Evaluating a Semantic Approach to Address Data Interoperability

by

Noh Teamrat Tewolde

Submitted in partial fulfillment of the requirements for the degree

Master of Science (Computer Science)

in the Faculty of Engineering, Built Environment and Information Technology,

University of Pretoria

1 December 2014

Abstract

Semantic approaches have been used to facilitate interoperability in different fields of study. Current literature, however, shows that the semantic approach has not been used to facilitate the interoperability of addresses across domains. Addresses are important reference data used to identify locations and /or delivery points. Interoperability of address data across address or application domains is important because it facilitates the sharing of address data, addressing software and tools which can be used across domains. The aim of this research study has been to evaluate how a semantic (ontologies) approach could be used to facilitate address data interoperability and what the challenges and benefits of the semantic approach are.

To realize the hypothesis and answer the research problems, a multi-tier hierarchy of ontology architecture was designed to integrate (across domain) address data with different levels of granularities. Four-tier hierarchy of ontologies was argued to be the optimal architecture for address data interoperability. At the top of the hierarchy was Foundation-Tier that includes vocabularies for location-related information and semantic language rules and concepts. The second tier has address reference ontology (called Base Address Ontology) that was developed to facilitate interoperability across the address domains. Developing optimal address reference ontology was one of the major goals of the research. Different domain ontologies were developed at the third tier of the hierarchy. Domain ontologies extend the vocabulary of the BAO (address reference ontology) with domain specific concepts. At the bottom of the hierarchy are application ontologies that are designed for specific purpose within an address domain or domains. Multiple scenarios of address data usage were considered to answer the research questions from different perspectives.

Two interoperable address systems were developed to demonstrate the proof of concepts for the semantic approach. These interoperable environments were created using the UKdata+UPUdata ontology and UKpostal ontology, which illustrate different use cases of ontologies that facilitate interoperability. Ontology reason, inference, and SPARQL query tools were used to share, exchange, and process address data across address domains. Ontology inferences were done to exchange address data attributes between the UK administrative address data and UK postal service address data systems in the UKdata+UPUdata ontology. SPARQL queries were, furthermore, run to extract and process information from different perspective of an address domain and from combined perspectives of two (UK administrative and UK postal) address domains. The second interoperable system (UKpostal ontology) illustrated the use of ontology inference tools to share address data between two address data systems that provide different perspectives of a domain.

Key Words: Address, Standards, Ontology, Semantics, Semantic Web, Interoperability, Foundation ontology, Reference ontology, Domain ontology, Application Ontology, Ontology Inference, and Web Ontology Language (OWL)

Table of Contents

Table of Figures

List of Tables

Chapter 1

1 Introduction

1.1 Background

There are some claims that the large majority of the world's data has some spatial component. In fact, the most commonly used statistic in this regard is the 80% (Kerski 2008). Many individuals and companies in the field of geographic information science (GISc), including major corporates such as ESRI, claim that 80% of world's data has a spatial aspect (ESRI 2012). This claim has been there since the early 1990s. It is difficult to give the exact ratio or percentage of spatial data in the world, but, from the large and ever increasing influence of spatial information to our daily activities and decisions, one can assume that no small amount of world data has spatial aspect.

Spatial information is used to analyze and make decisions in different sectors, including agriculture, forestry, education, health, emergency management, local and national governments, natural resources management, and many industry sectors such as mining, tourism, insurance, transport, retail, and utilities, amongst others (ESRI 2012). One of the most commonly used forms of spatial information is address data.

Addresses are important reference data used to identify locations and /or delivery points. They are omnipresent in our daily activities (Coetzee 2008b). Addresses are used for different purposes and as a result they have a variety of forms.

In many countries, address data are standardized to be significantly useful to their communities. Various bodies have developed national, regional, and international standards for the collection, maintenance, and use of address data (ISO/TC 211 2011). Examples of international address standards are, the Universal Postal Union's Standard-42 (UPU-S42) and the address data specification of the Infrastructure for Spatial Information in Europe (INSPIRE). These standards provide uniformity in the collection, maintenance, and use of address data for the specific task for which they are designed, in their scope of coverage. For example, UPU S-42 is used by postal service providers to render delivery point identifiers globally (UPU/POC Addressing Group

2006a), while the INSPIRE address data specification is published for use by European public authorities for governmental administration purposes (INSPIRE/TWG/ Addresses 2009).

Even though the various address standards provide advantages to the respective users, they also create new challenges for the integration of address data across standards. Interoperability of address data across standards is important because it motivates the sharing of address data and addressing software and tools which are used across regions. For cultural and technical reasons, however, it is impossible to develop an international address standard that *replaces* all the existing standards (ISO/TC 211 2011).

One of the challenges when considering national and regional address standards is the variety of formats used to describe and represent an address and/or addressable object in the different standards. For example, the Chinese, South Korean, and Japanese address standards prescribed for different address formats; as a result, it becomes difficult to execute regional projects that use address data from these countries. Thus facilitating interoperability based on the semantics of the terms rather than their syntactic representation could be a solution to addressing information interoperability. This scenario is researched and discussed in this dissertation.

Semantic ontologies have been used in a number of fields to create common vocabularies and improve interoperability across domains (Rosse & Mejino 2003; Andersson et al. 2006; Pardo et al. 2012). These kinds of ontologies capture the content (meaning) of information (data) on the web. This research study discusses the use of ontologies for addressing data interoperability by introducing a four-tier hierarchy of ontologies.

Some research has been done on the use of ontologies to capture, share, and exchange semantic data in geo-informatics (Lutz & Klien 2006; Ilayperuma 2007; Huang 2010). More research has also been done on ontologies for various kinds of information interoperability (Ilayperuma 2007). There are, however, yet no reports of a reference ontology for international address information interoperability. This research study on ontology for addressing information interoperability follows the best practices of reference ontologies used for information interoperability in other fields.

1.2 Problem Statement

A house, building, landmark, farm, or any other addressable object can have multiple addresses based on different address standards which are used for different purposes. For example, a house can have an address used by local government authorities and another one used for postal delivery. The addresses could be assigned based on different standards which have different formats and attributes (content) depending on their purpose. The use of different address standards leads to challenges for interoperability.

Address data contain a variety of attributes depending on the standard used to assign the addresses. The address data that are collected by different entities are difficult to share owing to their differences in content and format. Not only does the data differ, but address software and tools are also constrained in their use. Address data interoperability across address and application domains facilitates the sharing of addresses and addressing tools and software, amongst others.

Address assignment is a task that needs to be done carefully to avoid the duplication or skipping of addresses. Different countries use different ways of address assignment. Some of these countries have developed address standards. Others do not have a standardized way of address assignment, and they assign addresses based on the chronological order of the construction of addressable objects (Kazuhiko 2008).

Some address standards have address attributes that do not exist in other standards. As a result, the digital address data, stored in data bases or other data storage formats, do not have uniformity. Over time, these standards have become part of the culture and identity of societies. It is, therefore, impossible to replace all the existing standards with a single international standard for all countries and purposes.

In the past, international organizations, such as the Universal Postal Union (UPU), INSPIRE, and OASIS, have developed international address standards to facilitate their respective work (with in specific application domains). The standards facilitate analysis, query, and processing address data from different countries. Such standards are, however, designed for the unique purposes of the organizations (e.g. UPU-S42 for postal delivery, INSPIRE address data specification for local government administration) and are not necessarily useful for other purposes.

There is no single standard that can be used for all addressing purposes. It is impossible to develop such a standard because the existing standards are too diverse in purpose and scattered across all geographical boundaries. There are terminological conflicts among the different standards. Attributes which are used in one standard do not necessarily exist in others. It is, thus, not possible to merge all the existing address standards into one big standard that accommodates all of them. It is also not possible to change the existing address standards for cultural, financial, and other reasons.

The aim of this research is to investigate the use of semantic ontology as a tool to facilitate address data interoperability across address and application domains. Semantic ontologies have been used previously to facilitate interoperability in other fields but not for addressing. Addresses have unique characteristics and importance to businesses, governments, and the day-to-day life of individuals. Hence, this dissertation investigates the following research questions:

- How the semantic approach can be used to facilitate the interoperability of address data across multiple address and application domains?
- How efficient and effective the semantic approach would be?
- Are there any other options to facilitate interoperability? If so, how can the semantic approach be compared to and contrasted with the other approaches?
- What are the main challenges of using a semantic approach to facilitate interoperability?
- What are the advantages of such an approach?

At the end of the research the answers to the above questions are expected to contribute useful information to science and technology. It would also be impressive to open new doors for further research and contributions as a result of the answers to the above-mentioned questions.

1.3 Research objectives

The main objectives of this research are:

- ✓ To review literature and related technology;
    - To review existing address standards;
    - To review semantic ontology, description languages, and ontology languages;
    - To review previous attempts to facilitate interoperability in different fields;

- ✓ To design experiments for evaluating a semantic approach to address data interoperability;
    - To experiment different software tools used for semantic interoperability;
    - To design an n-tier hierarchy of ontologies for address data interoperability;
    - To implement this hierarchy on a small scale to ensure that it can facilitate interoperability using semantic ontologies;
    - To develop ontologies for specific address standards for the n-tier hierarchy of ontologies;
    - To conduct interoperability experiments with the hierarchy of ontologies developed in this research;
    - To conduct experiments for address data interoperability and exchange;
    - To experiment queries and inferences;
- ✓ To analyze and discuss the results;
    - To analyze critically the results of the experiments; and
- ✓ To suggest a way for future research.

1.4 Methodology

The goal of this research is to explore address data interoperability across domains using the semantics of the data. This section provides an overview of the methodology followed to conduct the research. The detailed description of the methodology is discussed in Chapter three.

An ontology-based approach was followed. A four tier (extendible) hierarchy of ontologies was designed to integrate address data which have different forms and syntactic representations (see figure 1).

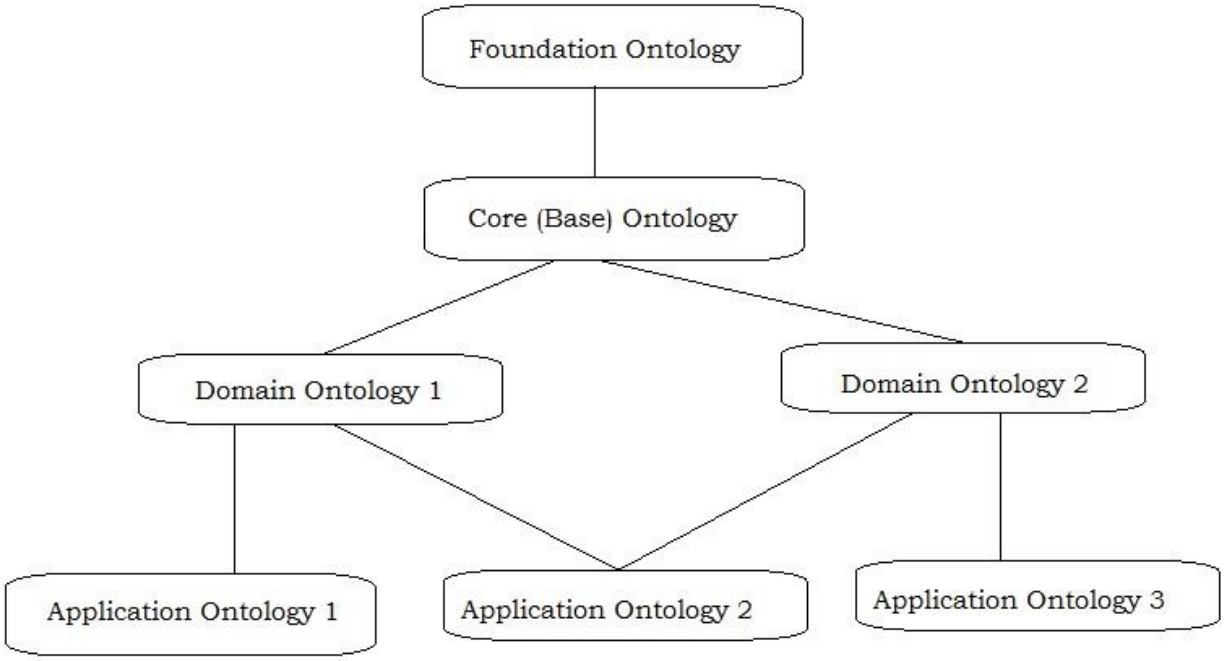


Figure 1: Four Tier Ontological layer

UML conceptual models were prepared before developing the ontology. The UML models help to standardize the representation of the address models from different address standards and to facilitate communication with domain experts, because ontologies are not convenient for human communication (McGuinness & van Harmelen 2004). It is easier for the domain experts to identify conceptual mistakes using UML rather than from OWL ontology.

