC the rock and wreck features are placed in the Hazard class along with Underwater Danger/Hazard and Crib. The third DNC class, Obstruct, consists of Perches or Stakes and Overfall or Tide Ripes. All in all this reflect the fact that DNC is an oceanic data set, concentration on oceanic phenomenon, and that VMAP1 is a general data set, not specializing on any specific application areas.

The Danger Points Metadata

The main difference between VMAP1 and DNC metadata in this case is the level of information detail. VMAP1 concentrates on the location, material and the placement, this is not enough information to use if trying to navigate through waters. The DNC data on the other hand firstly has an accurate and existence attribute. The accurate attribute describes the accuracy of the geographic position, and the existence attribute determines the state

Table 4.2: The danger attributes for VMAP1

VMAP1			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	BD130 BD180	Rock Wreck
arh	Area Coverage Attribute	-32768 0 <=39	Null Unknown
loc	Location Category	-32768 0 13 14 20 21 28	Null Unknown Hull Showing Masts Showing Funnel Showing Superstructure Shwoing Masts and Funnel Showing
mcc	Material Composition Category	-32768 0 24 84	Null Unknown Coral Rock/Rocky
nam	Name	text UNK	Null Character text string No entry
vrr	Vertical Reference Category	0 1 2 8	Unknown Above Surface/Does not Cover Awash at Sounding Datum Covers and Uncovers

Table 4.3: The danger attributes for DNC danger

DNC Danger Point			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	BD010 BD030 BD050 BD100 BD140	Breakers Discolored Water Foul Ground Pile/Piling/Post Snags/Stumps
acc	Accuracy Category	1 2 3	Accurate Approximate Doubtful
dat	Date	26	Information as of
exs	Location Category	1 2 3	Definite Doubtful Reported
hdi	Hydrographic Depth Info.	-32768 9 10 12 15	Null Depth Known by Other Than Wire Depth Known by Wire Drag Depth Unknown Not Applicable
hdp	Hydrographic Depth	NaN 0.0 0.1 to 12000.0	Null Unknown actual value to the nearest .1 meter
val	Value	0 1 to 32767	Unknown actual value (year)
vrr	Vertical Reference Cat.	-32768 1 4 8	Null Above Surface/Does not Cover Below Surface/Submerged Covers and Uncovers

Table 4.4: The hazard attributes for DNC, part one

DNC Hazard point part 1			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	BD000 BD020 BD130 BD180	Underwater Danger/Hazard Crib Rock Wreck
acc	Accuracy Category	1 2 3	Accurate Approximate Doubtful
cod	Certainty of Delineation	-32768 1 2	Null Limits and Info Known Limits and Info Unknown
dat	Date	26	Information as of
exs	Existence	1 2 3	Definite Doubtful Reported
hdh	Hydrographic Drying Height	NaN 0.0 0.1 to 1000.0	Null Unknown actual value to the nearest .1 meter
hdi	Hydrographic Depth Info	9 10 11 12 13 14 15	Depth Known by Other Than Wire Depth Known by Wire Drag Depth Unknown but Safe Depth Unknown Uncovering Height Known Uncovering Height Unknown Not Applicable
hdp	Hydrographic Depth	NaN 0.0 0.1 to 12000.0	Null Unknown actual value to the nearest .1 meter
loc	Location Category	-32768 4 13 14 20 21 28	Null Below Surface/Submerged/Undergr Hull Showing Masts Showing Funnel Showing Superstructure Showing Masts and Funnel Showing
mcc	Material Composition Cat	-32768 0 24 84	Null Unknown Coral Rock/Rocky
nam	Name	NaN UNK text string	Null Unknown e.g Smith Rock

Table 4.5: The hazard attributes from DNC, part two

DNC Hazard point part 2			
Attribute	*Description*	*Value*	*Value Meaning*
sfc	Sea Floor Feature Category	-32768 1 2 3 4 5	Null Unknown obstruction Other Fish Haven Well Submerged Production Platform
soh	Severity of Hazard	-32768 1 2	Null Dangerous Non-Dangerous
txt	Text Attribute	N/A None text string	Null No textual information
val	Value	0 1 to 32767	Unknown actual value (year)
vrr	Vertical Reference Category	-32768 0 1 2 4 8	Null Unknown Above Surface/Does not Cover Awash at Sounding Datum Below Surface/Submerged Covers and Uncovers

Table 4.6: The danger attributes for VMAP1

DNC Obstruct Point			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	BB105 BB180 BC080 BD040 BD060 BD080	Fishing Harbour Oyster Bed/Mussel Bed Perches/Stakes Eddies Kelp/Seaweed Overfalls/Tide Rips

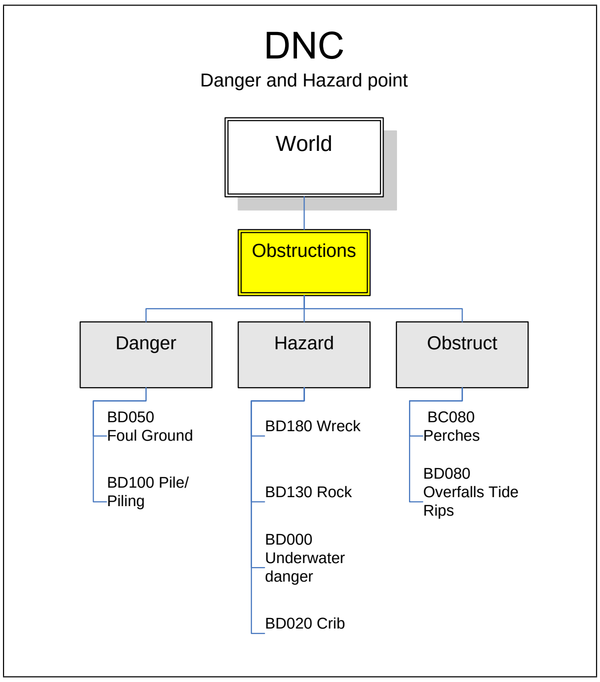

Figure 4.12: The DNC danger point hierarchy

or condition of the feature. In addition to these attributes there are three hydrographic attributes, Drying Height, Depth/Height Information and Depth. The Hydrographic Depth is the depth from the sounding datum down to the top or surface of the feature, and the Hydrographic Depth/Height Information is information about the accuracy of depth or uncovering height of a feature. Hydrographic Drying Height is the height of the feature that tidal water covers and uncovers. And finally the Severity of Hazard tells exactly how dangerous the danger is.

What is evident from the metadata is that DNC is more detailed in the description of metadata on several levels. DNC has 3 classes to cover all the dangers, and has 15 different feature types, in contrast to VMAP1s single class with 2 feature types. With the exception of Obstruct point, the metadata about each different feature is also significantly more detailed in DNC than in VMAP1.

The Semantic Integration

To integrate the danger points from DNC with the danger points in VMAP1 is a question of generalization vs. specialization. Basically the question is whether the solution should be based on the general VMAP1 hierarchy, or the specialized DNC approach.

In the VMAP1 the accuracy and description of oceanic features are not vital, but more meant as a general reference. In DNC the descriptions of oceanic features are important. Hence DNC need much more quality assurance connected to accuracy and data. If you were

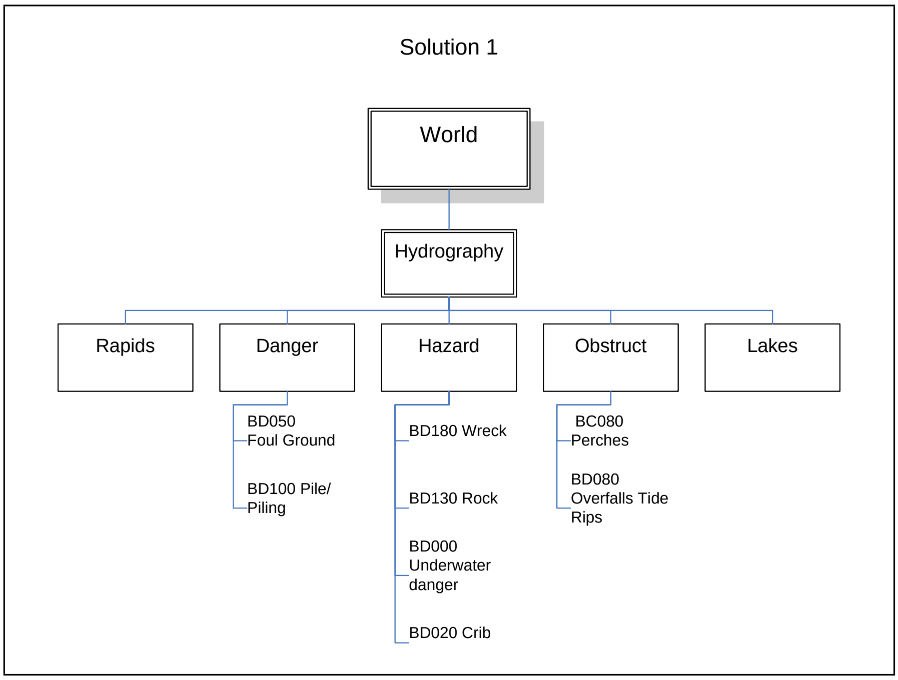

Figure 4.13: The hierarchy for danger point solution 1

a captain of an ocean vessel you could use VMAP1 to figure out your position and roughly find your position according to the coast. But when you were bringing your vessel into a fjord, bay or harbour, you would use the DNC data set to determine where the dangers, light beacons, buoy and currents are.

In the following sections four possible solutions have been outlined. The first solution is based on an integration of the three danger classes from DNC straight into an unchanged VMAP1 hierarchy. The second solution is a simple solution that keeps the DNC hierarchy. The third solution is to integrate the whole DNC hierarchy into the VMAP1 hierarchy as a sub-hierarchy to the hydrography class, and finally the fourth solution is to integrate all the DNC data into one VMAP1 class.

Solution 1 This first solution is to attach the three DNC classes, Danger, Hazards and Obstruct, to the VMAP1 Hydro class. This means that the VMAP1 danger point class will be integrated into the Hazard class. The resulting hierarchy is shown in figure 4.13.

Solution 2 The second solution is based on the DNC hierarchy. We simply keep the DNC hierarchy and add the information about VMAP1 Danger into the Hazards class. This solution does not change anything of the hierarchy, but like all solutions, it affects the attributes to the classes. The solution would be identical with the hierarchy in figure 4.12.

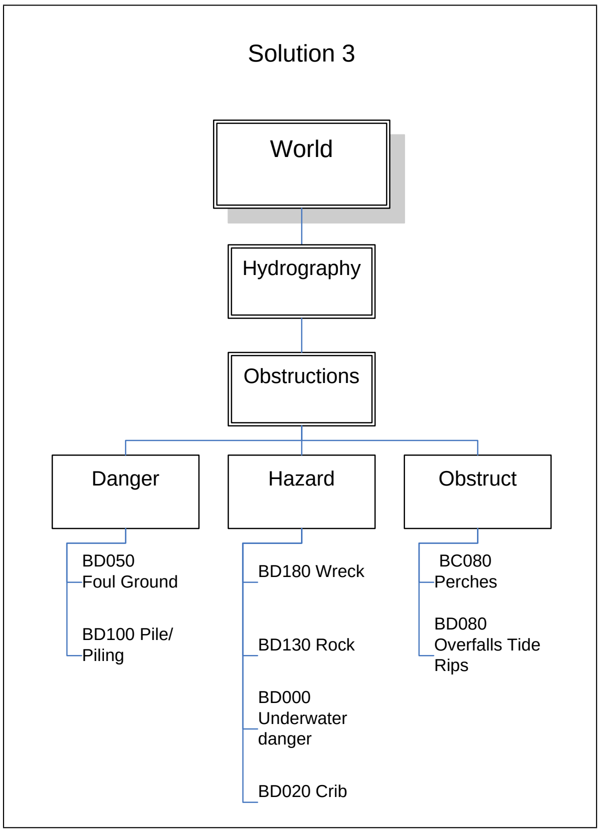

Figure 4.14: The hierarchy for danger point solution 3

Solution 3 The third solution is the only solution that tries to integrate and rebuild the whole structure. The DNC Obstacle sub-hierarchy is integrated as a whole. It is important to emphasize that it is not the whole DNC hierarchy that is integrated, only the classes shown in figure 4.12. This solution would as in solution 1 demand that the VMAP1 Danger class needs to be reclassified and integrated in the DNC Hazards class. The resulting hierarchy is shown in figure 4.14.

Solution 4 The fourth solution is to keep the VMAP1 hierarchy and integrate the DNC data. This solution would end up with one class to represent a diversity of features, and it could be wise to use roles on the class to describe all the features that it represent. It is also a question whether the term Danger is broad enough to encompass all 15 feature types. The concept danger might not be broad enough too encompass all 15 feature types, as a solution to this a new concept might be necessary.

Discussion of the Solutions The way that the two data sets arrange and classify data connected with oceanic dangers reflects their intended use. Which solution to choose relates to the settings, needs and requirements along with the intended usage of the ontology.

A specialized approach allows for a securer and more accurate classification of new data. The specialized approach also makes it easier to reduce time in connection with a peer review process, since it would be very clear where to place new data. The negative side of a specialized hierarchy is that in the event uncertainty arises, a peer review process could be time consuming, since a change would affect many other features.