At the top of the ontological hierarchy is the *foundation* ontology, which describes high level domain-independent geographic information concepts and their relationships. The foundation ontology used in this research is based on RDF vocabulary that provides namespaces for concepts about spatially-located things.

In the hierarchy after foundation ontology is the *base* ontology which is more specific than the foundation ontology, and, in this research, it specifically deals with addressing systems, but it is not specific to a single addressing standard. Developing a base ontology requires good knowledge of the domains (that are involved). Thus domain experts and knowledge engineers must collaborate to prepare the base ontology (De Nicola et al. 2009). In this context the domains are the different address standards which were developed for a variety of purposes and have different conceptual models.

Owing to the cultural attachment of addresses to societies, their purposes, and local legal regulations, it is impossible to develop an ideal universal address standard that would replace all the existing ones. The best option for the future, therefore, is to develop a standard that would facilitate interoperability among the existing address standards. A conceptual model for addresses, with common concepts from various address standards, was developed to serve as the basis for the development of a system that will facilitate address interoperability. Such a model is useful for designing interoperable software solutions that share and manipulate addresses. The Base Address Ontology (BAO) was developed from this conceptual model.

Below base ontology in the ontological hierarchy is domain ontology which has concepts of finer granularity that describe the knowledge within a specific domain. The two addressing domains which are carefully selected to illustrate address data interoperability using ontologies are the INSPIRE (data specification) and UPU-S42. The conceptual models of the standards were converted into UML models which extend the base conceptual model (of the BAO) prior to the implementation of the ontologies. Domain ontology for each model was developed, based on the respective conceptual model.

Application ontology is an ontology designed for a specific use, task, or purpose (Malone & Parkinson 2010), and it usually references domain ontologies to construct its ontological classes, their relationships, and properties. Based on their purposes, ontologies are classified as reference and application ontologies (Menzel 2003). The above-mentioned (foundation, base, and domain) ontologies are reference ontologies, and application ontologies are developed from them.

The dependence of different application ontologies on common domain ontologies increases the number of inferences that can be drawn. Application ontologies that use multiple domain ontologies but share a common base ontology can, furthermore, also be compared, and useful inferences can be made to complement their class extensions.

Four application ontologies were designed to illustrate the use and advantages of address ontologies. The first application ontology (UKdata) references only the INSPIRE domain ontology, and its purpose is to illustrate the use of an application ontology and to serve as a comparison ontology for the second one. The second application ontology (UKdata+UPUdata) references the UPU and the INSPIRE domain ontologies.

After the application ontologies were ready, the inference engine was run which made inferences and assigned class extensions to proper classes. Queries, furthermore, were used to generate new relationships automatically. These demonstrate that semantic ontologies can be used to share and exchange address data across contexts.

The research was concluded with a critical analysis and discussion of the findings.

1.5 Significance of the research

The addresses used for different purposes have their own unique forms. Owing to these differences it is difficult to achieve interoperability across standards. This research aims to investigate an approach for seamless address data, tools and software sharing, and exchange.

Cross checking address data for correctness is difficult if they have different formats. Incompatibility in address data standards can lead organizations to waste their money, time, and staff effort in collecting already existing information at great expense. Hence, the financial benefits of an interoperable system could be significant.

Among the benefits of address data interoperability are sharing and exchanging data, software, and other related tools. Interoperable address data could be used for address validation (cross checking) and, if used properly, could create a synergy for more information. Detailed examples of benefits of address data interoperability are given in chapter four.

1.6 Limitations of the research

Semantic ontologies are expressed by ontology languages such as Web Ontology Language (OWL) whose main purpose is to facilitate the processing of the content of information by computer applications and not presenting information to humans. The ontology language used in this research is OWL which facilitates a greater machine interpretability of digital data (especially Web content). The W3C recommended way of representing OWL codes is to use RDF/XML, which is not user friendly to develop. Thus OWL modeling and editing tools have to be used to avoid mistakes during coding. Tools are, furthermore, required to query and make inferences. The tools available at this stage, however, have many limitations. The main limitations are the quality of query tools, the ability to infer useful information (owing to the quality of data), and interoperability with different DBMSs to extract data.

To make sure that the ontologies and inferences are correct, manual coding and other interferences were done. The best (selected) available software (TopBraid) was used to generate better inferences. For time and simplicity reasons, the research was limited to three address standards, which are the ISO-19160-1 (ISO/TC211/WG7 2012), INSPIRE address specification (INSPIRE/TWG/ Addresses 2009), and UPU-S42 (UPU/POC Addressing Group 2006a). The principles of the research can, however, be applied to any other address standard and the ontology can be extended to accommodate standards in other application domains.

Limitation regarding specific parts and components of the research are discussed in detail in chapter three, section 3.6.

1.7 Definition of terms

The following are important terms used in the dissertation:

Address domain: the jurisdiction of the address standard. For example, SANS1883 is the South African address standard, and INSPIRE data specification is for 27 European countries.

Address Application Domain: the application domain in which the purpose of the address is defined. For example, postal services domain, local government administrative domain, insurance, or banking domain.

Addressing schemas: standardized ways of assigning and collecting addresses.

IRI: sequence of characters from the Universal Character Set (Unicode/ISO10646) that identifies a resource. IRIs are like URIs but with Unicode character capability rather than ASCII alone.

Resource: “can be anything, including documents, people, physical objects, and abstract concepts” that are described by ontology (Schreiber & Raimond 2014).

Class: a resource usually identified by an IRI and used to group resources (Brickley & Guha 2014).

Property: a named resource that describes a relation between a subject resource and object resources (Cyganiak et al. 2014).

Axioms: statements that are asserted to be true in the domain being described. For example, using a *subclass axiom*, one can state that the class *a:Student* is a subclass of the class *a:Person* Instances of concepts - individual (data) (Bock et al. 2012).

1.8 Guide to the remaining chapters of this dissertation

The next chapter discusses the literature survey on the various facets of the research done to achieve the research objectives. The literature survey discusses the theory base for the work done in the dissertation. To facilitate interoperability of addresses across standards, it is important to describe what standards are and how they are developed. The role of standards and standard organizations in facilitating address-related tasks is discussed. The background to the specific address standards used in the research is, furthermore, discussed in this sub-section.

The second chapter also discusses what ontology means in the field of Information (Computer) Sciences. The different components and terminologies of semantic ontology and their functionalities are also discussed in detail. Previous work on semantic web, linked data, and similar researches, and the limitations of semantic ontologies are discussed as background for semantic ontologies.

The third chapter discusses the methodology followed to conduct the research, research tools, data, and software used, the limitations, and ethical considerations of the research.

The next two chapters provide the main body of the dissertation by describing the experiments, findings, and analysis of the research conducted. As mentioned above, a four tier-hierarchy of ontology was developed to achieve the objectives of the research. Chapter four discusses each tier of the hierarchy in detail and demonstrates how the goal of the research is achieved. Chapter five critically analyses and discusses the finding of the research.

Finally, chapter six gives a brief summary of findings, provides the conclusion of the research, discusses contributions to science and technology, and provides suggestions for further studies.

Chapter 2

2 Literature Review

2.1 Introduction

The first chapter introduces the research challenge of facilitating address data interoperability across domains using semantic ontologies as a tool. Like many other scientific research projects, this research builds on top of existing knowledge in related fields of science, i.e. Geoinformation Science and Computer Science (Semantic Web technology in particular). This chapter provides the theoretical and scientific background to the research tools, methods, and approaches used in the research.

Standards are at the root of the research tools, and the methods and technologies used. Hence the chapter starts by discussing the basic principles of standards, what standards are, why, and how they are developed. The chapter also discusses the role of standards in facilitating address data interoperability across domains. The specific standards that are used in the research and why they are used are discussed in more detail.

The research uses semantic approach to solve the problem of address data interoperability. The theory, principles and the components of the Semantic Web are discussed in detail in this chapter. The challenges and limitations of the Semantic Web are also discussed in the chapter, however, the impacts of the limitations and how they are addressed in this research is discussed in the later chapters.

The chapter ends by providing a glance at related work in related fields of science. The literature review on related work provides the context, relative significance, and originality of the research.

2.2 Standards

This section discusses the concepts of standards and standard organizations. It also discusses the specific standards of concern for the research, which are address standards. Questions such as the following are answered in the sections that follow:

- What are standards and standard organizations?

- What are the uses of standards?
- What types of standards exist?
- What are the benefits of standards?
- What is the role of standards in facilitating interoperability?
- What are address standards?
- Which standards are used for the research? Why?

Throughout this document, the role of standards will be discussed at different levels.

2.2.1 Standards and Standards Organizations

It is ironic that the term "standard" does not have a standardized definition. Many standards organizations, individuals, and interest groups have given the term different definitions over time.

The most used definition of a standard is the one used by ISO and IEC, which states that "a standard is a document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context (ISO/IEC Guide 2:2004, definition 3.2). Note - Standards should be based on the consolidated results of science, technology and experience, and aimed at the promotion of optimum community benefits."

The International Organization for Standards (ISO) also defines a standard as "a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose" (International Organisation for Standards -ISO 2014).

IEEE has a similar definition for standards as "published documents that establish specifications and procedures designed to maximize the reliability of the materials, products, methods, and/or services people use every day" (Kraemer & Chandra 2014).

The above definitions are too focused on traditional standardization organizations (van Wessel 2008). (Vries 2006) discussed the definition by other authors such as Jakobs, "a publicly available definitive specification of procedures, rules and requirements, issued by a legitimate and recognized authority through voluntary consensus building observing due process, that establishes the baseline of a common understanding of what a given system or service should offer" and Tassey, "a set of specifications to which elements of products, processes, formats, or procedures under its jurisdiction must conform."

Similar to the variety of its definition, there are variety of standards, in fact Tanenbaum famously said, "The nice thing about standards is that you can have so many to choose from" (Tanenbaum 1988). There are different kinds of standards, for example some are purely technical standards, others are applied standards, such as the Electronic Data Interchange (EDI) which was used for inter-organizational communication in the 1990s (Wang & Seidmann 1995; Damsgaard & Truex 2000).

2.2.1.1 Aim of Standardization

The aim of standardization is to create simplification for society, facilitate interchangeability, facilitate communication, describe symbols and codes (to reduce the effect of different languages), enforce safety, facilitate and protect consumer and community interests, and reduce trade barriers (Sanders 1972).

Historically, standards have been used for different purposes over a long period of time. Some researchers claim that standards "have always been with us" (Cargill, 1989; Cargill & Bolin, 2007). Spivak and Brenner (Spivak & Brenner 2001) give different examples of the need for standards, starting from 3000 BC but also referring to the railway gauges, and the 1904 Baltimore fire incident where equipment from neighboring cities did not work because of differences in hose couplings.

In information technology, the birth of HTML has facilitated the development of the World Wide Web, and, as a result, more powerful Meta languages such as XML were developed. Zhao, Xia and Shaw (Zhao et al. 2007) showed that the invention of XML has boosted the development of B2B standards. The latest information technology trends, such as semantic web, web services, service oriented architectures, cloud computing, etc. are dependent on standards (Kreger 2003; zur Muehlen et al. 2005).

2.2.1.2 Types of Standards

There are different types of standards. Standards typologies can be categorized, based on one of these three perspectives, General, Economic, Technical/IT (Folmer & Verhoosel 2011). De Vries is one of the researchers who used the general perspective of standards to classify the typologies, specifically used the subject matter view for the classification ((de Vries 2006). Others used classification based on the three axes, level (from company, industry, to national, regional, international), subject, and aspect (legislation, testing, inspection, etc.) (Spivak & Brenner 2001). There are other classifications based on the general perspective of standards (Rukanova 2005). Reinstaller (Reinstaller 2008) developed a classification framework for standard typologies based on their economic perspective.

Yeung and Hall classify address standards based on the development process of addresses as open standards, closed standards, *de-facto* standards, international convention, and legislation standards (Yeung & Hall 2007). Standards can also be grouped based on their purposes as abstract standard, Meta standard, concrete standard, and profile.

This research has been more focused on IT standards, and so it was important to consider the classification of standards based on their IT perspective. One of the characteristics for classification was the timing of the standard in relation to the IT product and services, which can be categorized as anticipatory standards, enabling (participatory) standards, and responsive standards (Sherif 2006). IT domain standards could be classified as either implementation or conceptual standards, and they can also be grouped as either product or process standards (Cargill 1989).

2.2.1.3 Standardization Process

The process of standardization is “the activity of establishing and recording a limited set of solutions to actual or potential matching problems, directed at benefits for the party or parties involved, balancing their needs and intending and expecting that these solutions will be repeatedly or continuously used, during a certain period by a substantial number of the parties for whom they are meant” (de Vries 2006).

Studies show that there are many phases in the standard life cycle (Cargill 1995; Krechmer 2006; Söderström 2004). Standards organizations are mostly concerned with the development and maintenance phases of the life cycle, while other organizations (end users) can participate in either or both of the development and adoption phases of the standard life cycle.

2.2.1.4 Standards Organizations

Standard organizations can generally be classified into two main groups, which are the traditional or formal Standards Development Organizations (SDO) and the consortia-based Standards Setting Organizations (SSO) (Simcoe 2007; West 2007). The formal SDOs include:

- National organizations such as SABS, SA, SNZ, ANSI, NEN, SAC, JISC etc.;
- Regional organizations such as CEN, ETSI, SADCSTAN, ARSO, etc.; and
- Global organizations such as ISO, IEC. UPU, ITU, W3C, etc.

For the full words of the acronyms refer to Appendix A.

ISO and IEC have formed a Joint Technical Committee (JTC1) specifically for Information Technology (IT). Many of the important IT standards are, however, produced by the consortia SSOs such as W3C, OGC, OASIS, OMG, and other sector specific standardization organizations which are not related to ISO/IEC JTC1 (Folmer & Verhoosel 2011).

SDOs usually develop open standards, which allows for the voluntary participation of everyone interested and are approved by consensus with each participant having an equal vote. While SSOs usually develop closed (industry) standards where only members of the associated consortium participate in the process which is dominated by the major contributors (West 2007; Yeung & Hall 2007). Sometimes *de facto* standards evolve in the market without the involvement of any standard organization, for example the “Shape” file format of ESRI.

SSOs arguably develop IT standards more quickly than do SDOs (Van Wegberg 2004); some, however, question the need to develop IT standards more quickly than usual and they even question the assumption that SDOs standard developments are slow (Mahonen 2000). The lack of consensus in industry consortia standards impacts on the time of standard development positively, but it might have a negative impact on quality and openness. Sometimes, however, openness is not too important for some standards; in fact, confidentiality and intellectual property rights might be more important (de Vries 2006). The credibility of SSOs can be measured, based on the quality of the standards they produce and also on the way they provide the standards, i.e.

whether they "make the standards cash cows for the organization" (Samuelson 2006). IEEE and IETF are considered some of the best SSOs (Cole 2004; Krechmer 2006).

2.2.1.5 Impact of Standards

In the early twentieth century, the need for standardization increased to meet the demands of modern society for an increased volume of industrial goods and labor-saving devices in every aspect of life (Sanders 1972). Standards related innovation is a major driver of industrial productivity (Zhu et al. 2006). The network effects (or network externalities) and switching costs of standards are the main factors that affect the economic viability of a standard (Blind 2004).

The network effect of a standard is the positive correlation between the number of users of a standard and its utility (Weitzel et al. 2003). Switching costs relate to the cost that is incurred as a result of moving from one standard to another. When the switching cost is too high, "lock-in" occurs (Egyedi 2009). Lock-in is common in the information economy (Shapiro & Varian 1999).

Standards influence the economy by (Hesser et al. 2006):

- Reducing transaction costs;
- Gaining economies of scale;
- Reducing external effects; and
- Influencing the market constitution.

Standards provide a means of communication, guarantee performance, and ensure a wider and more reliable market. The standardization of a product results in a large volume of sales and, thus, the lower cost of goods (Sanders 1972). Semantic Information Systems standards facilitate trade and reduce transaction costs (Blind 2004).

Standardization has also changed the way business is done by (Shapiro & Varian 1999; Sanders 1972):

- Expanding network externalities;
- Reducing uncertainty;
- Reducing consumer lock-in;
- Increased competition in the market rather than for the market;

- Increased competition in price rather than features;
- Competition to offer proprietary extensions; and
- Increased competition for components rather than full systems.

There are still some challenges in quantifying or measuring the impact of standards (Weissinger 2014). The value of standards to someone depends (among other factors) on others using them (Weitzel et al. 2003). Even if the standard is good, if people do not jump on to the wagon to use it, the benefits could be minimal. The gap between individual and collective benefits of standardization can be reduced by communication or by the redistribution of standardization costs and benefits (Folmer & Verhoosel 2011). Another challenge relative to the impact of standards is the effect of network externalities. When a widely-used standard needs to be changed, the effort required becomes considerable (Hanseth et al. 1996).

Choosing the best befitting standard could be difficult owing either to the multitude of choices or the lack of choices. It is rare to get the right number of good quality standards because “if the cost of standardization is too high we face the startup problem, but on the other hand if standardization costs are too low, we will face the inefficient multi-standard equilibrium” (Weitzel et al. 2003). On many occasions, the trend in the market leads towards a single standard eliminating other competing standards (Stango 2004).

In semantic Information Systems standards the effect of market exclusion is not significant because the standards are often designed and modified in joint mode (Folmer & Verhoosel 2011). Semantic Information Systems standards are used to share data and information among firms which usually do not have direct competition, unlike IT product standards that can be used for competitive advantages in the market. The issue of intellectual property rights is also a problem for the IT product standards, but it not so significant with reference to the semantic IS standards (Zhao et al. 2007).

Standards can be used to facilitate interoperability across different companies and generate great value for individual participants or in the overall industry (Zhao et al. 2007). Interoperability has been defined in different ways over the last decades. Interoperability is the ability of different types of computers, networks, operating systems, and applications to work together effectively, without prior communication, in order to exchange information in a useful and meaningful manner (Kosanke 2006). Others define interoperability as the ability of two or more systems or components to exchange information and to use the information that has been exchanged (Legner & Lebreton 2007). There are three aspects of interoperability, semantic, structural (technical), and syntactical. Some also include the organizational aspect in the system (Kubicek & Cimander 2009). This research focuses on the semantic aspect of interoperability.

Semantics deals with the meaning of signs, symbols, words, and phrases in the special sense of how these notifiers relate to reality, how they represent, designate, and signify things (Rukanova 2005). Semantic referencing is necessary because it is not possible to have a single universal standard, even though core components are developed as steps towards standards convergence (Folmer & Verhoosel 2011). Different standards have been developed to facilitate semantic interoperability in various fields. RDF, RDFS, and OWL are some of the standards that are used to express knowledge about objects on the web.

RDF, RDFS, and OWL are all W3C standards that have been widely accepted and used in the semantic web community. More detailed discussion on the previously mentioned standards will follow later in the document.

Having discussed what standards and standards organizations are and how they are developed and operate, the specific standards of concern in this research (address standards) are discussed in the next section.

2.2.2 Address Standards

Addresses have become an important part of our daily life. Addresses are used for various purposes in our daily activities, starting from simple tasks, such as visiting a friend, to more complex tasks, such as public and private service deliveries. Addresses are the most common way of describing a location, often structured in to a spatial hierarchy that describes the location with increasing accuracy (Coetzee 2010). Address is used in physical service deliveries such as postal delivery, utility services, such as water, sewerage, telecommunication, electricity supply, and goods delivery, to abstract services such as credit application, household survey, serving summonses, tax collection, and land administration (Coetzee 2008a; Coetzee et al. 2010).

The aim of this research work was to investigate the use of semantic ontologies to facilitate interoperability across the different address domains. The domain of the address application is determined by the address standard (*de facto* or *de jure*) that was used to assign the address. It is, thus, important to discuss address standards and address standard organizations briefly.

Address standards have been, and are still being, developed by local, national, regional, and international organizations (Coetzee 2008a). Address standards define address schemas that can be used to assign, collect, and use addresses and address data. In the digital world, address standards promote interoperability and the reusability of address-related data and software (Coetzee 2008b).

Address standards are important components of the interoperable system proposed in the research because they determine the domain of address data. Even though many address standards (such as SANS1883) were referred to in the research, the three address standards that were specifically used for the research experiment were INSPIRE address data specification, UPU-S42, and ISO19160-1. The next sections discuss each one of the address standards in more detail.

This section does not discuss how the address standards are used to facilitate interoperability but rather what the specific address standards are. Chapter four discusses how the address standards are used to build ontologies and facilitate interoperability.

2.2.2.1 INSPIRE Address Data Specification

Public authorities across Europe have been struggling with the challenges regarding the lack of availability, quality, organization, and accessibility, and the sharing of spatial information. The lack of information has affected a large number of policies especially environmental policies and activities that impact on the environment (INSPIRE/TWG/ Addresses 2009). To counter these challenges, the European Parliament established the Infrastructure for Spatial Information in the European Community (INSPIRE). Based on the INSPIRE data specification template, the Thematic Working Group on Addresses (TWG-AD) developed the INSPIRE data specification on addresses.

The INSPIRE data specification on addresses is a regional (European) address standard that was designed to facilitate seamless interoperability of spatial data and services. Interoperability in INSPIRE means the ability to exchange and share spatial data and services across the European community "without involving specific efforts of humans or machines". The data specification (standard) was drafted and developed with the participation of many European national and regional organizations.

In the INSPIRE data specification, “address” is defined as “an identification of the fixed location of a property by means of a structured composition of geographic names and identifiers.” In the INSPIRE data specification the objects which can have address are referred to as addressable objects.

The main concept of the data specification is that an address has a “locator”, which distinguishes it from the neighbor addresses, and a “geographic position” which enables an application to locate the address spatially. It is possible for an address to have multiple locators.

An address is uniquely identified by its association with multiple “address components” which define its location within a geographic area (see figure 2). Address components represent spatial identifiers, such as name of street, region, or country. As demonstrated in figure 2, the class “Address Component” has four sub-classes, which are used to define a component. The four sub-classes are administrative unit name, address area name, thoroughfare name, and postal descriptor. This approach to the conceptual description of addresses supports the variety of address systems in the European Union.

In addition to the locator, geographic position, and address components, an address has other attributes, like a unique identifier, status, and a number of life-cycle attributes. Each one of the address components of an address could also have attributes such as unique identifier, status, and life-cycle information.

The information in this section was extracted from the INSPIRE Data Specification on Addresses – Guidelines (2009) which provides a more detailed explanation on how the address standard works, and it also provides sample address data from all the 27 member states.