The general approach gives a faster classification of new data. It is not necessary to think if this feature is suitable in this or that class, just put it in the big bulk. The negative side of a general approach, is that it can become too general, especially when trying to find very specific data. A general approach could also lead to vast amount of data in a few classes, making it difficult to do searches.

Solution 1, 2 and 3 all opt for the specialized approach. Solution 2 and 4 lead to least changes in the hierarchies, and simply refer to the process of integrating or dividing all properties to the appropriate class. Solution 1 and 3 perform changes on the hierarchy, but the result is different. In the first solution, danger, hazard and obstructions are all placed directly beneath the hydrography class. The third solution combines the generalization and specialization. It keeps the general obstruction class from DNC and attaches this to the even more general hydrography class from VMAP1, but in the process it keeps the specialization from DNC.

The generalization/specialization problem is a question of user needs, and level of detail. This kind of problem does not include conflicting but complementary views of the same domain [46], making this problem relatively easy to handle.

4.4.2 The Bridge Duality

The VMAP and DNC data sets refer to transportation data in different ways. VMAP data usually refer to transportation as moving items from point a to b using roads and railroads, whereas DNC mainly focus on transportation from an oceanic perspective. Bridges are an important asset for transportation by road and railroad, while bridges are not needed the same way by e.g. ships. The differences in classification is reflected in the hierarchies.

VMAP0 does not have bridges included as features in its data set. VMAP1 interprets bridges as a part of the transportation system, that is as part of a road, railroad, pedestrian trail or canal. As can bee seen in figure 4.15 VMAP1 has positioned bridges under the Transportation coverage, making bridges a part of the transportation network. DNC on the other hand, interprets bridges as obstacles. Figure 4.16 indicates the bridge placement in the DNC hierarchy. The bridge is placed beneath the Obstacles coverage, and thereby is considered as one of several obstacles.

The Bridge Metadata

The VMAP1 bridge metadata is summarized in table 4.7, and the DNC bridge metadata is summarized in table 4.8. Table 4.9 compares the meta-data from VMAP1 and the different DNC data sets. VMAP1 has information that is related to a transportation scenario,

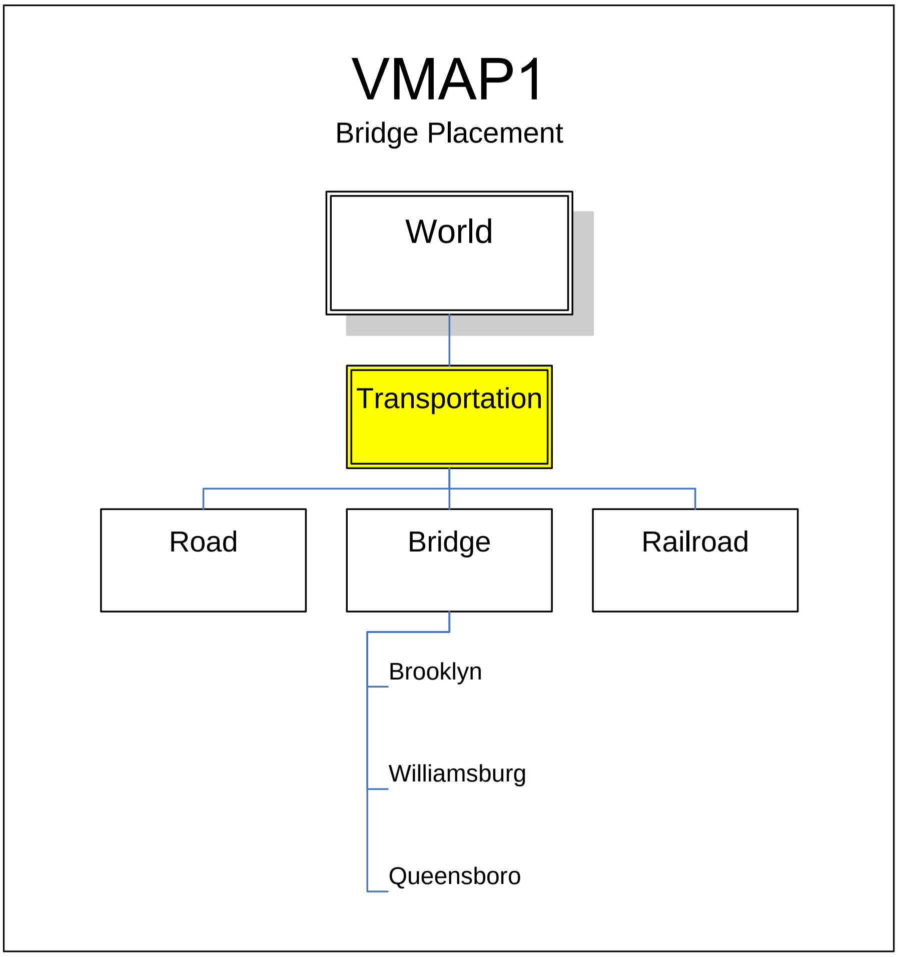

Figure 4.15: The VMAP1 bridge hierarchy

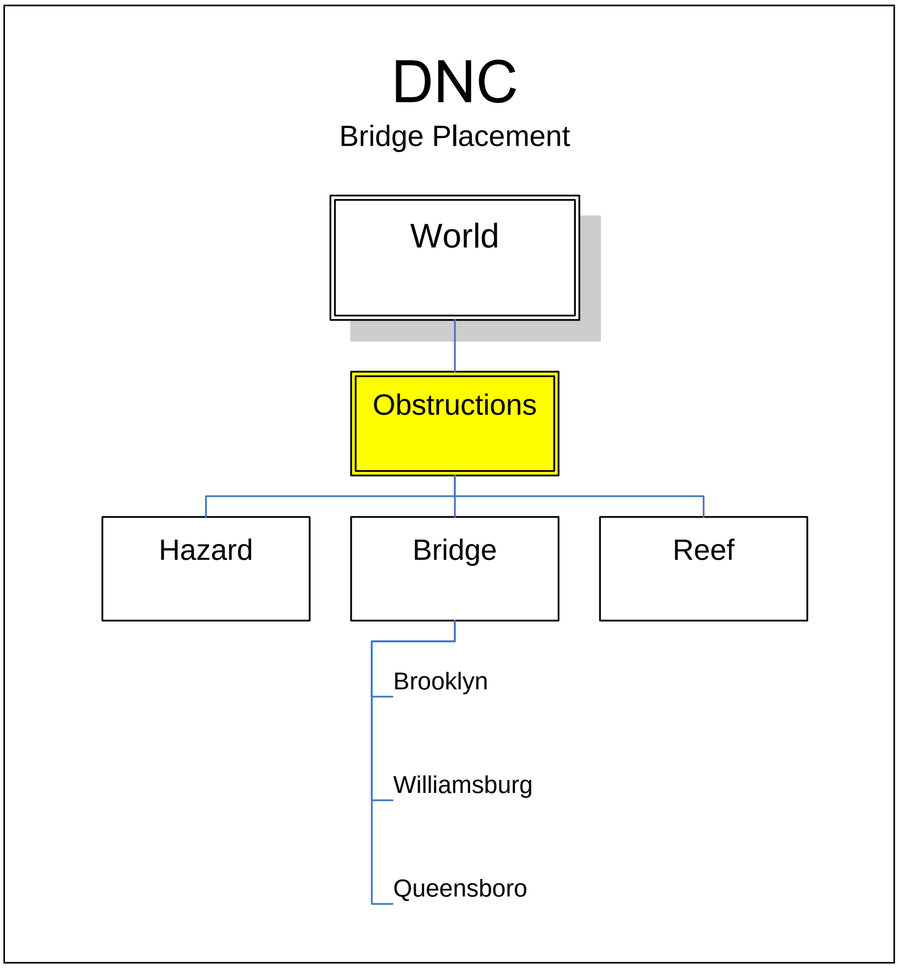

Figure 4.16: The DNC bridge hierarchy

such as the length of the bridge and the Transportation Use Category. The two least accurate DNC data sets, General and Coastal do not even include the Transportation Use Category attribute. The DNC attributes, Maximum Vertical Clearance, Overhead Clearance Category, Over Water Obstruction and Safe Horizontal Clearance, all describe information about the bridge that is necessary to know in order to pass safely under or through the bridge, and this information reflects the fact that a bridge is viewed as an obstacle.

The Semantic Integration

Semantically the problem arises when one tries to integrate the different data sets. According to VMAP1 and DNC, bridges is either part of the transportation network, or an obstacle. There is no answer to which view is correct, but the domain of the application should decide how to organize the data, and where bridges should be placed in the context.

In the following paragraphs several different solutions are suggested on how to integrate bridge data. Solution 1 and Solution 2 are very similar, and Solution 3 and Solution 4 are very similar. The main difference is that solution 1 and solution 3 are based on the VMAP1 hierarchy, while solution 2 and solution 4 are based on the DNC hierarchy. Solution 5 introduces new classes that simplify the placements of the bridge class. The two last solutions suggest two alternative ways for a bridge class to take on two different roles. Solution 6 uses multiple inheritance to achieve this, whereas solution 7 suggests that the class itself contains information about the roles it has.

Solution 1 In this solution, the result hierarchy is based on the VMAP1 hierarchy from figure 4.15. The semantic integration will consist of integrating the instances from the DNC data, and adding the DNC attributes into the ontology. The resulting ontology is completely similar to the already existing VMAP1 hierarchy.

Solution 2 Solution 2 is based on the DNC hierarchy, and the resulting ontology is exactly similar to the original DNC hierarchy 4.16.

Solution 3 The two preceding solutions are both based on minimal manipulation of the already existing hierarchy. The two next solutions have made small changes to the hierarchy structure.

This solution has integrated the DNC structure into the VMAP1 hierarchy, by placing the Obstruction class beneath Transportation. The hierarchy now have one more level, as shown in figure 4.17, and the VMAP1 bridge features are placed in their new position.

This solution means that all bridges now will be viewed as an obstacle in the transportation network. Is this valid, can all bridges be viewed as obstacles. The answer is clearly no. A bridge can also be an obstacle on land, just think about low railroad or road bridges, but some of the bridges, like one over a canyon, are clearly not obstacles, on the contrary they improve the transportation situation.

Table 4.7: The bridge attributes for VMAP1

VMAP1			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	AQ040	Bridge/Overpass/Viaduct
bdc	Bridge Design Category	0 5 6 7 8 9 11 12	Unknown Floating Brige/Pontoon Girder Stringer Truss Suspension Other Transporter
bot	Bridge Opening Type	0 4 10 11 12 13	Unknown Draw/Bascule Swing Lift Retractible Not Applicable
bsc	Bridge Superstructure Category	0 2 7 8 17	Unknown Cantilever Tower Suspension Truss Arch Suspension
exs	Existence Category	0 1 2 3	Unknown Definite Doubtful Reported
len	Length/Diameter(meters)	0 >= 125	Unknown
ohb	Overall Height of Bridge(meters)	0 1	Unknown to no upper limit
tuc	Transportation Use Category	0 1 3 4 17 38	Unknown Both Road and Railroad Railroad Road Pedestrian Canal
zv2	Highest Z-value(meters)	29999 -400	Unknown to 11999

Table 4.8: The bridge attributes from DNC

DNC			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	AQ040	Bridge/Overpass/Viaduct
bot	Bridge Opening Type	0 4 10 11 12 13	Unknown Draw/Bascule Swing Lift Retractible Not Applicable
bsc	Bridge Superstructure Category	0 1 2 3 5 6 8 9 12 15 16 999	Unknown Arch Cantilever Deck Floating Bridge/Pontoon Girder Truss Suspension Transporter Slab Stringer(beam) Other
dat	Date	26	Information as of
exs	Existence Category	5 28	Under Construction Operational
mvc	Maximum Vertical Clearance	0.0 0.1 to 2000000.0	Unknown actual value to nearest .1 meter
nam	Name	UNK text string	Unknown (e.g., "Brooklyn Bridge")
ohc	Overhead Clearance Category	0.0 0.1 to 998.0	Unknown actual value to nearest .1 meter
owo	Over Water Obstruction	1	Feature Crosses Navigable Water
shc	Safe Horizontal Clearance	0.0 0.1 to 1000.0	Unknown actual value to nearest .1 meter
tuc	Transportation Use Category	0 1 3 4 17	Unknown Both Road and Railroad Railroad Road Pedestrian
val	Value	0 1 to 32767	Unknown actual value (year)

Table 4.9: Comparison of the bridge attributes to the different data sets

Comparison of different metadata														
Name	*bdc*	*bot*	*bsc*	*exs*	*len*	*ohb*	*tuc*	*zv2*	*dat*	*nam*	*ohc*	*owo*	*shc*	*val*
VMAP1	X	X	X	X	X	X	X	X						
General		X	X	X					X		X	X	X	X
Coastal		X	X	X					X		X	X	X	X
Approach		X	X	X			X		X	X	X	X	X	X
Harbour		X	X	X			X		X	X	X	X	X	X

A classification like this one would mean that a bridge is an obstacle, and not an integrated part of the transportation system. On one hand the bridge is part of the transportation system, but it is also an obstacle. This means that a bridge is either an obstacle in general, or it is an obstacle within the transportation system. According to most definitions of hierarchies, the latter is most correct.