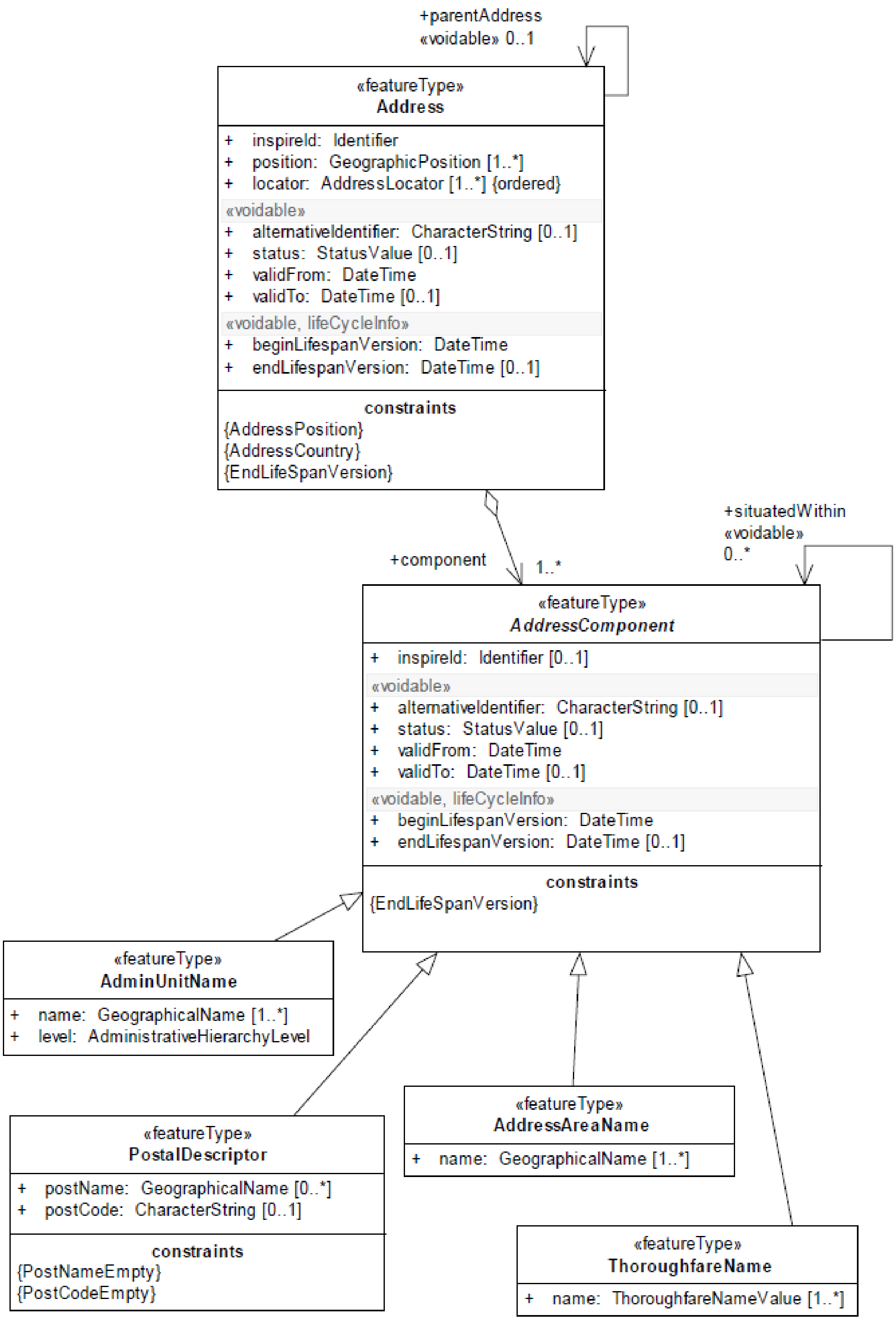


Figure 2: INSPIRE spatial object types (INSPIRE/TWG/ Addresses 2009)

2.2.2.2 UPU-S42

The Universal Postal Union (UPU) is an international organization established in 1874, with its headquarters in Berne, Switzerland. The UPU is a forum for the cooperation among postal sector players from 192 countries. The organization's Standards Board develops and maintains standards that help to facilitate a universal network of products and services (Communication Programme International Bureau Universal Postal Union 2012).

Standard 42 of the Universal Postal Union (UPU-S42), "International postal address components and templates", is an international addressing standard that covers the definition of postal address components and provides templates for international postal addresses (UPU/POC Addressing Group 2006a).

The automation of postal services has become essential owing to the large volume of services required and also to increasing labor cost rates. The majority of postal items carry printed addresses extracted from computer databases (UPU/POC Addressing Group 2006a). To facilitate proper maintenance, sharing, exchanging, and trading of address data, it is important to have a standard. The UPU-S42 standard deals with these tasks and challenges.

UPU-S42 has two parts, one provides the description of postal address components, and the second part provides templates, samples, and instructions on how to use them. The UPU-S42a (Part-A) standard specifies three hierarchical levels of postal address components, which are segment, construct, and elements, sequentially from top to bottom in the hierarchy. There is, however, a fourth level that covers multiple occurrences of elements in an address (see figure 3 and figure 4). According to the standard, a postal address specification can have three optional and one mandatory segment. The mandatory segment is "Delivery Point Specification", while "Addressee Specification", "Mailee Specification", and "Mail Recipient Despatching Information" are optional segments.

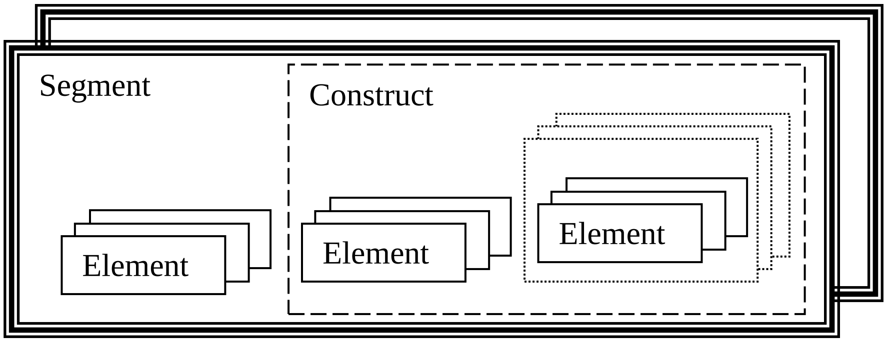


Figure 3: Postal Address Components – Segments, Constructs & Elements (UPU/POC Addressing Group 2006a)

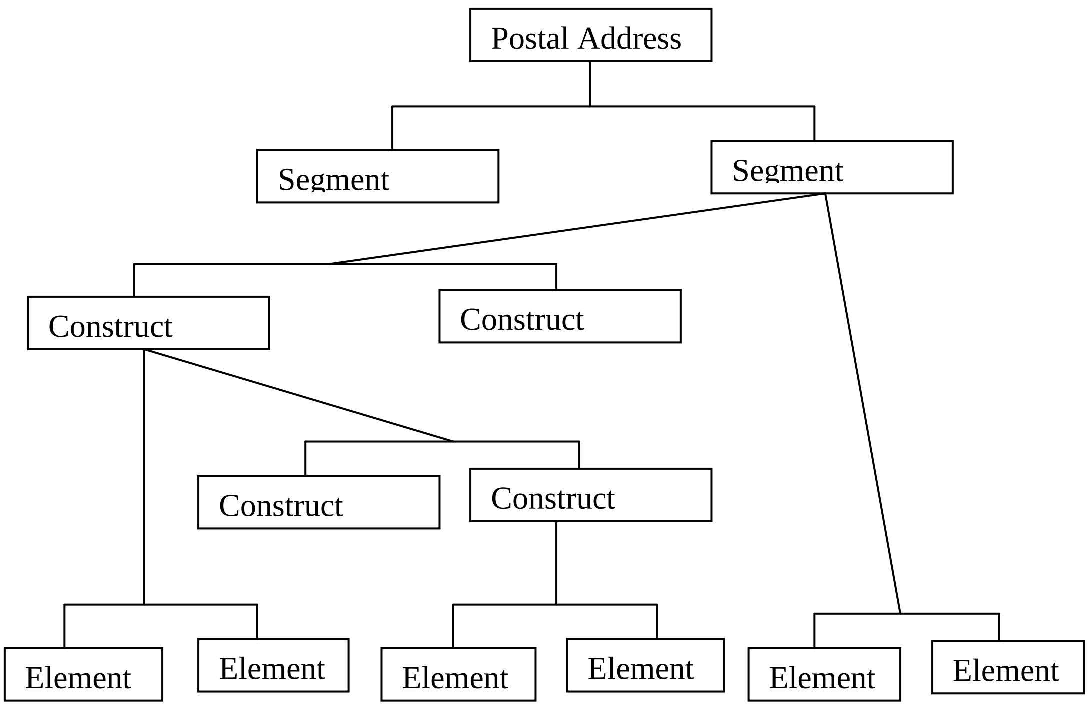


Figure 4: Postal Address Components – Segments, Constructs & Elements (UPU/POC Addressing Group 2006a)

UPU-S42b (Part B) describes address templates for each country (UPU/POC Addressing Group 2006b). This part of the standard is country specific and tends to change frequently. Even though the templates and sample addresses from part-B of the standard are used in this research, it is mostly the first part-A which is discussed and used in depth in the experiments.

2.2.2.3 ISO19160-1

ISO (International Organization for Standardization) is an independent, non-governmental, membership organization established in 1947 with its headquarters in Geneva, Switzerland. ISO is the world's largest developer of voluntary international standards with 163 member countries (as of July 2014). ISO has published more than 19,500 international standards covering almost all aspects of technology and manufacturing (ISO Central Secretariat 2014). As of July 2014, there are 3,368 technical bodies to develop and maintain ISO standards. ISO/TC211, Geographic Information/Geomatics is the technical committee responsible for the development and maintenance of standards in the field of digital geographic information. The aim of the committee is "to establish a structured set of standards for information concerning objects or phenomena that are directly or indirectly associated with a location relative to earth" (ISO/TC211 2005).

ISO19160, Addressing, is a working draft for international address standard developed by ISO/TC211. Stakeholders from different member countries of ISO/TC211 have participated in the process of drafting the standard. South Africa led the working group that actually drafted the details of the standard.

This research started during the early stages of development of the standard, when it was a working draft. The early draft model was, thus, used throughout the experiments. In the process of standardization, the model changed many times, and it might probably go through some changes in the future before it is published. It was, thus, impossible to get the final model without keeping the research on-hold until the standard is finalized, which is a process that takes years to be completed.

ISO19160 was not designed for a unique set of customers or locations; it is rather a bridge between the various existing address specifications. The objective of ISO19160 is to facilitate interoperability across multiple address specifications. In the context of ISO19160, "interoperability" across specifications means "cross mapping of conceptual models, conversion of address data from one specification to another; geocoding of address data by matching address data; and the linking of historic addresses to current address" (ISO/TC211/WG7 2012).

ISO19160: Addressing has five parts (in the stage zero projects). Currently only two of the proposed parts are being developed. The proposed parts of ISO19160 are:

- Addressing – Conceptual model;
- Addressing – Good practices for address assignment schemes;
- Addressing – Quality management for address data;
- Addressing – International postal address components and templates; and
- Addressing – Address rendering for purposes other than mail.

The standard of major importance for this research was the first one "Addressing – Part 1: Conceptual model" referred to from here on as ISO19160-1. ISO19160-1 defines a conceptual model for address information together with the terms and definitions that describe the concepts in the model (ISO/TC211/WG7 2012).

As mentioned above, the main goal of ISO19160-1 is to facilitate interoperability across different address specifications. A Project Team (PT) in a working group (ISO/TC211/WG7 – Information Communities), thus, reviewed many address standards before drafting a conceptual model and it discovered that there are very few high level concepts that exist across all address specifications, but there are some more basic address-related concepts that exist in a variety of forms across most of the specifications.

The ISO19160-1 conceptual model has three mandatory classes, Address, AddressableObject, and AddressComponent (see figure 5). The conceptual model also includes two optional classes, AddressSpecification, and ReferenceSpatialObject.

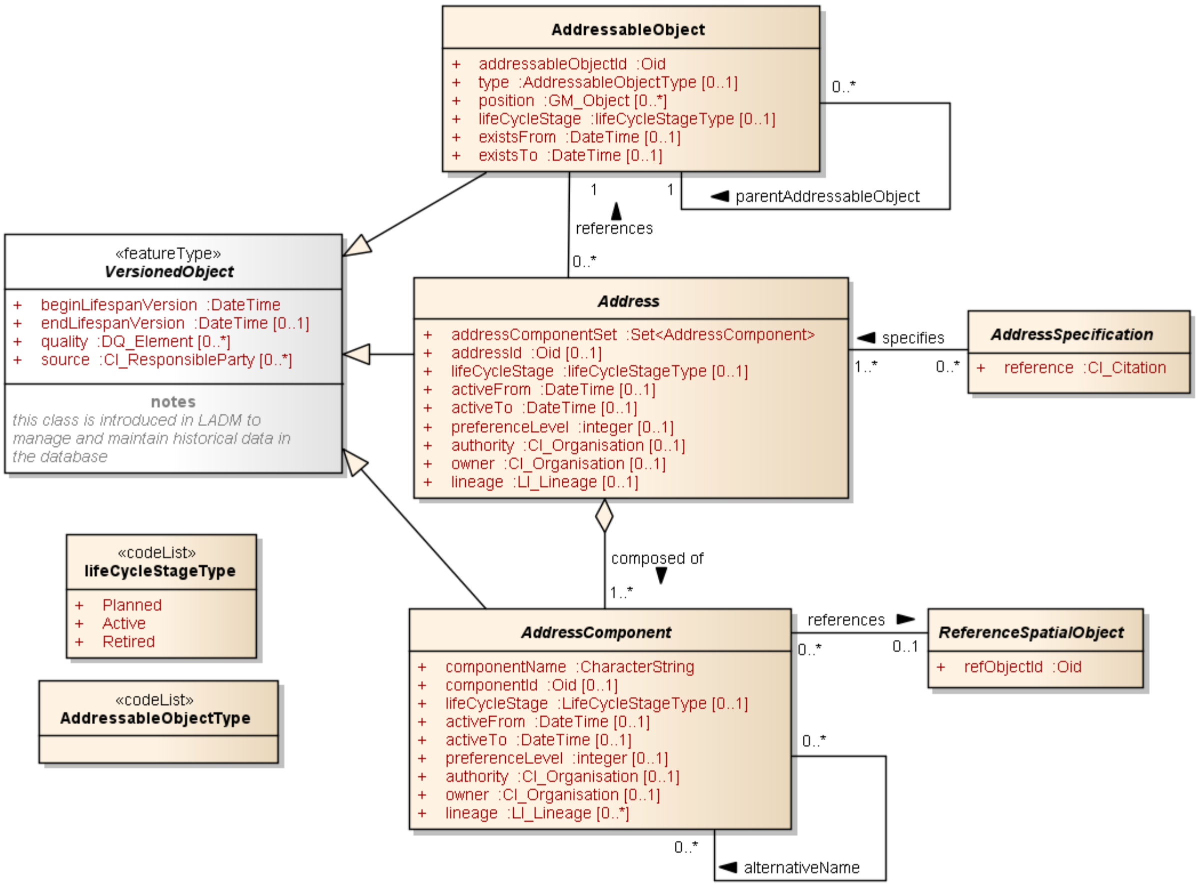


Figure 5: Conceptual Addressing Model (ISO19160-1, 2012)

The conceptual model of ISO19160-1 is an international address model that needs to be extended by local profiles to be used in the process of interoperability. The conceptual model of ISO19160-1 is presented in UML format supported by a textual description document that describes the components. The profiles for local standards need only to extend the UML model with new classes or just attributes to the existing classes. Figure 6 shows a South African profile for ISO19160-1.

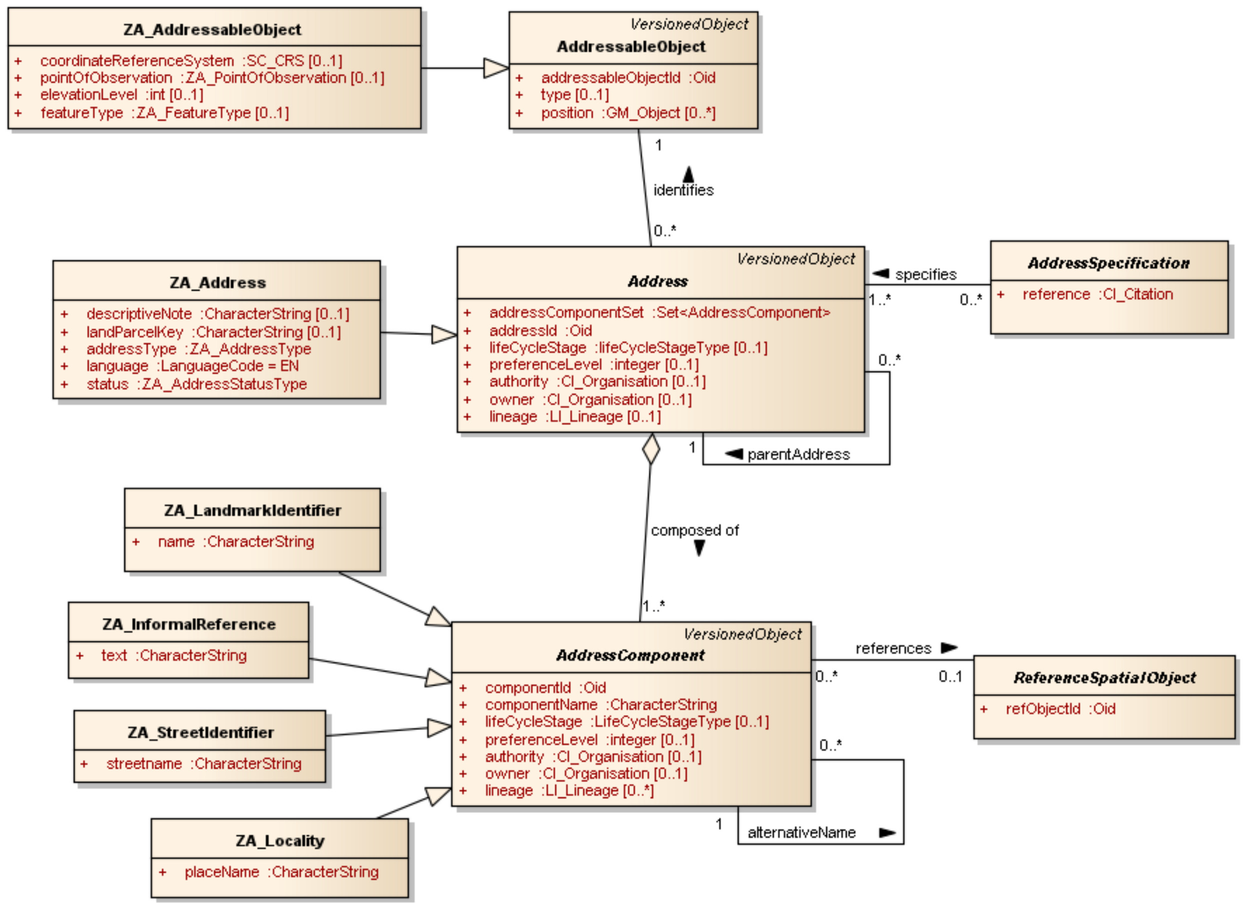


Figure 6: South African Informal Addresses (SANS1883-1) Profile for ISO19160-1
(ISO/TC211/WG7 2012)

The conceptual model has optional temporal attributes that are components of the three mandatory classes (Address, AddressableObject, and AddressComponent). The model also includes codelists for lifeCycleStageType and AddressableObjectType (see figure 5). The codelists are completed (extended) by the local profiles (see figure 5 and figure 7). Additional codelists that are specific to the local model can also be added to the profile.

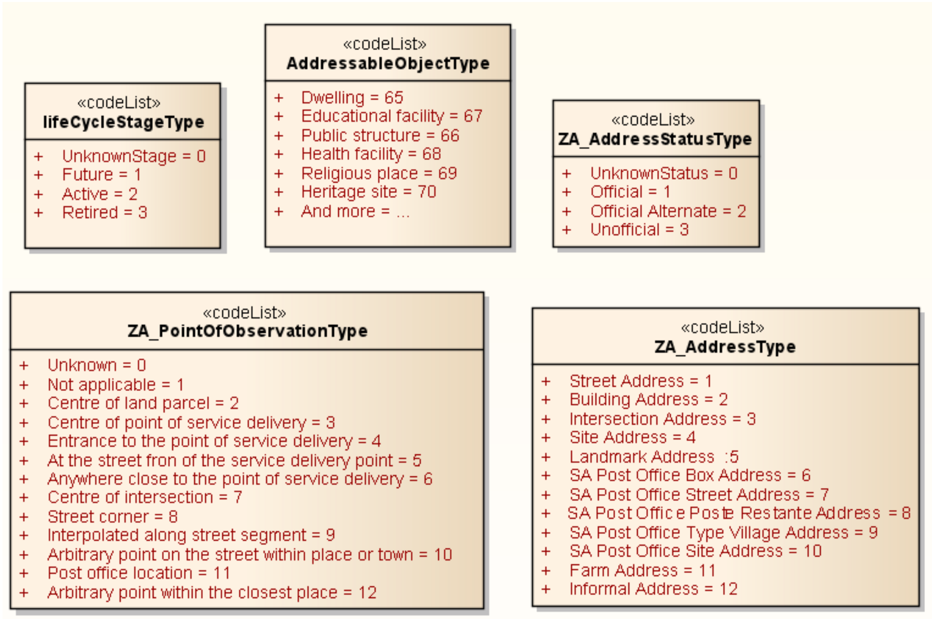


Figure 7: South African Profile codelists for ISO19160-1 (ISO/TC211/WG7 2012)

Section 4.3 discusses how ISO19160-1 was used to develop an ontology that facilitates the interoperability of address data across different domains.

2.3 Semantic Ontology

This section discusses the concept of ontologies and knowledge representation. It also discusses the standards and recommendations that are used to develop ontologies and the semantic environment (Semantic Web) where ontologies could be utilized to greater extent. The challenges and limitations of the Semantic Web and ontologies are discussed at the end of the section.

Questions such as the following are answered in the sections that follow:

- What are semantic ontologies?
- What is the Semantic Web?
- Which standards and recommendations are used in the Semantic Web?
- What are the challenges and limitations of the Semantic Web and ontologies?

2.3.1 Semantic Web

The Semantic Web is a movement, led by the W3C community, to provide a common framework that allows data (with its semantics) to be shared and reused across application, enterprise, and community boundaries (Hawke et al. 2013). The Semantic Web provides common formats for integration and combines data from diverse sources (see figure 8 below).

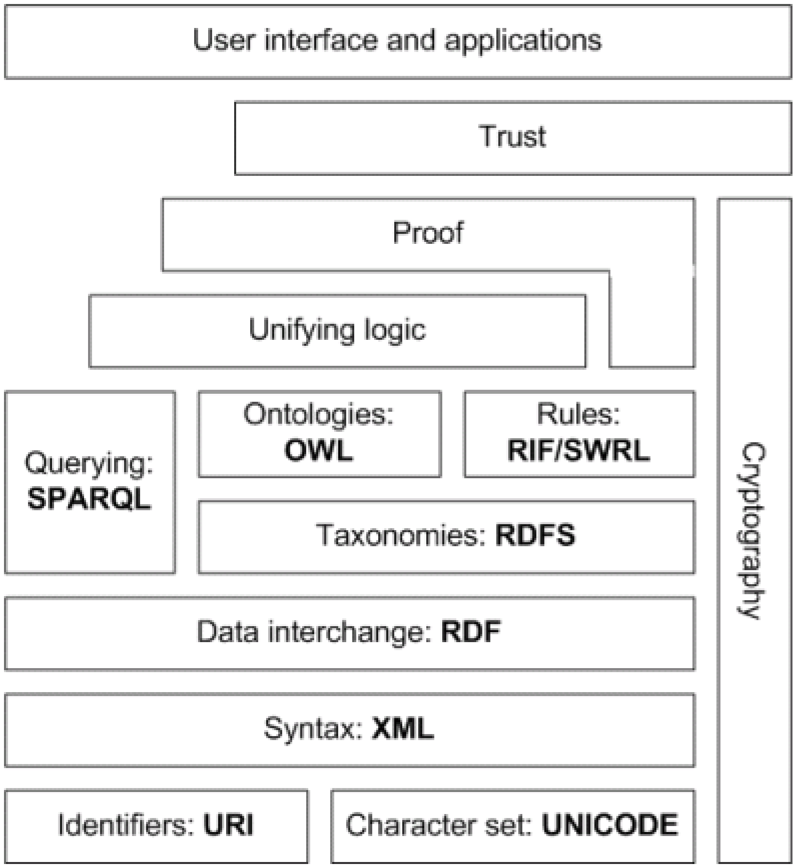


Figure 8: Semantic Web Stack (Obitko 2007)

As figure 8 demonstrates, the Semantic Web is an extension of the hypertext web. The architecture of the Semantic Web (i.e. the Semantic Web Stack) is evolving as the layers of the architecture become more standardized and accepted. The Semantic Web is based on the Resource Description Framework (RDF) (on top of the hypertext web).