Solution 4 The fourth solution is also an integration of the two hierarchies, this time based on the DNC hierarchy. The result is shown in figure 4.17. This solution would also mean some work as for to reclassify, or reposition the DNC bridges.

Here all bridges are viewed as part of the transportation system, but the whole transportation system is viewed as an obstacle. That means that roads, railroads, bridges and tunnels, among others, are classified as obstacles. In this setting one can argue as above, whether bridges should be viewed as a part of the transportation system, or as an obstacle.

Solution 5 In this solution a new class is introduced. To separate the transportation on land from the transportation at ocean, two new classes are introduced. This way it becomes clear that a bridge is a part of the land-based transportation network. This solution is mainly based on the VMAP1 hierarchy and the VMAP1 definition of bridges and is shown in figure 4.18.

Solution 6 The solution shows a hierarchy where a bridge is defined as both an obstacle and a part of the transportation network, see figure 4.18. This solution allows multiple inheritance as suggested by Kavouras [46], making it possible for classes to take on multiple roles or meanings. Multiple inheritance can cause some confusing situations, and the debate is whether or not its benefits outweigh its risks. This debate will not be followed up here, for further information see [13].

Solution 7 The final solution is not concerning the hierarchy but the information embedded in classes. Thesauruses have a function to relate to terms to each other regardless of hierarchical position. This is called Related Term or RT as described in section 2.2. In

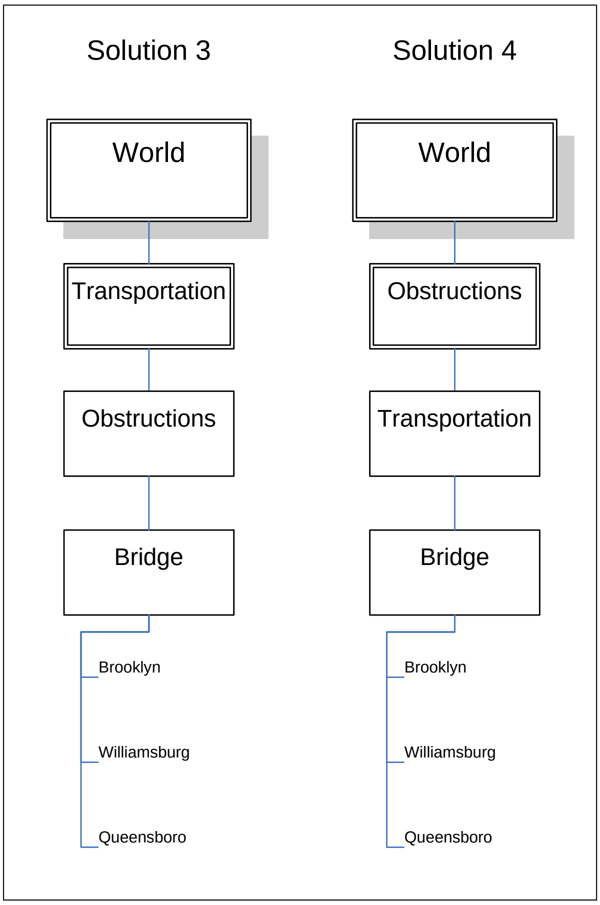

Figure 4.17: The third suggested solution on bridge duality scenario

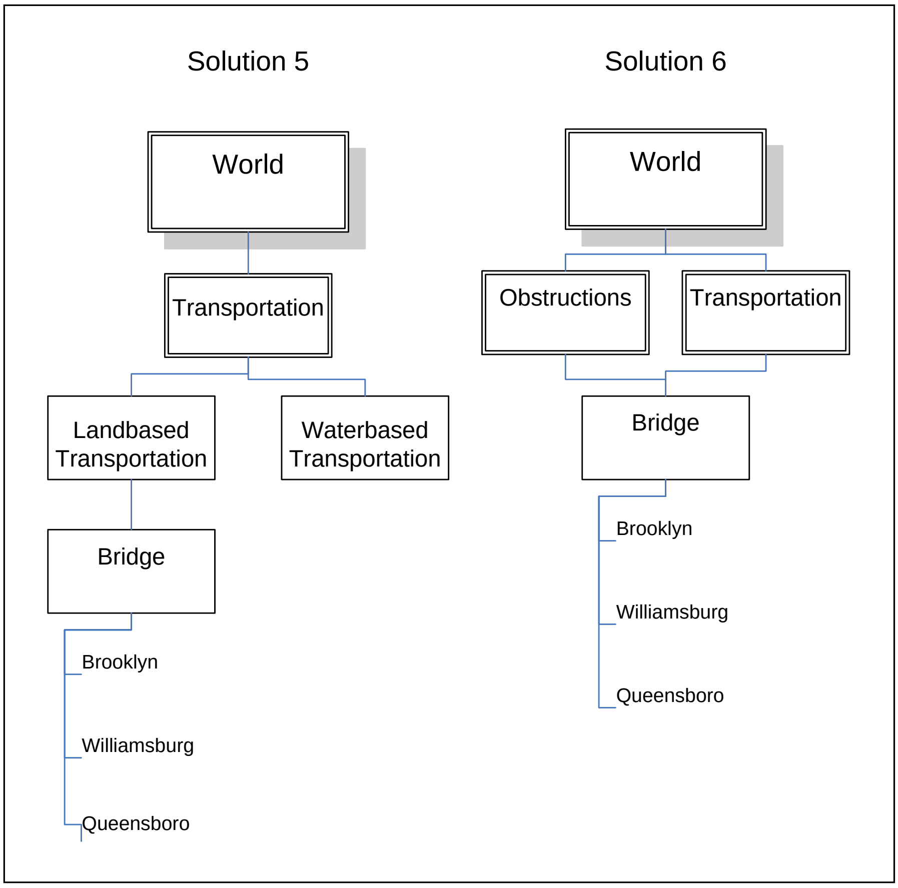

Figure 4.18: The fifth and sixth solution. The fifth solution introduces two new classes, one for land-based transportation, and one for water-based transportation. The sixth solution allows multiple inheritance.

fact such an approach has already been suggested by Fonseca et al. [25]. However they do not use the term related, but give a class additional *roles*. Meaning that a class has one placement in the hierarchy, but several different roles.

In the problem raised here a bridge could be classified as a part of the transportation network, but it would have an additional role as an obstacle.

Summary of the solutions The Bridge Duality problem can be viewed in two ways. It is mainly a classification problem due to different classification perspectives, but it can also be viewed as a conceptualization problem. In a conceptualization problem, the term Bridge, would have different or slightly different definitions in DNC and VMAP1 making it a homonymy or overlap problem. The definition of the bridge class is elaborated on later.

Of all the solutions, solution 1 and solution 2 look most suitable. Both solutions would not lead to changes in the hierarchy, they would only lead to an expansion of the bridge class in the means of properties. Solution 3 and 4 do not offer good solutions to the problem, although solution 4 is better than solution 3. A bridge that is part of the transportation network is not an obstacle, but a bridge can be part of the transportation system which again can be viewed as an obstacle or obstruction. Solution 5 presents a refinement of the hierarchy, making a clearer statement that bridges are a part of land-based transportation. There are some application areas where this might be a plausible solution. Solution 6 solves the problem by using multiple inheritance, which can be a good solution, but in the context of several users it might be too complex. Solution 7 can be combined with all solutions mentioned above, except solution 6, where it would become redundant information. To add extra descriptive information might help to clarify uncertainty, and would be important in a big and complex ontology.

All the above solutions can be viewed as correct. Which solution to use depends on the application and usage of the ontology.

4.4.3 Monuments, Industrial Installations or Landmarks?

The last scenario presents a conceptualization problem. VMAP1 classifies monuments as landmarks, while DNC classifies monuments as industry, or industrial installation. The different classifications are shown in figure 4.19 and 4.20.

Section 4.4 established that a concept consists of a term T and a definition D. In this scenario the two concepts are Landmark and Industry. The term is Monument, but the definitions are different. VMAP1 and DNC do not provide definition for these concepts. Instead the definitions are collected from different online categorization tools, like GEMET, WordNet and ADL Gazetteer. The definitions for industry and landmark are shown in table 4.10. A Monument is defined as "The position of a prominent or well-known object in a particular landscape." in VMAP1, and "The organized action of making goods and services for sale" in DNC. The definitions are clearly different, and according to table 4.18 the problem is one of homonymy.

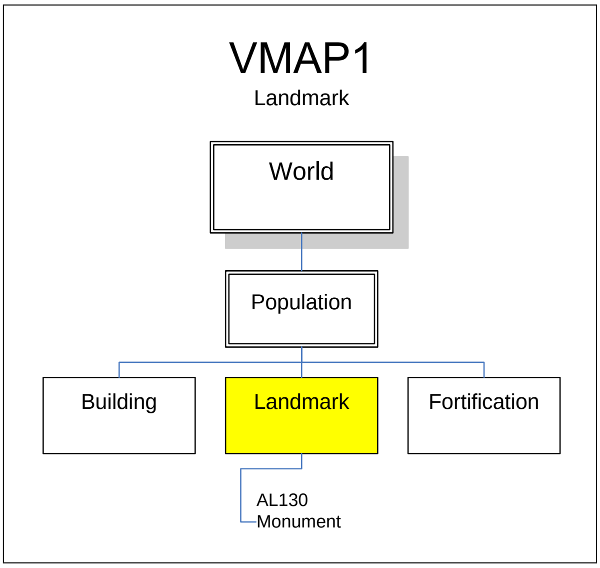

Figure 4.19: The placement of the concept monument in the VMAP1 hierarchy

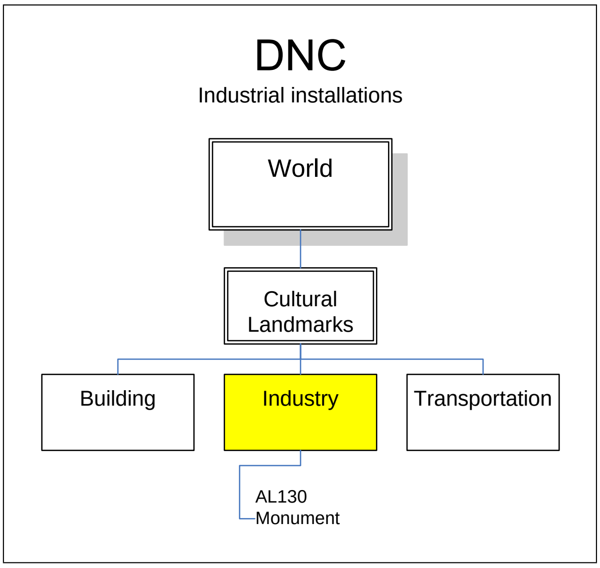

Figure 4.20: The hierarchy for industrial installations in DNC

Table 4.10: Definition of industry and landmark, according to WordNet, GEMET and ADL Gazetteer.

Definition of Industry and Landmark			
Term	*WordNet*	*GEMET*	*ADL Gazetteer*
Industry	The organized action of making of goods and services for sale.	An industry is a group of establishments engaged in the same or similar kinds of economic activities.	Groups of associated structures functioning as a unit used for refining a material of manufacturing a product
Landmark	The position of a prominent or well-known object in a particular landscape.		N/A, but Landmark points directly to monuments.

The Monument Metadata

Table 4.12 and 4.13 display all the metadata attributes for DNC. The VMAP1 landmark attributes can be found in table 4.11. A comparison between the two data sets can be found in table 4.14.

It is a bit surprisingly to see that there are few distinctions between the two data sets. DNC contains all except one of the VMAP1 attributes. DNC contains more attributes, but this is partly due to the fact that the class embraces a wider variety of features. The additional attributes are also oriented towards the industrial domain.

The Semantic Integration

The integration process opts for several different solutions. Two obvious solutions are to base the ontology on either the DNC hierarchy or the VMAP1 hierarchy and integrate the remaining data accordingly.

The following paragraphs present a number of different solutions. Solution 1 bases itself on the VMAP1 hierarchy, solution 2 is based on the DNC hierarchy. The three last solutions give the monument feature additional meaning, by using multiple inheritance and related terms.

Solution 1 Presents a solution based on the VMAP1 hierarchy as in figure 4.19. The different features in the DNC Industry class are divided and classified among the appropriate VMAP1 classes. Monument ends up in the class Landmark.