The main purpose of the Semantic Web is to create semantic knowledge models that enable machines to understand what data means, where to find it, and where it fits in to the knowledge model. The Semantic Web enables machines to query for data based on its semantics and to make inferences of useful information. The Semantic Web is based on notions of information sharing like; the "Anyone can say Anything about Any topic" (AAA) logic, the one world assumption, and the "no unique name" principle, which provide knowledge synergy (Allemang & Hendler 2011).

Humans use their intelligence to retrieve data, move it from one web page (or application) to another (based on its meaning), do subsequent processing on another web page (or application), and obtain useful information. For example, to find out the list of players who are younger than 23 years old, born in Africa, and currently playing in the Barclays Premiership, one has to visit the website of every team in the league, get the list of players and their date of birth information, and process the data, using an application, to obtain the result. Machines, however, cannot perform such a process without human interference because the meaning of the data is not visible to the machines. Some of the components of the Semantic Web Stack are, thus, designed to store the semantics of the data in machine-identifiable (understandable) format.

Knowledge modeling is a process of creating a computer interpretable model of knowledge or standard specifications about a kind of process and/or about a kind of facility or product. Knowledge modeling is important to represent the semantics of the data for machine processing. Ontologies are used to represent knowledge formally within a domain. Ontologies (in Information Sciences) are explicit specifications of conceptualization (Gruber 1995). Ontologies are used to model the structure of a system as a set of concepts and relations that exist among them (Antoniou & Van Harmelen 2004). Ontologies have been used as a form of knowledge representation in the Semantic Web, Artificial Intelligence (AI), software engineering, and other fields of computational sciences.

This research study was about sharing address data from various sources across the globe. Hence the Semantic Web was selected as a medium to achieve the goal in a more internationally standardized and accepted manner. One of the most important building blocks of the Semantic Web is ontologies (Maedche & Staab 2001). Semantic ontologies are expressed with ontology languages that are XML compatible and support the definition of ontology vocabularies that are identifiable by URI (IRI) references.

The Semantic Web programming has two main aspects, which are knowledge representation and application integration (Hebeler et al. 2009).This research study focuses on knowledge representation for address data to facilitate interoperability across domains. It does, however, briefly discuss the application integration aspect of it. The core W3C specifications for knowledge representation on the Semantic Web are RDF (RDFS) and OWL. The following sections explain these W3C recommendations and the query language (SPARQL) used to query ontology on the Semantic Web.

2.3.2 RDF

The Semantic Web has three complementary representation languages, which are the Resource Description Framework (RDF), RDF Schema (RDFS), and the Web Ontology Language (OWL) (Allemang & Hendler 2011). The RDF serves as the foundation for the more powerful features of RDFS and OWL. RDF is a framework to describe resources, and, since semantic ontologies are made up of resources, all other Semantic Web standards build on this foundation (Hebeler et al. 2009). As demonstrated by the Semantic Web layer cake (figure 8), RDF relies on the infrastructure of the hypertext Web, and it extends the Web features to represent semantic information in the Web (RDF Working Group 2014).

The Web is currently made up of documents linked to one another (each other is for two only) without any concrete relation to the content a document holds. It, thus, requires human interaction to make the connection between the real world representations of the contents of the document. For example, on the webpage of Professor Coetzee (the supervisor of this research) the link to the modules she lectures (COS787 and GIS120) are provided; the Web, however, does not provide a mechanism for machines to identify what the entity "Prof Serena" is or what the link to COS787 means. It requires human intervention to interpret the meaning of the content, and so it becomes impossible to query based on the meaning of the concepts. For example, it would be impossible to query the Web for the number of modules "Prof Serena" lectures, because the Web does not know the meaning of Web content.

In the Semantic Web, information about things (in the universe) is represented as a set of assertions called statements or triples that contain three parts, subject, predicate, and object (Cyganiak et al. 2014). Statements on the Semantic Web are expressed using RDF. A set of RDF statements make an RDF graph which can be visualized as a network of nodes connected by directed arcs. A statement in an RDF graph is represented as node-arc-node triple (see figure 9). The nodes can be URIs/IRIs, literals, or blank nodes.

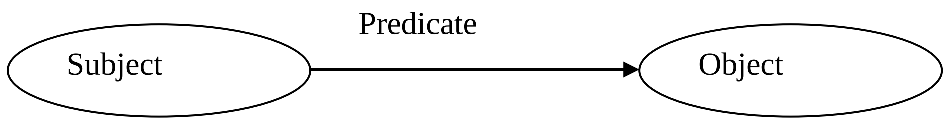


Figure 9: Basic RDF statement format

Figure 9 is the simplest form of an RDF graph containing a single statement that says some relationship (predicate) holds true between the subject and object resources. A resource in RDF is any IRI or literal that represents something in the domain of the RDF. Multiple RDF statements can be joined to one another to make a bigger RDF graph. The object resources of one statement can be the subject of another statement. For example, look at the following English statements and the RDF graph for the statements (Figure 10).

Alan lives in Pretoria. Pretoria is capital city of South Africa.

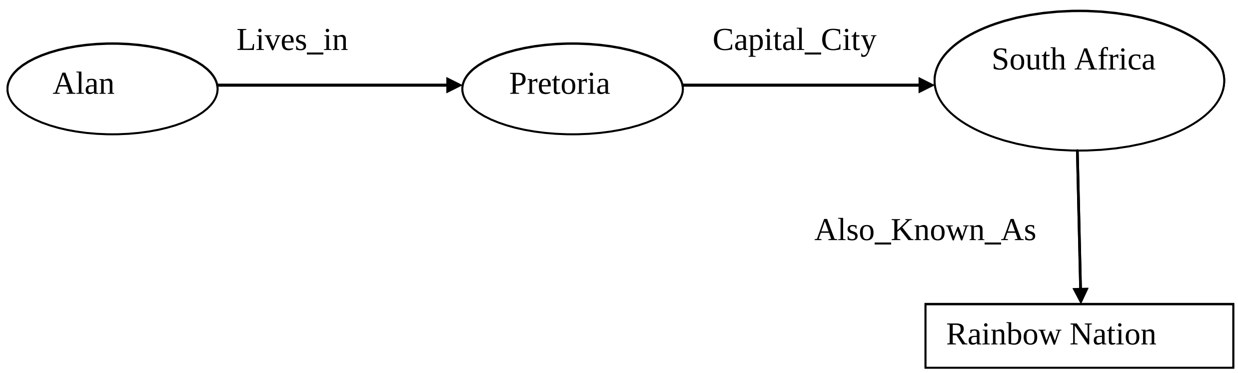


Figure 10: RDF graph with three statements

The RDF graph in figure 10 has multiple statements. The nodes on the graph are the subjects and objects of statements, while the arcs are predicates that relate them. The subject of a triple can be either an IRI or a blank node, the predicate is an IRI, and the object can be an IRI, a literal or a blank node. An IRI (Internationalized Resource Identifier) denotes a referent of a resource; a literal denotes literal value which is a concrete value of certain datatype. A blank node, however, does not identify a specific resource. A blank node in a statement simply indicates that a relationship exists with an object (resource) but does not explicitly name the subject (Cyganiak et al. 2014).

The ovals in figure 10 indicate resource referents, the rectangle indicates literal-value, and the arcs indicate the relationship between them. Such an RDF graph model is the simplest way to visualize RDF statements. In figure 10, the literal-value is "Rainbow Nation" of type XSD string. Literals denote concrete values with datatypes that define their range, for example XSD datatypes, such as strings, dates, and numbers. The referents, on the other hand, are just names that identify objects, acts, or concepts (Hebeler et al. 2009). Referents have an IRI form which has global scope and is a unique identifier. RDF specification only defines IRI for certain referents such as `rdf:property`, `rdf:type`, and other RDF vocabulary referents. The RDF specification, however, does not define the IRI of application-specific classes and properties. Other (extension) specifications are, thus, needed to define the application-specific referents.

2.3.3 RDF Schema

One of the most important extensions to RDF specification is the RDF Schema (RDFS). RDFS provides data modeling vocabulary for RDF data (Brickley & Guha 2014). RDFS does not provide actual description to application specific referents; instead, it provides a framework to describe such referents.

RDFS provides a mechanism to describe a group of related resources in the universe, such as People, Capital-City, and Country, using the class and property system. RDFS specification also has vocabulary resources that can be used to specify the characteristics of other resources. For example, the `rdfs:domain` and `rdfs:range` can be used to determine the domain and range of a property respectively.

Unlike in Object Oriented Programming (OOP), RDFS properties do not necessarily belong to a single class. In fact, they are described by the classes of resources to which they apply (Brickley & Guha 2014). For example, the "reference" property (in the application-domain of addresses) can be defined with domain "Address" and range "Addressable Object", where "Address" and "Addressable Object" are `rdfs:classes`. The property "reference" is an IRI, and it has domain and range restrictions. It, however, does not belong to any of the classes. This is a property centric approach (as opposed to the class centric approach of OOP), where the properties determine relations between classes. RDFS classes are also IRIs which can be easily extended by using more (application specific) relations or restrictions. Good examples of such extensions are demonstrated in Chapter 4, where classes from a base ontology are extended by domain specific ontologies.

The W3C provides the namespace IRIs for RDF and RDFS vocabulary (see table 1). Conventionally the prefixes `rdf` and `rdfs` are used to represent shortly the full namespace IRIs.

Namespace Prefix	Namespace IRI	RDF vocabulary
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	The RDF built-in vocabulary
rdfs	http://www.w3.org/2000/01/rdf-schema#	The RDF Schema vocabulary
xsd	http://www.w3.org/2001/XMLSchema#	The RDF-compatible XSD types

Table 1: RDF, RDFS and XSD namespace prefixes and IRIs

The RDFS specification does not have enough vocabulary to describe the meaning of all possible classes and properties. To describe the meaning all objects and concepts (in the universe) more vocabulary is required from Semantic Web languages such as OWL and inference rule languages such as Semantic Web Rule Language (SWRL). The next section provides a brief introduction to OWL.

2.3.4 OWL

The Web Ontology Language (OWL) is a Semantic Web language that extends the RDFS with vocabulary that can be used to represent rich and complex knowledge about things, ideas, or objects (in the universe) and the relationship between them (Hitzler et al. 2012). The knowledge expressed with OWL ontology can be reasoned with computer programmes to verify consistency of the knowledge or to make implicit knowledge explicit (Hitzler et al. 2012). OWL adds restrictions to RDFS and RDF specifications to make more computationally decidable processing and reasoning (Antoniou & Van Harmelen 2004). OWL is computational logic-based language.

OWL1 first became W3C recommendation on 10^{th} February, 2004 (McGuinness & van Harmelen 2004). Even though OWL1 was successful, it had certain weaknesses in its design. The most noticeable weakness of OWL1 is the lack of built-in datatypes, because it relies on the XML schema (XSD) list of built-in datatypes. Thus OWL1 was extended with a more powerful OWL2. OWL2 adds several new features to OWL1, including increased expressive power of properties, extended support for datatypes, simple metamodeling capabilities, extended annotation capabilities, and some syntactic changes (Patel-Schneider 2012). OWL2 officially became the W3C recommendation in October 2009 and the second edition of OWL2 was recommended on 11th December, 2012. OWL2 is backward compatible to OWL1 (Hitzler et al. 2012). Throughout this document the acronym OWL refers to OWL2 rather than the original OWL1. The software tools used in the research (TopBraid Composer) support OWL2. OWL2 uses the same namespace as OWL1. OWL ontologies also use RDF, RDFS. and XML schema namespaces (see table 2).

Namespace	**Prefix**
http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf
http://www.w3.org/2000/01/rdf-schema#	rdfs
http://www.w3.org/2001/XMLSchema#	xsd
http://www.w3.org/2002/07/owl#	owl

Table 2: Namespaces used in OWL ontology

OWL ontologies are core components of the Semantic Web because they are used to model knowledge domains. RDF graphs can represent resources and literals with simple direct relationships but cannot categorize resources. RDFS can define terms (to describe group of resources) and identify a hierarchical structure of concepts, but it cannot represent the domain specific meaning of concepts without OWL extension (Antoniou & Van Harmelen 2004).

The Web has a distributed network of documents; the Semantic Web builds on top of the existing hypertext Web. Hence the knowledge models of the Semantic Web are also distributed over the net. As a component of the Semantic Web, and also because of its inherent dependence on RDF, OWL fully supports distributed knowledge models. Owing to the distributed nature of the knowledge models, OWL uses the open world assumption in order to make inferences. The open world assumption states that what is not known to be true is not necessarily false; it is just unknown. The closed world assumption is the opposite; hence, by such an assumption, what is not known to be true must be false. The two assumptions lead to different inferences on the same set of data and rules. A relational database and OWL ontology examples are given below to demonstrate the two assumptions.

RDBMSs use closed world assumption. For example, consider a database with a table called "Address" (see Table 3), that has "Name" and "Home Address" attributes. Let us assume that every person has a unique name and that a person can have only one home address. The absence of a record with a certain unique "Name" and "Home Address" pair values would be interpreted as false record; in other words, as if the person does not live at that address. For example, in the database (closed world assumption) the absence of the address information of "Samuel" (a person) would imply that "Samuel" does not have an address, but, in an open world assumption system, it would simply mean that it is unknown whether he has an address.

Address	
Name	**Home_Suburb**
Sean	Lynnwood
Teboho	Hatfield
Victoria	Queenswood
. . .	

Table 3: "Address" table for Student Address Database

The effect of the closed and open world assumptions is not limited to whether the result is "no" or "unknown"; it also affects the way inferences are made based on existing data. RDBMS strictly checks for the existence of a record in a database and makes inferences based on the closed world assumption. For example, if the record with the pair "Sean" and "Brooklyn" (from statement "Sean's home address is Brooklyn") was to be added to the "Address" table (Table 3), the RDBMS would flag an error message, because of the rule that every person can have only one address.

OWL ontologies make one world assumptions, where the absence for the knowledge about the truth of a statement makes it unknown. Consider the previous example of "Address" information, where "Sean's home address is Brooklyn" to the knowledge model that already has the statement "Sean's home address is Lynnwood". Since there is a rule that specifies a person can have only one home address, the OWL ontology would infer that "Lynnwood and Brooklyn are the same". The logic behind the inferred statement is, "If a person can only have one home address, and

Sean's home address is Lynnwood and Brooklyn, then Lynnwood and Brooklyn must be the same address."

Note that RDBMS (with closed world assumption), considered "Lynnwood" and "Brooklyn" different, but the ontology (with the open world assumption) considered them to be the same. This problem is called the "unique name assumption" problem. The unique name assumption is that resources identified by different identifiers (IRIs) are different. Owing to the nature of the Semantic Web, this assumption is unrealistic for ontologies. Different people use their own (different) IRIs to describe the same resource. Restrictions could, however, be added to an ontology to resolve the problem of unique name assumption. For example, if the list of all suburbs is provided, then a restriction explicitly stating that every one of them is unique to others could be added. After adding the restriction (hence the statement, "Lynnwood is different from Brooklyn") the ontology inference would now generate an inconsistency. The logic behind the inference would be, "if a person can have only one home address, and if Sean's home address is Lynnwood and Brooklyn, then Lynnwood and Brooklyn must be the same, but, hold on, Lynnwood is different from Brooklyn, so they cannot be the same. Something is wrong!!!"

OWL ontology documents are made up of an ontology header (optional), annotations (that are non-semantic descriptions), classes, properties, individuals, and datatype definitions. Owing to the underlying principles of OWL, the concepts and instance data are mixed in the same ontology document. In this research study, multi-tier hierarchical ontologies were used to separate the ontology vocabulary from the instance data. Chapter four discusses the reason for the separation of ontologies in multiple tiers and how it was done.

Ontology documents have an optional header, which is a resource that describes the ontology. If an ontology imports other ontologies to reference, the information about the imported ontologies must be included in the header so that any ontology that imports the current ontology would be able to dereference the IRIs of resources from the original ontology. For example, if a domain ontology about "South African Addresses" imports the Base Ontology for Addresses (BAO) (developed for this research and discussed in Chapters three, four, and five in detail), then any SA address application ontology that uses the South African Address ontology needs to identify the BAO concepts (classes) used in the domain ontology. Chapter four has examples of this nature.

The fundamental building blocks of OWL ontology are classes, properties, and individuals. Ontology is the formal representation of knowledge as a set of concepts, their instances, and the relationship between concepts, within a specific domain (Hitzler et al. 2012). In order to build a knowledge base for a domain it is important to define the vocabulary of the domain. A vocabulary is a collection of unambiguously defined concepts, where every concept is unique unless explicitly identified as redundant (Hebeler et al. 2009). OWL classes and properties are used to define domain vocabulary.

Besides the terminological (conceptual) knowledge from the vocabulary, ontology might also contain assertional knowledge that deals with concrete objects (individuals) of the domain (Hitzler et al. 2012).

OWL class is a special kind of resource that represents concepts (real or abstract) in the world. OWL individuals are resources that are instances of a class. OWL individuals become members of a class by explicitly making a direct axiom that establishes the relationship between the class and the individual. For example, the class "Car" can represent common properties of a group of resources with similar properties (see figure 11). Making axioms such as "SUVs are type of Car" and "Sedans are type of Car" makes the "SUV" and "Sedan" resources instances of "Car".

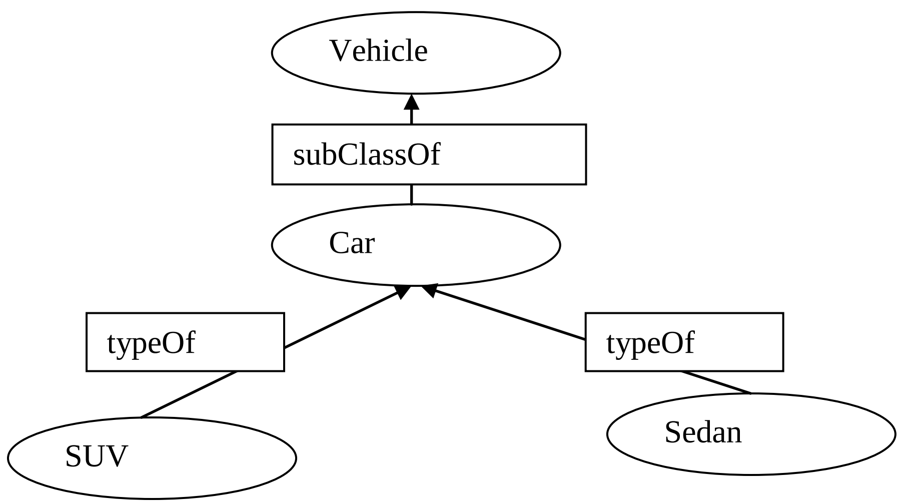


Figure 11: OWL ontology classes represent group of similar resources

Another way for an individual to become a member of a class is as a result of a reasoner's inference from existing axioms. For example, if there are axioms "Car is sub class of Vehicle" and “Automobile is same as Car”, then the reasoner infers (a new OWL axiom) that “Automobile is sub class of Vehicle” (See figure 12).

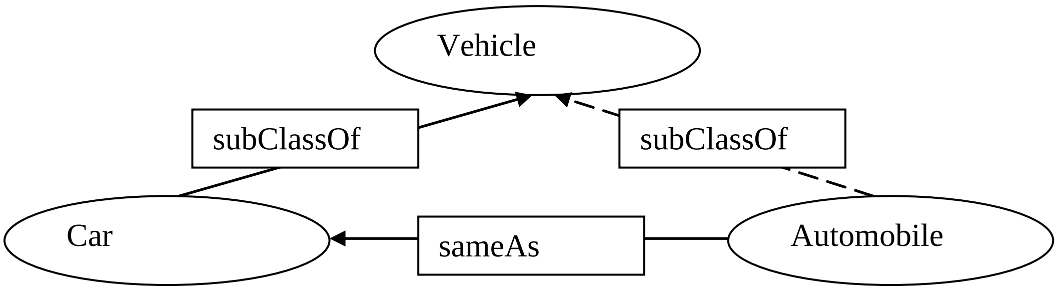


Figure 12: OWL axioms, dashed lines indicate inferred axiom

OWL property is special kind of resource that is used as a predicate in OWL axioms to describe individuals of a class. OWL property could be either *object property* or *datatype property* based on the type of individuals it describes. Object property relates two individuals (IRI resources), while a datatype property assigns a data (literal) value to an individual. Figure 13 demonstrates the two types of OWL properties. The property “same as” in the axiom “Car is the same as Automobile” is an object property because both “Car” and “Automobile” are objects. The property “invented in” of the axiom “Car is invented in 1886” is a datatype property because the individual in this case is a literal value of type date.

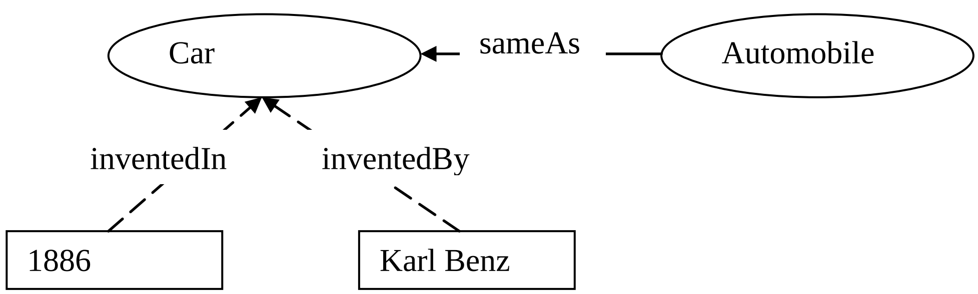


Figure 13: OWL Properties, solid line indicates object property, dashed lines indicate datatype properties

More detailed discussion on OWL ontology vocabularies is left for chapters four and five, when the research ontologies are discussed.

2.3.5 Challenges and Limitations of Semantic Web Ontologies

The Semantic Web is still evolving and is incomplete. The recommendations (standards) for the top layers of the Semantic Web Stack (see figure 8) are still under development (Bratt 2008). W3C recommendations on rules, transformations, deployment, application spaces logic, proofs, and trust need to be developed. The real benefits of the Semantic Web will be harvested once the Stack is fully developed and the Semantic Web is furnished with large volumes of semantic data.

The challenges of the Semantic Web are of two kinds, knowledge representation and application integration challenges. Some of the challenges of the Semantic Web are (Bratt 2008) (Polleres et al. 2010):

- Providing information in Semantic Web friendly ways;
- Exposing existing data stores as RDF;
- Automated and human assisted tools to create RDF stores, ontologies, mappings;
- Making data accessible to people and programmes;
- Usable interfaces to masses of semantic data;
- Search, filtering, aggregation, processing, graphics;
- Access via multiple modes, multiple devices;
- Addressing broader operational and social needs;
- Universality, quality, provenance, versioning, safety, privacy, access control, authorization, trust;
- Too little cross domain vocabulary to answer complex queries; and
- Varying quality of linked data on the Semantic Web.