Table 4.11: The Landmark metadata for VMAP1

VMAP1 Landmark			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	AK020 AK150 AK160 AL130	Amusement Park Attr Ski Jump Stadium/Amphitheater Monument
exs	Existence Category	0 1 2 3	Unknown Definite Doubtful Reported
hgt	Height Above Surface Level	0 > 1 >= 46	Unknown
nam	Name	N/A UNK text	Null Unknown text string
ssc	Structure Shape Category	-32768 0 12 17 21 23 25 77 109 999	Null Unknown Pyramid Spherical Artificial Mountain Ferris Wheel Roller Coaster Arch Obelisk Other
zv2	Highest Z-value (meters)	29999 -400	Unknown to 11999

Table 4.12: The industrial points in DNC

DNC Industrial			
Attribute	*Description*	*Value*	*Value Meaning*
id	Identifier	Sequential	
f code	FACC Feature Code	AF010 AF040 AF070 AH050 AJ050 AK020 AL130 AL240 AM070 AQ060 AQ080 AT045	Chimney/Smokestack Crane Flare Pipe Fortification Windmill Amusement Park Monument Tower Tank Control Tower Ferry Site Radar Transmitter
acc	Accuracy Category	1 2	Accurate Approximate
col	Character of Light	N/A UNK text	Null Unknown text string
hgt	Height Above Surface Level	-2147483648 0 1 to	Null Unknown 2147483647
loc	Location Category	-32768 8 22	Null On Ground Surface Offshore
nam	Name	N/A UNK text	Null Unknown text string
pro	Product Category	-32768 0 31 999	Null Unknown Electric Other

Table 4.13: The industrial points of DNC, continous

DNC Industrial continuous			
Attribute	*Description*	*Value*	*Value Meaning*
ssc	Structure Shape Category	-32768	Null
		0	Unknown
		12	Pyramid
		17	Spherical
		21	Artificial Mountain
		23	Ferris Wheel
		60	Mast
		77	Arch
		87	Dome
		107	Tower
		108	Scanner
		109	Obelisk
		999	Other
tuc	Transportation Use Category	-32768	Null
		0	Unknown
		1	Both Road and Railroad
		3	Railroad
		4	Road
		12	Marine
		13	Air
		17	Pedestrian
use	Usage	-32768	Null
		132	Container
		999	other
zv2	Highest Z-value	-2147483648	Null
		29999	Unknown
		-400	to 11999

Table 4.14: Comparison of the landmark attributes from VMAP1 and the industrial attributes from DNC. Identifiers and FACC codes has been left out.

Comparison of VMAP1 landmark and DNC industrial metadata											
Name	*exs*	*hgt*	*nam*	*ssc*	*zv2*	*acc*	*col*	*loc*	*pro*	*tuc*	*use*
VMAP1	X	X	X	X	X	X	X				
DNC		X	X	X	X	X	X	X	X	X	X

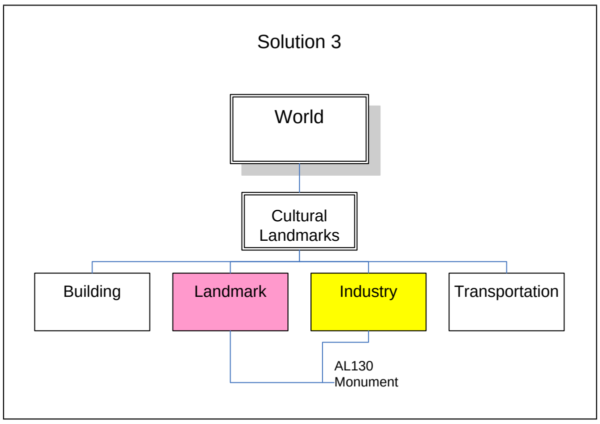

Figure 4.21: Monument solution 3, multiple inheritance. Based on the DNC hierarchy, the landmark class has been integrated.

Solution 2 Solution 2 uses the DNC hierarchy as its starting point as in figure 4.20. Features from the landmark class are categorized and placed in the correct classes. Monument ends up in the class Industry.

Solution 3 This solution uses multiple inheritance to avoid the problem of deciding where to put the Monument feature. The feature is given multiple roles or meanings. The solution is shown in figure 4.21.

Solution 4 With two different concepts, a solution could be to introduce a parent class that embeds both concepts. The Monument feature could be moved up a level in the hierarchy. This would make the definition more general, and the monument feature classified as Culture Landmark as shown in figure 4.22. The example given here, is of course an example. One could for instance have made a completely new class instead of using the Cultural Landmark class.

Solution 5 The fifth and final solution does not need to manipulate the hierarchy. By giving the Monument feature several related roles, meaning that the feature has one place in the hierarchy, but can have several different roles. This solution is similar to solution 7 in section 4.4.2.

Summary of the solutions The kind of semantic heterogeneity that has been described in this scenario is difficult to solve. There are no obvious way in which the two hierarchies can be merged such that a Monument feature can be represented by a concept that represents both conceptualizations. The two first solutions represent the feature is *either* a landmark or an industry. The three last solutions try to merge the two conceptualization and give the Monument feature several roles, or meanings.

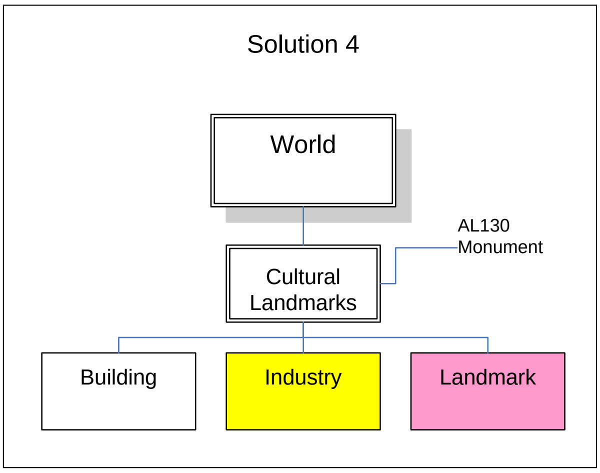

Figure 4.22: Monument solution 4. The hierarchy is based on the DNC hierarchy. The Landmark class has been integrated. Instead of classifying the monument feature in either class, it is moved up a level and classified as a Culture Landmark

None of these solutions are more correct than the other ones. Which solution to choose depends on the application, domain and the users conceptualization of a monument.

4.5 Full design of a Scenario

In this section one of the scenarios above will be used to fully implement an ontology. The ontology will be a simple one, containing mainly classes and properties.

The meaning of an ontology is that it should represent a domain, and in doing so provide a common base, on which several actors can communicate and understand each other. The ontology arranges everything in a system with the use of classes in hierarchy. Sometimes a class name is inadequate to classify features. In such situations, further information about the classes could be useful. In the following sections we will see how metadata can provide attributes, or properties, that can help to further describe classes in the ontology.

4.5.1 Properties

A property[1] can be seen as the attributes to the class. A property is not the same as a metadata attribute for a feature. This means that some properties will represent attributes as they are, while other properties can be an universal word for several different attributes. Consider for instance an area attribute. This could be named area in one feature, but size or space in other features. The property Name could be any of these three, depending on

[1]The properties of classes is sometimes referred to as slots [23], but with the W3C OWL language Recommendation in February 2003, properties have been established as a standard.

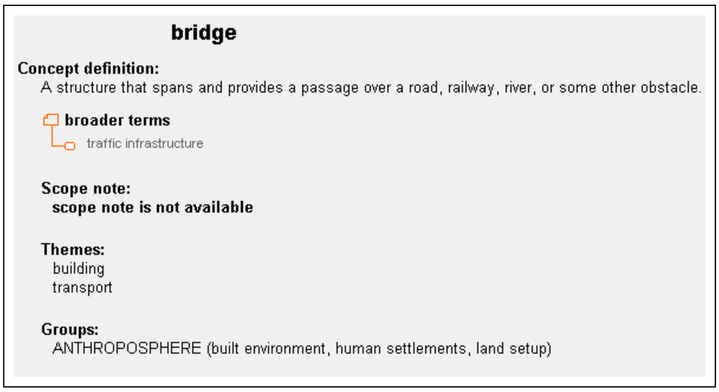

Figure 4.23: The GEMET definition of a Bridge

what the ontology designers decide on. It is important to remember that the *property* area differs from the *attribute* area.

Properties describe the internal structure of concepts, a concept here being a class in the ontology. Attributes, become properties attached to classes. Each property has one or several facets. Facets describe the value type(String, Number, Boolean or Enumerated), allowed values and number of values(cardinality). Two more issues to take into consideration are the range and domain of a property. The range of a property is the classes that can have the property. Example here is that Transportation Use Category can be valid for several classes, for instance bridge or tunnel. The Domain of a property is represented by the class the property is attached to. For example the domain for the Bridge Construction Category is Bridge.

The ontology can be viewed as a hierarchy with classes, relations and sub-classes, where a sub-class is the specialization of a class. This is very similar to thoughts one find in object oriented programming. Another quality taken from object oriented programming is inheritance. All sub-classes inherit the properties from their parent. The domain of a property should be defined in the most general class.

4.5.2 The Bridge Duality

Before the properties of the Bridge class can be resolved, a decision on which of the seven solutions to use has to be taken. According to GEneral Multilingual Environment Thesaurus (GEMET) [72], a bridge is defined as "A structure that spans and provides a passage over a road, railway, river, or some other obstacle.", see figure 4.23. Further the figure implies that bridge is considered a part of the traffic infrastructure. According to the DIGEST specification [19], a bridge is defined as "A man-made structure spanning and providing passage over a body of water, depression, or other obstacles". A third definition is given by the Alexandra Digital Library Feature Type Thesaurus(ADL Feature Type Thesaurus, University of California), "Structures erected over a depression or obstacle to carry traffic." see figure 4.24. Based on these three definitions we here chose solution 1, see 4.4.2, to

Table 4.15: Definition of the different classes, according to DIGEST, GEMET and ADL Gazetteer.

Definition of the different classes			
Class Name	*DIGEST*	*GEMET*	*ADL Gazetteer*
World		The Earth with all its inhabitants and all things upon it	
Transportation		The act or means of moving tangible objects from place to place, system of lines of movement or communication by road, rail, water or air	
Bridge	A man-made structure spanning and providing passage over a body of water, depression or other obstacles	A structure that spans and provides a passage over a road, railway, river or some other obstacle	Structure erected over a depression or obstacle to carry traffic

integrate the DNC bridge data into the VMAP1 hierarchy.

Table 4.15 shows the definition of the three main classes. The definitions are taken from three different sources, the GEMET thesaurus, the DIGEST standard and the ADL Gazetteer. These definitions can be used when talking to other people about the concepts as a reference and making communication or integration in a peer-review environment easier.

Step 1 the existing VMAP1 hierarchy, with its properties

The VMAP1 hierarchy is shown in figure 4.15. This will be the starting point. From this a fully integrated ontology with properties will evolve. The first step is to look at the VMAP1 Bridge class, but this time with properties. In this first simple model, see figure 4.25, the properties are simply the attributes to a VMAP1 Bridge feature, as shown in table 4.7. The properties should in some way describe the internal structure of concepts. Since a sub-class inherits all properties from its parent, all attributes will be subjected to an analysis. The analysis will decide where in the hierarchy to place the attributes. An attribute should be attached to the most general class that can have that property. The properties Bridge Design Category, Bridge Opening Type, Bridge Superstructure Category and the Overall Height of Bridge are most certain unique for Bridge features. All the other properties are of a more general type. FACC Feature Code, Existence Category, Length/Diameter and Highest Z-value are all put at the top of the ontology in the world class. The Transportation Use Category is placed in the Transportation class.

In order to have something more specific to refer to, table 4.16 shows how the DIGEST

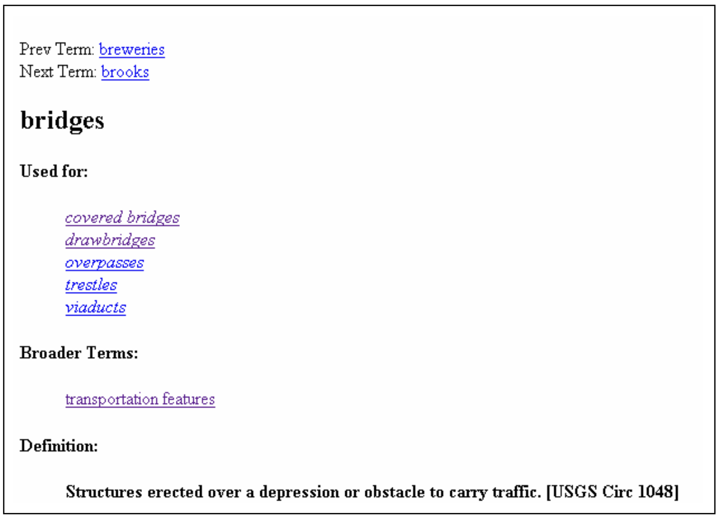

Figure 4.24: The ADL Feature Type Thesaurus definition of a Bridge

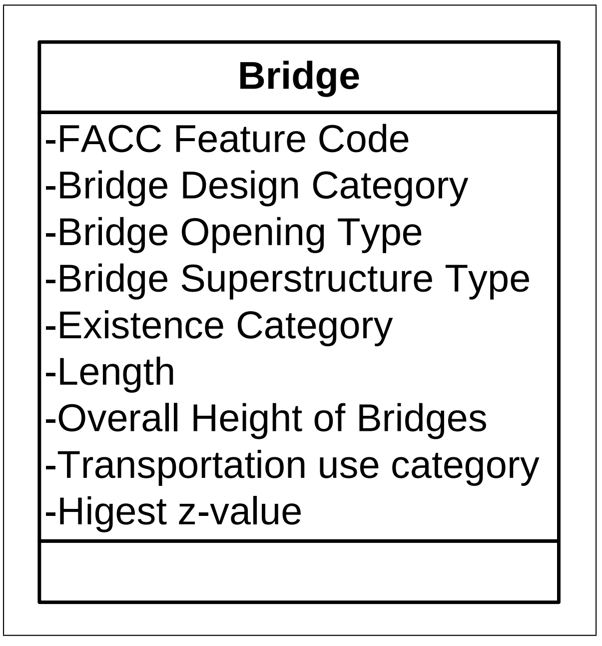

Figure 4.25: The VMAP1 Bridge class with attributes as slots

Table 4.16: Definition of the different attributes according to the DIGEST standard

Definition of VMAP1 attributes	
Attribute	*DIGEST definition*
Bridge Design Category	Structural design characteristics of the bridge or a bridge segment.
Bridge Opening Type	The type of structure or mechanism by which a portion of a bridge is moved to allow passage of a vessel.
Bridge Superstructure Type	Structural design characteristics.
Existence Category	State or condition of the feature.
Lenght	A measurement of the longer of two linear axes in meters. For a round feature, measure the diameter. For a bridge, the length is the distance between the bridge abutments.
Overall Height of Bridges	Vertical distance measured from the lowest point to at ground or water level to the highest portion of bridge.
Transportation use Category	Identifies the primary user, function, or authority of the transportation system.
Highest Z-Value	Elevation above a given datum to the highest portion of the feature.

standard has defined the different attributes. This information is used to easier understand where a property is best placed in an ontology.

Figure 4.26 displays how the properties are placed in the ontology. This solution is primarily based on the attribute definitions. The most uncertain placement was that of the length attribute. A thing to take into consideration here is that it is not *necessary* for a feature to have all properties. A new feature *might* have some of the properties in the ontology. It is also important to remember that properties are part of what makes classes different from each other. Unique properties can ease the process of classification of new features.

Step 2, Expanding the Ontology with the DNC data

The next step in the process is to expand the ontology by using the DNC data. The expansion does not provide any changes to the hierarchical structure in the ontology, but new properties need to be integrated. From the DNC data set there are 7 new attributes, that can be viewed as properties. The attributes are described in table 4.17. The tree properties, Date, Name and Value can all be moved upwards in the ontology. They are all placed in the World class. There is an uncertainty whether the four remaining properties

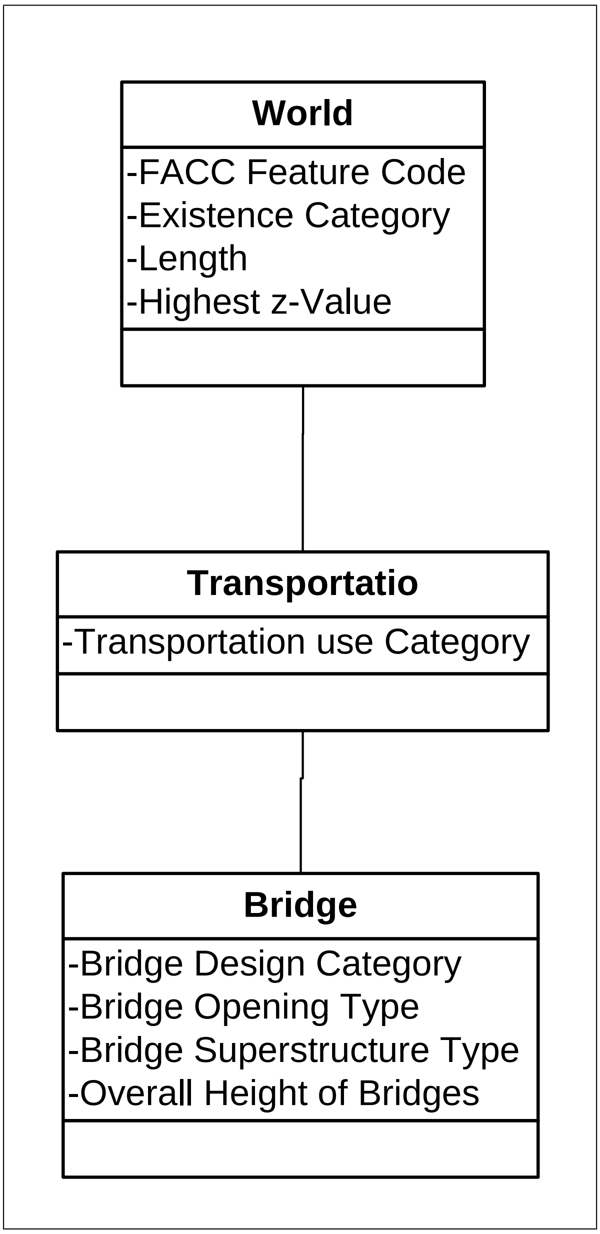

Figure 4.26: The VMAP1 bridge hierarchy after the slots has been rearranged

Table 4.17: Definition of the different attributes in the DNC data set according to the DIGEST standard

Definition of DNC attributes	
Attribute	*DIGEST definition*
Date	Date of report of activity
Maximum Vertical Clearance	The greatest distance between the traveled way and any obstruction vertically above it
Name	ANy identifier or code.
Overhead Clearance Category	The least distance between the traveled way and any obstruction vertically above it.
Over Water Obstruction	Indicates the presence of an obstruction over an area of navigable water.
Safe Horizontal Clearance	Minimum safe horizontal distance between adjacent bridge support structures on either side of a navigable channel passing under the bridge
Value	Numeric Value (used for year).

should be placed in the Transportation or Bridge class.

Overhead Clearance Category is defined as "The least distance between the traveled way and any obstruction vertically above it." and Maximum Vertical Clearance is defined as "The greatest distance between the traveled way and any obstruction vertically above it.". Based on these two definitions one can conclude that this kind of attributes can be valid for larger parts of a transportation system. Therefore these two properties are placed in the Transportation class. The two last slots are defined as followed: Over Water Obstruction, "Indicates the presence of an obstruction over an area of navigable water.", Safe horizontal Clearance, "Minimum safe horizontal distance between adjacent bridge support structures on either side of a navigable channel passing under the bridge.". These properties clearly relate to the fact that they come from DNC data. In a DNC setting these properties might be put higher in the ontology, but here these properties are both placed in the Bridge class. The resulting ontology with properties can be seen in figure 4.27. It is important to remember that the arrangement of properties is based on the choice of hierarchical structure in the ontology, and that changes to the hierarchy also could lead to changes in the property placement.

Another thing to consider is the aspect of the domain to a property. If a domain is a class, then a property can have many domains. This is where the balance between generality and duplication comes into consideration. Sometimes a property is valid for a portion of an ontology, if this is the case the property can be moved up to a higher level in the hierarchical structure. On other occasions a property is only valid for one domain in one branch, and one domain in another branch. In this situation, it is better to describe that the property is valid for two domains, rather than moving it up the hierarchy. The importance of this problem is decided by the type and level of detail on the ontology. As the goal here is to create a simple ontology, the properties will also be kept as simple as possible.

Step 3, Aggregating Similar Properties

The next step in the process is to aggregate properties that are similar. Properties are usually similar when they are synonyms, like size and area. Aggregation of properties to use the inheritance principle to see if two or more properties represent the same kind of meaning.

The World class contains 7 properties, and the properties FACC Feature Code and Name can both be viewed as means to identify a feature. One could maybe merge these two properties into a more general property, called Identifier or ID. The Feature Code is a five letter word that uniquely identifies the type of a feature. The name is defined as, "any identifier or code", which indicates that a name can be used as an identifier.

An aggregation or generalization could also be done to the Transportation class, here with the two properties Maximum Vertical Clearance and Overhead Clearance Category that could be merged into a more general Clearance Distance property. Finally the three properties Bridge Design Category, Bridge Opening Type and Bridge Superstructure Type could be merged into a single Bridge Construction property.

An ontology that fulfills these assumptions is shown in figure 4.28. What has to be taken

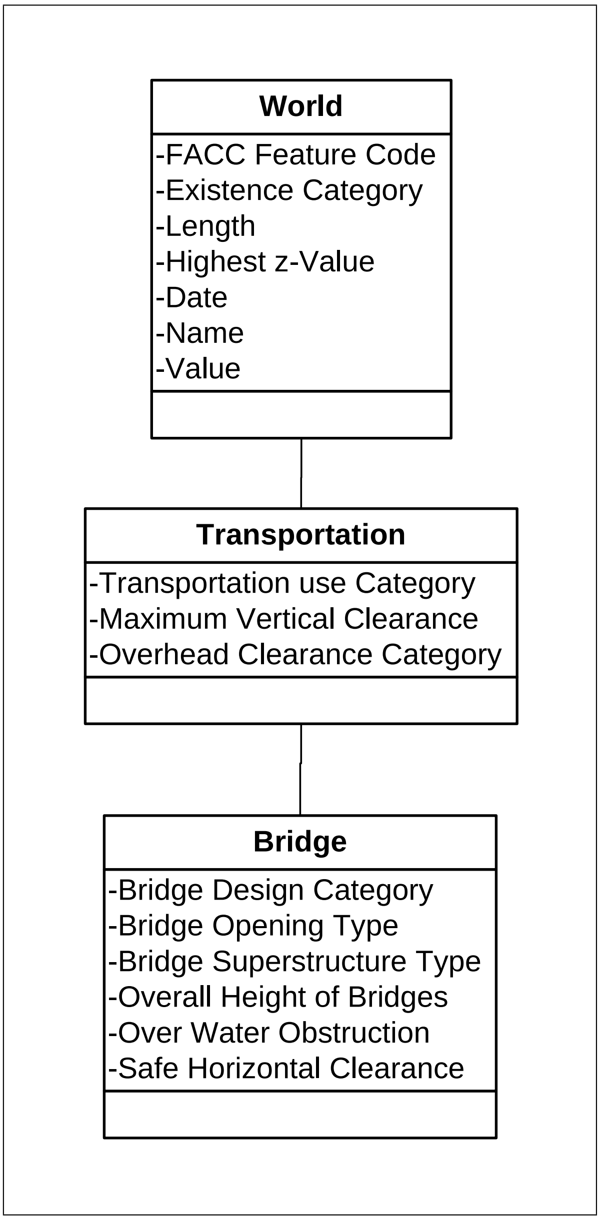

Figure 4.27: The VMAP1 bridge hierarchy after the DNC slots has been integrated and arranged accordingly

into consideration when aggregating and moving properties up or down the hierarchical structure is to balance the properties, so that they do not get too general or too specialized. The danger of making a property too general is that "all" attributes fits into this and therefore it will consist of too much different information. Likewise a class containing very specialized properties can quickly become very big. It is not necessary to have one property for each attribute, since some attributes are bound to be similar.

4.6 Implementation discussion

Ontologies are very helpful facilities when integrating data. Before an integration the data need to be classified. A successful classification depends on a good framework that can aid the user in the classification process. To make a good framework, or ontology for a number of various users with different interests is difficult. As a consequence of this, the ontologies to be used here are small and based on the scenarios in section 4.4

The OneMap project aims to build a large, global map which will be constructed incrementally and uncoordinated by many submissions [30]. From a semantic perspective problems that arise are connected with the diversity of users and data that such an approach presents. This diversity in users will also lead to a diversity of conceptualization about concepts. The scenarios in section 4.4 showed how many different solutions that could be applied to problems between two sources. Imagine what it could be like with 3 or 4 different data sources. The multiplicity in users indicates that use has to be as intuitive as possible.

Submission of new data will affect the ontology. In addition it might lead to structural changes in the ontology, the three most common is shown below.

1. New data might contain attributes which again lead to new properties in an ontology class.

2. New data might create new classes

3. New data can lead to structural changes in the ontology.

Examples on the second and third step is shown in figure 4.29 where a new bridge feature is added. The inclusion of the new feature leads to a refinement of the Bridge class, and two new subclasses are added.

To make an ontology expandable is never easy. By taking some assumptions it is possible to make some basic rules that dictate the way in which an ontology can evolve. A rearrangement of the whole ontology every time new data is added will be very inefficient, and should be avoided at great length. Basically, guidelines and usage should dictate the way an ontology can evolve. In the setting described here, a peer-review process would also regulate the development of the ontology.

Based on the setting described here the following operations should be possible.

- Users should be able to browse the ontology, either by browsing the OWL-file, or through a visualization of the hierarchy.

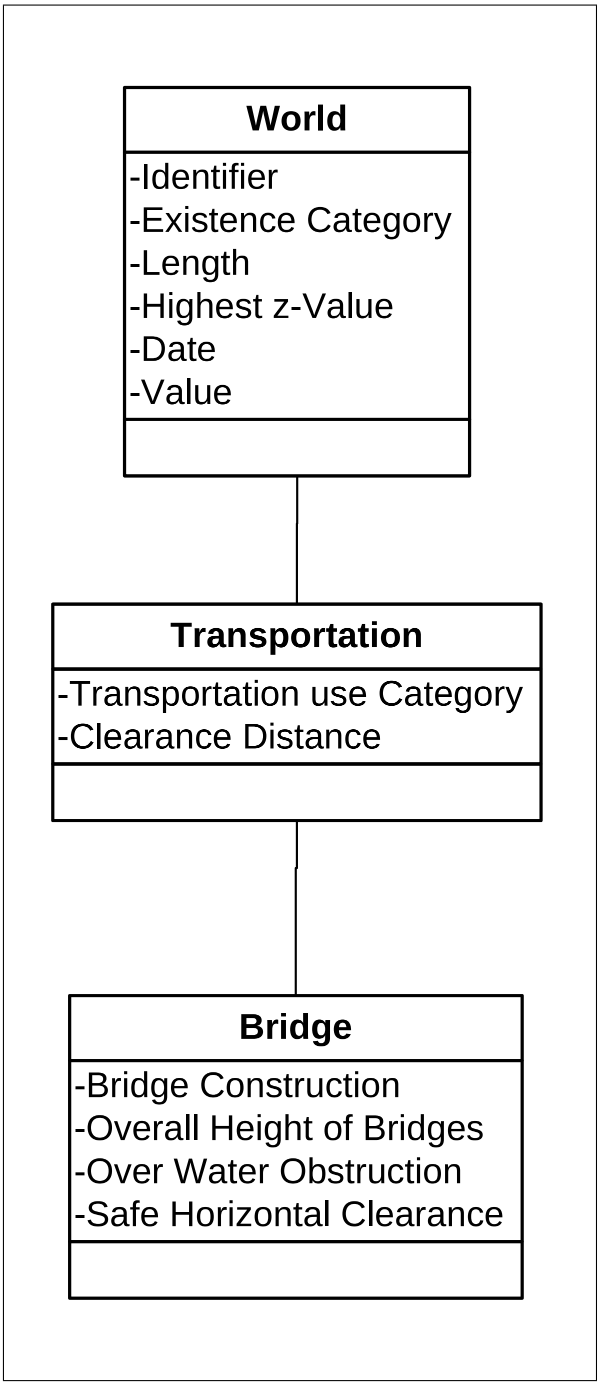

Figure 4.28: The VMAP1 bridge hierarchy after the slots have been merged.

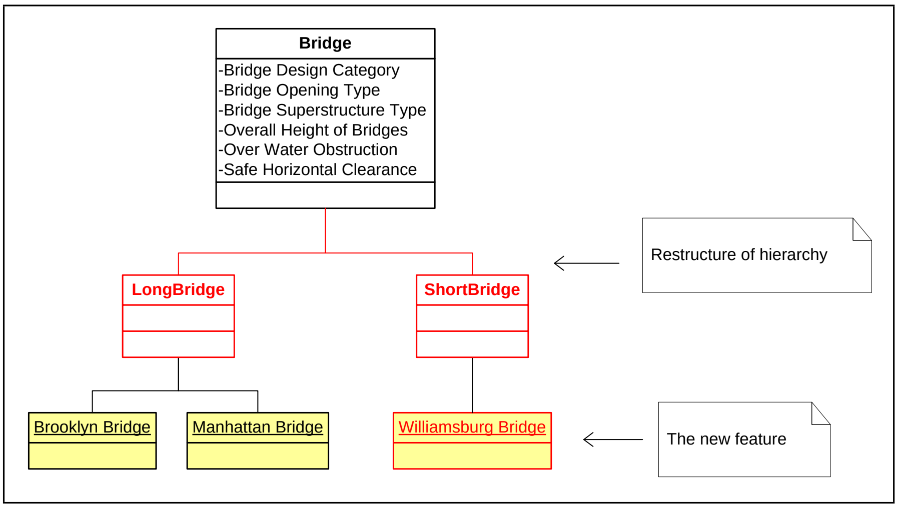

Figure 4.29: A simple case where a new feature leads to restructure of the ontology. The old hierarchy is inadequate to support the new feature, and is expanded to meet the demands.

- A user should be able to classify new data, by using the classes and their properties. One problem that might arise here is that no appropriate class exists. When this occurs the user should be able to suggest changes to the ontology.
- The changes can either be, to add properties, to add new classes, or a restructure of the whole ontology, by introducing completely new concepts, or by refinement of the already existing ones.

An additional functionality that could be of interest is the operation of retrieving data based on the ontology. By using the classes in the ontology one can search after features that fulfil the classification. To combine this with a geographical search could help in answering questions like, give all the coastlines of Australia, or features that are classified as bridges in Nepal, or all road segments in British Columbia, Canada, that are classified as regional highways.

An optimal approach would be to develop an application that could be used as a visual facility to better understand and use the ontology, but there are three factors that need to be examined further. These are: choice of ontology language, examination of available APIs and existing tools.

4.6.1 Ontology Language

When making an ontology a concern is which language to use. Several aspects of the language has to be taken into consideration. For instance what reasoning is supported in the language, how well does it describe classes and properties. As long as the ontology is kept simple as here, McGuinness [57] claims that the choice of ontology language is of lesser importance.

Three main languages used in ontologies are Resource Description Framework (RDF), DAML+OIL and Web Ontology Language (OWL). Here the OWL is chosen, since this is the emerging standard for ontology representation and provides the basic functionalities needed.

The OWL language provides basic and advanced functionalities for ontologies. What functionality to use depends on the complexity of the ontology. In addition the developer can also make choices towards which functionalities to use. To describe a simple ontology, as has been chosen here, it is desirable to only use the basic functionality like classes and datatype properties. Since the ontology will consist of a hierarchy it will also be possible to trace parents.

One of the strengths with the OWL is that it is understandable for both humans and applications, which indicate that it enables information share and reuse. This means that humans do not need applications to visualize the information embedded in OWL files, making it easy to share ontologies by simply sharing text files. The listing below show the classes in the ontology from figure 4.28 in OWL syntax.

```
<owl:Class rdf:ID="World">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      The Earth with all its inhabitants and all things upon it.
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Transportation">
  <rdfs:subClassOf rdf:resource="#World"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      The act or means of moving tangible objects from place to place,
      using a system of lines of movement or communication by road,
      rail, water or air.
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Bridge">
  <rdfs:subClassOf rdf:resource="#Transportation"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      A structure that spans and provides a passage over a road,
      railway, river or some other obstacle.
  </rdfs:comment>
</owl:Class>
```

From the code one can see that a class know about its parent, this makes it possible to derive the whole hierarchy from the owl file. The properties are not listed in the classes but they contain a reference to the class they belong to in the rdfs:domain tag, as shown in the following:

```
<owl:DatatypeProperty rdf:ID="Transportation_Use_Category">
  <rdfs:domain rdf:resource="#Transportation"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Identifies the primary user, function, or authority of the
      transportation system.
  </rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Table 4.18: Overview of OWL tools and which criteria the different tools support

	1	2	3	4	5	6	7
Protègè	X		X	X	X	X	
OntoTrack	X			X	X		
OilEd	X		X	X	X	X	
Swoop	X		X	X	X	X	
OntoXpl	X		X	X	X	X	
Ontolingua	X		X	X	X	X	
ezOWL	X	X	X	X	X	X	X

4.6.2 The Existing OWL Tools and APIs

In section 2.7.1 we presented different tools that support OWL development and usage. All in all there where six tools. It is desirable to establish which of these tools, if any, that support the requirements outlined in section 4.6. Fonseca [24] establishes the importance of a combination of roles and hierarchies during integration. The results showed that using a hierarchal structure to represent ontologies had a positive effect on the potential of information integration. Based on these results and the requirements a set of criteria are made, which are listed with decreasing importance.

1. The ontology should be OWL based
2. It should be possible to browse the ontology
3. It should be possible to add new properties
4. It should be possible to add new classes
5. The ontology should be easy to edit(expand, delete and rearrange)
6. The properties should be visible for the user so that new features are easier to classify
7. It should be possible to read the definition of classes and properties during browsing

The second criteria needs a little specification. It is desirable to see the classes in a hierarchy with their properties, similar to Unified Modeling Language (UML) models. Some tools make their ontologies browsable, but it is mainly the classes, and not properties, that are browsable.

A survey of the different tools based on these criteria has been collected in table 4.18. The table shows that most tools support the desired functionalities, but almost none of the tools support browsing with classes and properties. The only tool that supports all seven criteria is the ezOWL tool. This is in fact not a standalone application[2], but a plug-in to the Protègè tool.

[2]The ezOWL homepage (http://iweb.etri.re.kr/ezowl/index.html) indicates that a standalone application might be on its way.(As of june 2005)

In addition to all the tools, there also exists a number of OWL-APIs. These allow programmers and software engineers to develop their own OWL applications, custom-made for their own needs. Protègè and OilEd provide APIs. In addition to these two Hewlett Packard also provides an API called Jena.

4.6.3 Choice of Development

All in all there are three possible solutions to the implementation question.

1. Use existing tools
2. Develop something completely new
3. A combination of the two other possibilities.

The first solution is to use existing tools. This is a good choice if all requirements are met. It is time and labour saving and often represents well proven solutions. The negative part is that tools sometimes provide more functionality and possibilities than needed. To develop something new makes it possible to create customized applications. On the other hand development can be time consuming. The last combination is maybe the best approach, in that one can combine the advantages from the two first approaches. The negative part is that this requires an understanding of the existing tools and how these can be expanded.

The solution chosen here is to use already existing tools, namely the ezOWL tool that fulfilled all the criteria. The tool is reliable, supports OWL, allows for simple and advanced manipulation of the ontology, and provides graphical view of the hierarchy, as can be seen in figure 4.30 , figure 4.31 and figure 4.32.

4.7 Summary

This chapter presented the data sets to be used in the development process. Three scenarios was given to highlight different problems that might occur. The scenarios also provided a number of different solutions to each problem. After this one of the scenarios was developed into a simple ontology with classes, properties and relations between the classes. The ontology was presented in OWL. Finally a discussion about implementation was given, presenting the different possibilities that exist, and a choice of implementation was made.

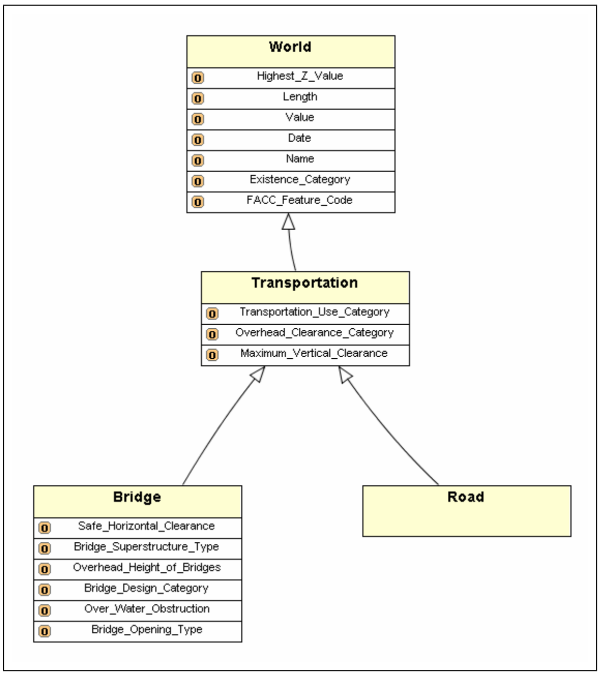

Figure 4.30: The Bridge hierarchy with properties as shown in ezOWL

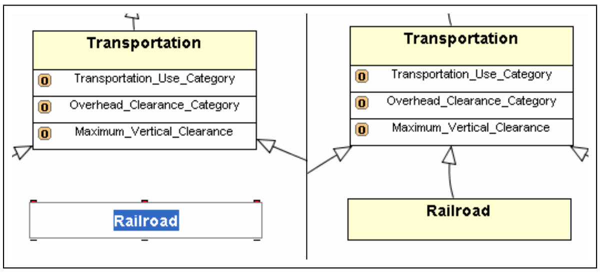

Figure 4.31: This figure shows how a class is created in ezOWL. First the class is created and given a name(to the left), then the class is linked to its parent(to the right)

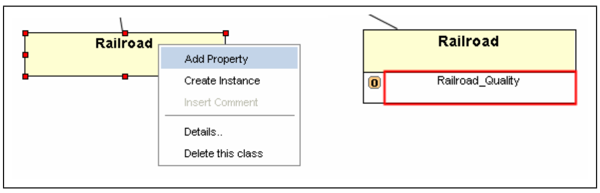

Figure 4.32: The figure shows a property being added in ezOWL

Chapter 5

Discussion and Conclusion

This chapter sums up the work presented in this thesis. First there is a discussion about the results from the scenarios. Then there is a conclusion, and the last section outlines some guidelines for further work

5.1 Discussion

Much of earlier research on ontology usage within geographical societies has based itself on upper-ontology making. These ontologies base themselves on the content and not the classification of existing data sets [34] [84]. Some has also tried to make upper-ontologies, not based on geographical classifications, but on general abstract classifications like CYC and GEMET [27]. Guidelines and methods presented by others have been helpful in the process of developing ontologies. The different methodologies offer various ways to approach problems. One of the most useful guidelines is to decide the scope and purpose of the ontology. When the purpose and scope have been identified a number of the possible solutions can be discarded. A determination of scope and purpose also helps to find concepts and arrange these [59] [22] [23]. Instead of finding concepts, one could try to find an appropriate existing classification that can act as a starting point. Indeed many methodologies suggest to use existing ontologies [27] [86] [69] [23], but few or none show how this can be done.

The scenarios in this thesis show that it is possible to make simple ontologies based on existing classifications. It is difficult, and there can be numerous possible solutions. This is shown trough the scenarios described in section 4.4, where a solution based on two different classifications produced as many as seven different solutions, and the three scenarios produced an average of 5 solutions. The solutions could be separated into three main bulks. One for each of the existing classifications and a third for a combination of the two. This mean that one could remove several of the possible solutions by deciding on which existing classification to base a solution on. As discussed in the previous paragraph.

Existing classifications that describes the same domain is bound to have some similarities. For instance by being similar as based on the same mapping standard, or similar in the form that they try to describe a domain in the same level of detail.

The data sources used in the scenarios where of a similar type. Similar in that they are both based on the VPF standard. This kind of similarity can lead to faster ontology building since the existing classifications already bear resemblance and are based on the same principles. On the other hand, using data sets based on different standards might lead to a more robust ontology, making it able to handle data from a greater range of sources.

When it comes to the level of detail, a classification can either be viewed as general or specialized. General in that it tries to describe a domain in a general manner, and specialized in that it tries to describe certain areas of the domain in a more specific manner. Two examples could be the harbour authority that needs a specific description of oceanic features, and the transportation council that needs a more specific description of the transportation features. The scenarios used in this thesis use one general, and one specialized classification. The general classification provides an overview of the domain, and the specialized classification adds more detailed classifications to certain areas of the domain. These two classifications complement each other and make the resulting ontology richer. An approach using two general classifications could again produce an ontology that was to general, whereas two specialized classifications could focus too much on specific problems and neglect some of the generality. Nevertheless if one decides to use one of the two last approaches, the data sets should be based on different standards.

Another question to address is the number of existing classifications to use. In this thesis two different were used, but two classifications produced at least 4 different solutions. An increase in existing classifications could either lead to, a) greater complexity and more solutions, or b) clarify problems, and help to determine which solution to use. Instead of integrating a third classification, one could use it as a control mechanism against the ontology whenever uncertainties arrived. It could be interesting to make ontologies based on more than two existing classifications and compare the results. It would also be useful to insert data in the ontology, to see how it performed during use. One could even use a conceptualization of the domain as a control against the simple ontology.

A choice should be made concerning which classification to use as starting point. This would rapidly decrease the number of possible solutions, and provide the developer with a quick overview of the domain at hand. One of the disadvantages with the existing classification approach, is that it limits the developers possibility to some length. By providing concepts and hierarchies, the developer might be locked in the existing conceptualization of a domain. This again might hinder the developer in developing his own conceptualization and understanding of the domain.

5.2 Conclusion

This thesis has focused on what kind of methods and best practices that can be identified in the development of simple ontologies or classifications. To find these methods and best practices three different scenarios have been used. The scenarios only represents parts of ontologies, and not complete ontologies.

Through the scenarios some basic methods or steps in the process of developing simple classifications have presented themselves. Firstly the intended usage and scope should be established. Secondly the level of detail should be outlined. Both of these criteria would assist in the process of finding an existing classification to use as a starting point. The next step would be to determine how many, and which other existing classifications to use. It is important to emphasize that there should always be at least two existing classifications involved in the development of a new one.

Based on the scenarios it looks like it is possible to create simple ontologies based on existing classifications. Such ontologies can either be used as simple ontologies or they can be viewed as a starting point for further development towards more complex ontologies. But more research is needed before one can be certain that it is possible to make full ontologies based on existing one.

5.3 Further Work

As mentioned above further research should be done before making any certain conclusion. For example decide on an existing classification and make a simple ontology based on this. Afterwards one could insert data from different sources and see how the ontology handled the new data.

Another aspect that could be followed up is that of number of existing classifications. One could use 2, 3, 4 and maybe 5 to determine whether there is an optimized number to use.

This thesis used two existing classifications based on the same standard. It would be very interesting to use classifications based on different standard, or from different mapping agencies and compare the results. There could also be a comparison on the methods on how to develop. Maybe the best practice and methods differ dependent on what kind of classifications that are used.

The last thing to point out again concerns usage. In an increment approach the ontology would be in constant development and change as new data is added. Supposedly adding of new data would enhance the ontology, making it easier to classify new data. A visualization tool could make the classification even easier. Making the user able to find the right place in the hierarchy in the combination with comparing attributes could lead to a more certain classification.

Bibliography

[1] Wikispecies. URL:http://species.wikipedia.org/wiki/Main_Page.

[2] Richard Fikes Adam Farquhar and James Rice. The Ontolingua Server: a Tool for Collaborative Ontology Construction. 1997.

[3] Bijan Parsia Aditya Kalyanpur, Evren Sirin and James Hendler. Hypermedia Inspired Ontology Engineering Environment: SWOOP. November 2004.

[4] National Geospatial-Intelligence Agency. DNC data. URL:http://www.nga.mil/portal/site/dnc/.

[5] Jochen Albrecht. Towards interoperable geo-information standards: A comparison of reference models for geo-spatial information. 1999.

[6] Chukwudozie Ezigbalike Andrew Phillips, Ian Williamson. Spatial Data Infrastructure Concepts. 1999.

[7] Grigoris Antoniou and Fran van Harmelen. *Handbook of Ontologies in Information Systems*, chapter Ontology Representation and Reasoning, pages 67–92. Springer, 2003.

[8] Ivan DeLoatch Hank Garie Barbara J. Ryan, Mark L. DeMulder and Karen Siderelis. A Clear Vision of the NSDI. *Geospatial Solutions*, 2004.

[9] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens. OilEd: a reason-able ontology editor for the semantic web. In *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence*, number 2174 in Lecture Notes in Computer Science, pages 396–408, Vienna, September 2001. Springer-Verlag.

[10] Tim Berners-Lee. Semantic Web on XML, 2000. URL:http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html.

[11] Tim Bray. What is RDF? URL:http://www.xml.com/pub/a/2001/01/24/rdf.html?page=1, 2001.

[12] Terminology Working Group Cambridge. Thesaurus principle and practice. February 1992.

[13] Luca Cardelli. A Semantics of Multiple Inheritance. *Information and COmputation*, pages 138–164, 1988.

[14] David M. Danko. The Vector Product Format, An Overview. pages 67–72, 1995. URL:http://libraries.maine.edu/Spatial/gisweb/spatdb/acsm95/ac95059.html.

[15] Djun Kim Dave Blasby, Martin Davis and Paul Ramsey. GIS Conflation using Open Source Tools. URL:http://www.jump-project.org/assets/JUMP_Conflation_Whitepaper.pdf.

[16] Trevor Bench-Capon Dean Jones and Pepijn Visser. METHODOLOGIES FOR ONTOLOGY DEVELOPMENT. 1998.

[17] Lorcan Dempsey and Rachel Heery. *Journal of Documentation*, chapter Metadata: a current view of practice and issues, pages 145–172. Emerald, London, England, 1998.

[18] Michael Denny. Ontology Building: A survey of Editing tools. 2002.

[19] Digital Geographic Information Working Group (DGIWG). The Digital Geographical Information Exchange Standard, Part 4 Feature and Attribute Coding Catalouge. URL:http://www.digest.org/html/gp40.htm.

[20] Michael Lutz Eva Klien, Udo Einspanier and Sebastian Hübner. An Architecture for Ontology-Based Discovery and Retrieval of Geographic Information. 2003.

[21] Sabine Timpf Farshad Hakimpour. Using Ontologies for Resolution of Semantic Heterogeneity in GIS. April 2001.

[22] Mariano Fernandez, Asuncion Gomez-Perez, and Natalia Juristo. METHONTOLOGY: from Ontological Art towards Ontological Engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, Stanford, USA, March 1997.

[23] Natalya F.Nov and Deborah L.McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. 2000.

[24] Fredrico Torres Fonseca. *Ontology-Driven Geographic Information Systems*. PhD thesis, The University of Maine, USA, 2001.

[25] Peggy Agouris Frederico T.Fonseca, Max J.Egenhofer and Gilberto Câmara. Using Ontologies for Integrated Geographic Information Systems. 2002.

[26] Christiane Fellbaum Derek Gross George A. Millner, Richard Beckwih and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[27] Heiner Stuckenschmidt Gerhard Schuster. Building Shared Ontologies for Terminology Integration. 2001.

[28] Alan Gilchrist. Thesauri, taxonomies and ontologies - an etymological note. *Journal of Documentation*, 2003.

[29] Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. 1993.

[30] Knut-Erik Johnsen Gunnar Misund. The One Map Project. July 2002.

[31] V. Haarslev and R. Möller. Racer: An owl reasoning agent for the semantic web. 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,Ê Halifax, Canada, October 13*, pages 91–95, 2003.

[32] Farshad Hakimpour and Andreas Geppert. Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach. 2001.

[33] Nicolaas J.I. Mars Harry T.Uitermark, Peter J.M. van Oosterom and Martien Molenaar. Ontology-Based Geographic Data Set Integration. September 1999.

[34] Nicolaas J.I. Mars Harry T.Uitermark, Peter J.M. van Oosterom and Martien Molenaar. Ontology-Based Geographic Data Set Integration. April 2002.

[35] Linda L. Hill. Core Elements of Digital Gazetteers: Placenames, Categories, and Footprints. URL:`http://www.alexandria.ucsb.edu/~lhill/paper_drafts/ECDL2000_paperdraft7.pdf`.

[36] Diane Hillmann. Using Dublin Core, 2003. URL:`http://dublincore.org/documents/usageguide/index.shtml`.

[37] Natalya F.Noy Holger Knublauch, Ray W.Fergerson and Mark A.Musen. The Protègè OWL Plugin: An Open Development Environment for Semantic Web Applications. November 2004.

[38] Ian Horrocks. DAML+OIL: an Ontology Language for the Semantic Web. URL:`http://www.cs.man.ac.uk/~horrocks/Slides/daml-pi-feb-01.pdf`.

[39] Ian Horrocks. DAML+OIL: a Description Logic for the Semantic Web. 2001.

[40] Catherine Houstis. Ontology-Driven Conceptual Modelling. URL:`http://support.inf.uth.gr/courses/CE428/Ontologies.ppt`.

[41] H.T.Uitermark. Ontology Construction for Geographic Data Set Integration. 1999.

[42] Patrick J.Cassidy. An Investigation of the Semantic Relations in the Roget's Thesaurus: Preliminary Results. February 2000.

[43] Ian Dickinson Andy Seaborne Chris Dollin Jeremy J.Carroll, Dave Reynolds and Kevin Wilkinson. Jena: Implementing the Semantic Web Recommendations. May 2004.

[44] Valorie Lee Joel Tochterman. Terminology Project Report, 2000. URL:http://eea.eionet.eu.int:8980/irc/DownLoad/kjedAgJ2mjG9usiEUEv_VvAKd0qYIfZUE_-sfP1QBF8j5/annex2.pdf.

[45] York Sure Jürgen Angele. Whitepaper: Evaluation of Ontology-based Tools. 2002.

[46] M. Kavouras. A unified ontological framework for semantic integration. October 2003.

[47] Michael Harrington Kevin Corrigan. Pseudo-Dionysis the Areopagite. URL:http://plato.stanford.edu/entries/pseudo-dionysius-areopagite/.

[48] Werner Kuhn. Semantics of geographic information. *Geoinfo-series, Vol. 7*, 1996.

[49] Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[50] Thorsten Liebig and Olaf Noppens. OntoTrack: Fast Browsing and Easy Editing of Large Ontologies. October 2003.

[51] Thorsten Liebig and Olaf Noppens. ONTOTRACK A New Ontology Authoring Approach. November 2004.

[52] James Frew Linda L. Hill and Qi Zheng. Geographic Names, 1999. URL:http://www.dlib.org/dlib/january99/hill/01hill.html.

[53] mapability.com. VMAP0 data. URL:http://mapability.com/info/vmap0_index.html.

[54] mapability.com. VMAP1 data. URL:http://mapability.com/info/vmap1_index.html.

[55] Moya K. Mason. How Does the Dewey Decimal Classification System Satisfy Some of the Fundamental Laws Govering Knowledge. URL:http://www.moyak.com/researcher/resume/papers/dewey.html.

[56] Alan Rector Robert Stevens Chris Wroe Matthew Horridge, Holger Knublauch. *A Practical Guide To Building OWL Ontologies Using The Protègè-OWL Plugin and CO-ODE Tools.*

[57] Deborah L. McGuinness. Ontologies Come of Age. 2001.

[58] Lance McKee and Cliff Kottman. Inside the OpenGIS Specification. URL:http://portal.opengeospatial.org/files/?artifact_id=6205&version=1&format=pdf.

[59] Michael Gruninger Mike Uschold. ONTOLOGIES: Principles, Methods and Applications. 1996.

[60] MIL-PRF-89023. Digital Nautical Chart. 1997.

[61] MIL-PRF-89033. Vector Smart Map Level 1. 1995.

[62] MIL-PRF-89039. Vector Smart Map Level 0. 1995.

[63] Eric Miller. An Introduction to the Resource Description Framework. *D-Lib Magazine*, 1998.

[64] Gunnar Misund. One Map. *Høit*, (1), 2002.

[65] Gunnar Misund and Harald Vålerhaugen. Integration of Heterogeneous GML Sources. July 2004.

[66] Patricia Charlton Myriam Ribère. Ontology Overview Motorola Labs. 2000.

[67] S.Decker M.Crubezy R.W.Fergerson N.F.Noy, M.Sintek and M.A.Musen. Creating Semantic Web Content with Protege-2000. *IEEE Intelligent Systems*, 2001.

[68] University of Concordia. Ontology Explorer Tool (OntoXpl). URL:`http://www.cs.concordia.ca/ying_lu/`.

[69] Asunción Gómez-Pérez Oscar Corcho, Mariano Fernández-Lopéz and Angel López-Cima. *Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*, chapter Building legal ontologies with METHONTOLOGY and WebODE, pages 142–157. Springer-Verlag GmbH, 2005.

[70] ESRI White paper. Whitepaper: Metadata and GIS. 2002.

[71] Herman A.Dohmann Pierre J.C. Beaulieu. THE DIGITAL GEOGRAPHIC INFORMATION EXCHANGE STANDARD AND MILITARY MAPPING. 1997.

[72] Marc Steinhaus Paolo Plini Ralf Nikolai, Ralf Kramer and Bruno Felluga. GenThes: A General Thesaurus Browser for Web-based Catalouge Systems. April 1999.

[73] Alessio Bertone Riccardo Albertoni and Monica De Martino. Semantic Web and Information Visualization. 2001.

[74] V.Richard Benjamins Rudi Studer and Dieter Fensel. Knowledge Engineering: Principles and Methods. 1998.

[75] Alan Saalfeld. Conflation: Automated Map Compilation. *International journal of geographical information systems*, 1988.

[76] Raphael Volz Sean Bechhofer and Philip Lord. Cooking the Semantic Web with the OWL API. October 2003.

[77] Terence R. Smith. A digital library for geographically referenced material. *IEEE Computer*, 29(5):54–60, 1996.

[78] Leonid Stoimenov and Slobodanka Dordevic-Kajan. Framework For Semantic GIS Interoperability. 2002.

[79] U.S. Geological Survey. Vector Smart Map Level 0 (VMap0). URL: http://geography.usgs.gov/esic/cdrom/vmap.html.

[80] C.Parent T.Devogele and S.Spaccapietra. On Spatial Database Integration. *International Journal of Geographic Information Science 4*, 1998.

[81] Gerhard Schuster Thomas Vögele, Sebastian Hübner. BUSTER - An Information Broker for the Semantic Web, 2003. URL:http://www.kuenstliche-intelligenz.de/archiv/2003_3/buster.pdf.

[82] James Hendler Tim Berners-Lee and Ora Lassila. The Semantic Web. 2001.

[83] Joyce A. Hoffman Tony Moore, David Wesloh and Richard Glass. Vector Smart Map Level 1 - A New DMA Product. pages 83–92, 1995.

[84] Harry T. Uitermark. *Ontology-Based Geographic Data Set Integration.* PhD thesis, 2001.

[85] Mike Uschold. Building Ontologies: Towards a Unified Methodology. December 1996.

[86] Mike Uschold and Martin King. Towards a Methodology for Building Ontologies. July 1995.

[87] Lluis Pèrez-Vidal Villie Morocho and Fèlix Saltor. Semantic Integration on Spatial Databases, SIT-SD prototype. November 2003.

[88] W3C. OWL Web Ontology Language Guide. URL:http://www.w3.org/TR/owl-guide/.

[89] Shuxin Yuan and Chuang Tao. Development of conflation components. June 1999. URL: http://www.umich.edu/~iinet/chinadata/geoim99/Proceedings/yuan_shuxin.pdf.

[90] Villie Morocho Zurita. *Semantic-based Approach to Spatial Data Sources Integration.* PhD thesis, 2004.

Appendix A

Glossary

Category: A collection of items that are related in some way.
Classification: To place things, concepts, objects into categories or classes according to defined criteria for each category/class.
Coverage: A set of thematically associated data considered as a unit. A coverage usually represents a single theme, or layer, such as streams, roads or buildings.
CYC: An artificial intelligence project which attempts to assemble a comprehensive ontology and database of everyday common-sense knowledge, with the goal of enabling AI applications to perform human-like reasoning
DAML+OIL: DARPA Agent Markup Language and Ontology Interface Layer. DAML+OIL is a syntax, layered on RDF and XML, that can be used to describe sets of facts making up an ontology.
Datum: A mathematically defined reference surface used to represent the size and shape of the earth. A horizontal datum is defined by its ellipsoid, latitude and longitude orientation, and a physical origin.
Descpription Logic: The term "description logic" refers to a logic that focuses on descriptions as its principal means for expressing logical expression. Description logics refers to concepts descriptions used to describe a domain.
DNC: Digital Nautical Chart. A vector based product designed to provide an up-to-date seamless database of the world. It is produced in the VPF standard. The main focus of DNC is on coastline, harbour, and near harbour/coastline related information.
Feature: A geographic component of the earth's surface that has both spatial and attribute data associated with it.
Gazetteer: A geographical dictionary or index with location.
GEMET: GEneral Multilingual Environmental Thesaurus. A general thesaurus aimed to define a common general language, a core of general terminology for the environment.
GIS: Geographic Information System. A system of hardware and software used for storage, retrieval, mapping, and analysis of geographic data
Hierarchy: A hierarchy is a system of ranking and organizing things.
Metadata: Data about data.
Ontology: An explicit formal specification of how to represent the objects, concepts, and

other entities that are assumed to exist in some area of interest and the relationships that hold among them.
OWL: Web Ontology Language. A markup language for publishing and sharing data using ontologies on the Internet.
RDF: Resource Description Framework. A markup language that enables the encoding, exchange and reuse of structured metadata.
The Semantic Web: Having data on the Web defined and linked in a way that it can be used by machines not just for displaying purposes, bur for automation, integration and reuse of data across various applications. RDF and OWL are important parts of the Semantic Web.
Taxonomy: A hierarchical classification of things in an ordered system that indicates natural relationships.
Thesaurus: A list of words showing similarities, differences, dependencies and other relationships to each other.
VMAP0: Vector Smart MAP Level 0. A map that provides worldwide vector-based geospatial data coverage of the world.
VMAP1: Vector Smart MAP Level 1. A vector-based map over the world. Only small parts of the map is available for the public.
VPF: Vector Product Format. A digital geographic vector-based format used by the US Defense Mapping Agency for the distribution of vector data sets. The VMAP and DNC data sets are based on this standard.

Appendix B

OWL files

B.1 OWL Danger Point, Hazard Point, Solution 3

```
<?xml version="1.0"?> <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="World"/>
  <owl:Class rdf:ID="Obstructions">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Any structure that makes progress difficult</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Hydrography"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Hydrography">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The science of the measurement and description and mapping of the
        surface waters of the earth with special reference to navigation
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#World"/>
  </owl:Class>
  <owl:Class rdf:ID="Hazard">
    <rdfs:subClassOf rdf:resource="#Obstructions"/>
  </owl:Class>
  <owl:Class rdf:ID="Obstructs">
    <rdfs:subClassOf rdf:resource="#Obstructions"/>
  </owl:Class>
  <owl:Class rdf:ID="Danger">
    <rdfs:subClassOf rdf:resource="#Obstructions"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Danger at the sea represented as a point. </rdfs:comment>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="Sea_Floor_Feature_Category">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Hydrography"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Type of object or area on the sea floor or below the water surface
    </rdfs:comment>
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Hydrographic_Depth_Information">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Information about the accuracy or availability of depth or
      uncovering height of a feature
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Hydrography"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Text_Attribute">
  <rdfs:domain rdf:resource="#World"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Narrative or other description</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="FACC_Feature_Code">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#World"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Material_Composition_Category">
  <rdfs:domain rdf:resource="#World"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Characteristics of primary material composition of feature</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Accuracy_Category">
  <rdfs:domain rdf:resource="#World"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Accuracy of geographic position</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Existence">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >State or condition of the feature</rdfs:comment>
  <rdfs:domain rdf:resource="#World"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Location_Category">
  <rdfs:domain rdf:resource="#Hydrography"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Status of feature relative to surrounding area or water
  </rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Certainty_of_Delineation">
  <rdfs:domain rdf:resource="#Obstructions"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Indicates knowledge of the features limits or information</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Name">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#World"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Any identifier or code</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Vertical_Reference_Category">
  <rdfs:domain rdf:resource="#World"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Relative location referenced to sounding datum, unless
```

```
        otherwise indicated
    </rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Area_Coverage_Attribue">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >The absolute area within the delineation of the feature</rdfs:comment>
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Value">
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Numeric Value</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Hydrographic_Drying_Height">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The height of the feature, which tidal waters cover and uncover,
        referenced to a specified vertical datum
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Hydrography"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Severity_of_Hazard">
    <rdfs:domain rdf:resource="#Hazard"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Severity of hazard</rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Date">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Date of report of acitivity</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#World"/>
  </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 1.3, Build 225.4)
http://protege.stanford.edu -->
```

B.2 OWL Bridge Duality, Solution 1

OWL file for the transportation system, containing an instance of Brooklyn Bridge.

```
<?xml version="1.0"?> <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="World">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The Earth with all its inhabitants and all things upon it.
    </rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Transportation">
    <rdfs:subClassOf rdf:resource="#World"/>
```

```
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The act or means of moving tangible objects from place to place,
        using a system of lines of movement or communication by road, rail,
        water or air.
    </rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Road">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        An open way maintained for vehicular use
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Transportation"/>
  </owl:Class>
  <owl:Class rdf:ID="Bridge">
    <rdfs:subClassOf rdf:resource="#Transportation"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        A structure that spans and provides a passage over a road, railway,
        river or some other obstacle.
    </rdfs:comment>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="Safe_Horizontal_Clearance">
    <rdfs:domain rdf:resource="#Bridge"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Minimum safe horizontal distance between adjacent bridge support
        structures on either side of a navigable channel passing under
        the bridge.
    </rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Length">
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        A measurement of the longer of two linear axes in meters. For a
        round feature, measure the diameter. For a bridge, the length
        is the distance between the bridge abutments.
    </rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Bridge_Superstructure_Type">
    <rdfs:domain rdf:resource="#Bridge"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Structural design characteristics
    </rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="FACC_Feature_Code">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        A unique Feature Code.
    </rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Highest_Z_Value">
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Elevation above a given datum to the highest portion of the feature.
    </rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Maximum_Vertical_Clearance">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The greatest distance between the traveled way and any obstruction
```

```
        vertically above it.
    </rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#Transportation"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Existence_Category">
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        State or condition of the feature
    </rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Overhead_Height_of_Bridges">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Vertical distance mesaured from the lowest point at ground or water
        level to the highest portion of bridge.
    </rdfs:comment>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Bridge"/>
          <owl:Class rdf:about="#Road"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Over_Water_Obstruction">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Indicates the presence of an obstruction over an area of
        navigable water
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Bridge"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Name">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The name of the feature, any identifier or code.
    </rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Overhead_Clearance_Category">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The least distance between the traveled way and any obstruction
        vertically above it.
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Transportation"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Value">
    <rdfs:domain rdf:resource="#World"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Numeric value (used for year).
    </rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Bridge_Design_Category">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Structural design characteristics of the bridge or a bridge segment
```

```
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Bridge"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Transportation_Use_Category">
    <rdfs:domain rdf:resource="#Transportation"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Identifies the primary user, function, or authority of the
        transportation system.
    </rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Bridge_Opening_Type">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        The type of structure or mechanism by which a portion of a bridge is
        moved to allow passage of a vessel.
    </rdfs:comment>
    <rdfs:domain rdf:resource="#Bridge"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Date">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Date of report of activity
    </rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#World"/>
  </owl:DatatypeProperty>
  <Bridge rdf:ID="Brooklyn_Bridge">
    <Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        0
    </Value>
    <Maximum_Vertical_Clearance
        rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
        39.0
    </Maximum_Vertical_Clearance>
    <Existence_Category
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        28
    </Existence_Category>
    <FACC_Feature_Code
        rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        AQ040
    </FACC_Feature_Code>
    <Date rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        26
    </Date>
    <Bridge_Superstructure_Type
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        0
    </Bridge_Superstructure_Type>
    <Transportation_Use_Category
        rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        4
    </Transportation_Use_Category>
    <Safe_Horizontal_Clearance
        rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
        411.0
    </Safe_Horizontal_Clearance>
    <Highest_Z_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        29999
    </Highest_Z_Value>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
```

```
        The Brooklyn Bridge from Manhattan to Brooklyn in NYC
    </rdfs:comment>
    <Over_Water_Obstruction
        rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        1
    </Over_Water_Obstruction>
    <Bridge_Opening_Type
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        13
    </Bridge_Opening_Type>
    <Bridge_Design_Category
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        0
    </Bridge_Design_Category>
    <Overhead_Clearance_Category
        rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
        34.0
    </Overhead_Clearance_Category>
    <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        Brooklyn Bridge
    </Name>
    <Length rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
        877
    </Length>
    <Overhead_Height_of_Bridges
        rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
        0.0
    </Overhead_Height_of_Bridges>
  </Bridge>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 1.3, Build 225.4)
http://protege.stanford.edu -->
```