The Semantic Web is a Web of data, and Linked Data lies at the heart of it. Linked Data is set of data (of different types such as dates, place names, events, book titles, movie titles, actor names, chemical properties, political party names, and any others) linked to one another in such a way that they can be accessed and processed by Semantic Web technologies (such as RDF, GRDDL, POWDER, RDFa, OWL, SPARQL, etc.). There is large amount of data on the hypertext Web. It is, however, mostly not available in a Semantic Web friendly way (in RDF format). There is too little Linked Data to generate useful information using Semantic Web reasoners or complex SPARQL queries (Polleres et al. 2010). Considering the current size of the Web, the effort required to create a comparable size of Semantic Web is tremendous. Exposing the existing information on the Web to the Semantic Web would require some kind of automated mapping to RDF or OWL data from the traditional data sources (such as XML and RDBMS) without affecting the original content on the Web. Languages such as R2RML, which is W3C recommendation (Das et al. 2012), can be used to facilitate the process of mapping from relational DBs to RDF datasets. There is an effort by researchers to develop a query language (called XSPARQL) that combines XQuery and SPARQL capabilities (Polleres et al. 2010). XSPARQL is expected to facilitate the automated transformation of XML based Web data to RDF/OWL data.

The coverage of the vocabularies on the Linked Open Data (LOD) cloud is growing; the traditional hypertext Web is, however, growing at much faster rate. The vast majority of Web developers ignore the Semantic Web technologies. There is, thus, a need to introduce an easy way to help them to incorporate semantics (RDF/OWL) to Web development tools. To develop vocabularies for more domains would require the participation of domain experts and software engineers. The standards (W3C recommendations), languages (RDF/OWL), and principles (such as open world assumption) of Semantic Web applications are different from traditional Web-based or stand-alone programming principles. More people, thus, need to learn the skills to develop Semantic Web applications.

The Semantic Web community can gain lessons from the techniques and tools used to facilitate storing and searching of data on the Web efficiently. As the size of the Semantic Web becomes bigger and bigger, more powerful reasoning and semantic searching tools need to be developed in order for the Semantic Web to become useful in the day-to-day activities of people. There are some challenges that need to be tackled, such as data integration, scalable storage (Tummarello et al. 2007), indexing (Delbru et al. 2010), and efficient querying (probably using query federation) (Florescu & Kossmann 1999).

2.4 Related Work

The main goal of this research is to investigate and find an approach to facilitate address data sharing and interoperability across domains using semantic ontologies. Hence, in this section, current researches, implementations, and practices of address data sharing and also the role of ontologies to facilitate cross domain interoperability are discussed in order to provide a better context and show the significance and originality of this research.

Standards in general and address standards in particular were discussed in Section 2.2. Address standards are the first step towards address data sharing and exchange across domains. There are many address standards at national, regional, and international levels. These address standards have been used for different purposes by various individuals or organizations. The literature shows that address standards have different address models (South African Bureau of Standards 2009) (INSPIRE/TWG/ Addresses 2009) (UPU/POC Addressing Group 2006a), depending on their purpose, historical background, and culture. Even though this is the case, there are significant benefits from sharing address data. Owing to their underlying address models, it has been difficult to share or exchange addresses amongst multiple domains. Since the end of the last decade, ISO/TC211 has been working to develop an address standard that would facilitate interoperability (ISO/TC211/WG7 2012). Once published, the ISO19160-1 will serve as a foundation for systems that facilitate interoperability. This research study was conducted to investigate an interoperable system that uses semantic ontologies based on the ISO19160-1 standard as a foundation conceptual model.

The task of address data sharing becomes more complex because of the diversity of the digital address data storage mechanisms. The same address would be represented in different formats and possible with different attributes, based on the address standard used. One thing that may be common is the meaning of the address data; whether an address is represented in one format or another it still is the same address or addressable object. The weapon of choice to tackle the challenge of address data sharing was, thus, to use the semantic ontology approach and capture the meaning of the data rather than its format and syntax.

Similar approaches have been used to facilitate interoperability in many other domains, including, enterprise (Chen et al. 2008), military, software engineering, telecommunications, e-government (Ojo et al. 2009), construction, publishing (Angrosh et al. 2014), social media, and medical industry (Hammond 2008; Dolin & Alschuler 2011). No research work on semantic interoperability of address data has, however, been found. Since the problem of interoperability is common across multiple domains, researchers have been working on researches to formalize the process of semantic information interoperability in a scientific way (Naudet et al. 2010).

One of the most effective and fascinating Semantic Web projects is the Linked Open Data (LOD) project that enables users to use the Web to relate data based on the meaning of the contents and eliminating or lowering barriers to linking data (Heath & Bizer 2011; Görlitz & Staab 2011). The LOD cloud has vocabularies from many domains (see figure 14), including publishing, life sciences, media, geographic, social networking, government, linguistics, and others.

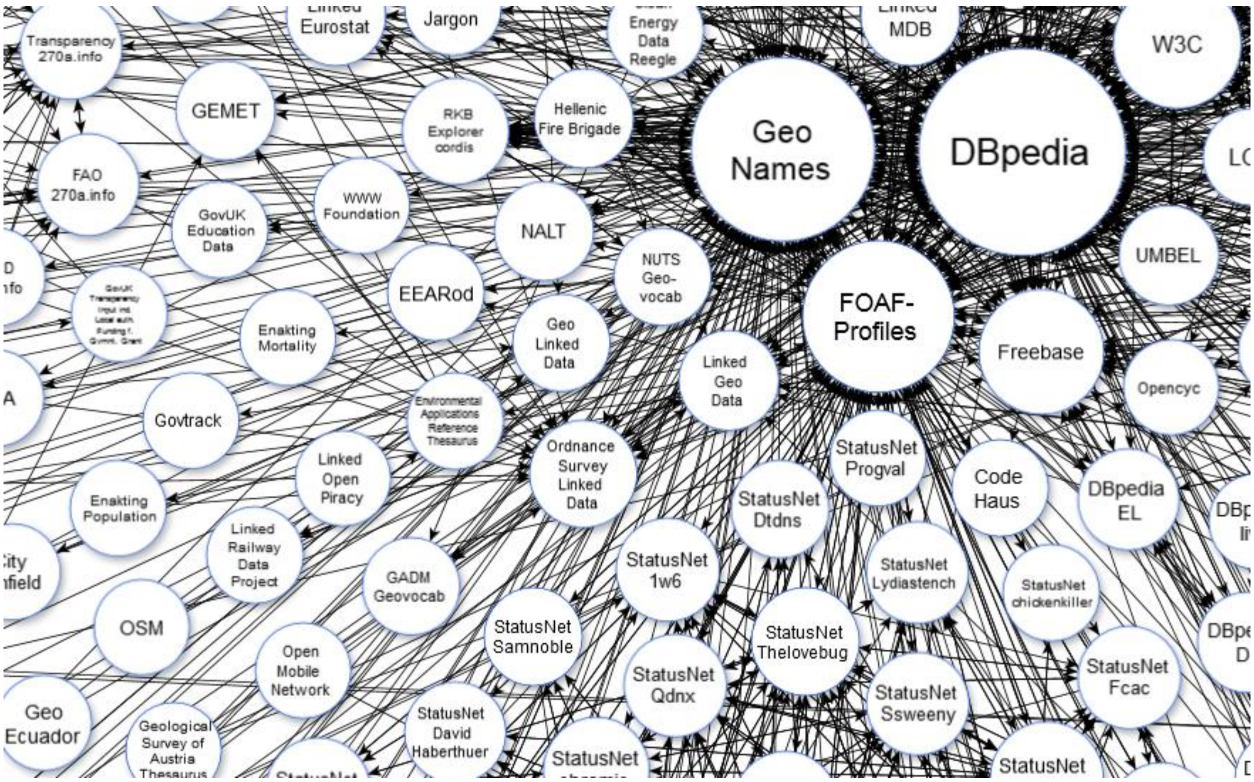


Figure 14: Part of the Linked Open Data (LOD) project (Cyganiak & Jentzsch 2014)

The Base Address Ontology (BAO) developed in this research (see Chapters 4 and 5) could be added as an address domain vocabulary to the LOD cloud, thereby becoming a small contribution to science and providing easy access and reference for address domain ontology developers.

Chapter 3

3 Method

3.1 Introduction

Address data interoperability across standards is important for sharing and exchanging of address data, software, and other related tools. Owing to the variation in the format of digital address data used and conflict and/or mismatch of terminologies used across standards, however, it has been challenging (if not impossible) to share or exchange address data.

Semantic ontologies have been used in other fields to facilitate interoperability. This dissertation investigates how semantic ontologies can be used to facilitate address data interoperability across standards. It has been important to do this research because addresses have special features that need to be handled in a unique way.

The research aims to evaluate an approach to facilitate address data interoperability and to be able to generate more information by combining the different address data sources. Semantic ontology was chosen as an approach to perform the task.

The next section of this chapter discusses the scientific research methods that are used to address the research problem. It is followed by a section that discusses the experiment design used in the research. The software and tools that are used in the research are critically analyzed and discussed in section 3.4. The data and address standards that are used to facilitate the experiment and to test the semantic approach are discussed in section 3.5. The last two sections of this chapter discuss the limitations and ethical considerations of the research.

3.2 Research Design

Research techniques such as an extended literature review, prototype/simulation, experiment, and arguments were used in this research. Some interdisciplinary research techniques were also used.

Extended literature reviews were done to obtain an overview of addresses, address standards, the semantic web, and ontologies amongst other things. The literature survey was used to understand the theoretical background and applications of addresses and the process and purpose of address standards. A literature survey was also done on the semantic web and ontologies to understand and analyze the theoretical background, standards used, similar or related research done in the past, and to learn about the tools and software needed for the development. The literature survey, however, gives only background knowledge and previous work done, and, since this research is unique and the area has not been investigated previously, more needs to be done to make new findings.

Achieving address data interoperability across standards across the globe requires considering many address standards and huge amounts of address data. This, however, is a time and resources consuming process and so a prototype which considers some carefully selected standards and address data is used to simulate the global address system. Global address data interoperability would require standardization and the involvement of all the countries. Prototyping allows one to consider all the major issues and implement them on a small scale, saving time and resources and showing that it can be achieved.

Experiments were done using sample address data and standards to test whether the prototype can be used to demonstrate address data interoperability across standards. Queries and inferences were, furthermore, used in the experiment to generate new knowledge. The experiments produced research results that need to be analyzed critically.

The results of the experiments were recorded, evaluated, and analyzed. The results are discussed in detail in chapter 5. Arguments were made about how the research can be applied for the further use of semantic ontologies for address data interoperability on a larger scale.

3.3 Experiment Design

The goal of this research was to explore ways to accomplish digital address data interoperability across domains using the semantics of the data regardless of its syntactic representation. The domains in this context refer to the different addressing systems used all over the world.

Ontology is the formal representation of knowledge in the form of concepts and their relationships within a specific domain. An ontology based semantic approach was, thus, applied to achieve address data interoperability across different addressing systems.

A four-tier ontology model was designed to integrate address data which have different forms and syntactic representations (see figure 15). Chapter four discusses the experimental approach of the four-tiered ontologies to address interoperability in depth.

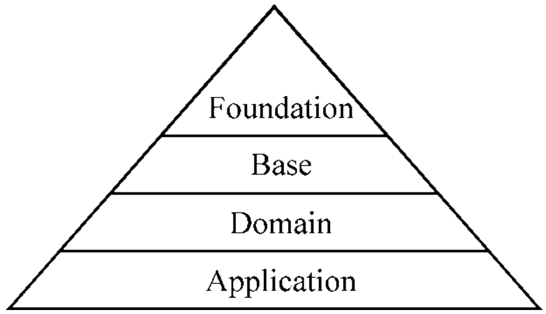


Figure 15: Four Tier Ontology Architecture

The address ontology that facilitates interoperability is designed to be used by many people for a variety of purposes. The ontology model has to be able to represent the concepts at different granularities. It also needs to be an extendible model for different users to add new concepts for their own specific purposes. For example, one might want to extend the ontology with South African address concepts; another one might extend the ontology with Chinese or Japanese addresses. For the ontology to be practically useful and flexible, it should have decentralized control allowing different users to add new concepts without any dependence on a central managing entity. No central entity can modify the ontology for every single possible purpose. The flexibility, extendibility, and decentralization of the ontology, however, causes new challenge because someone might not need (even might want to avoid) the extensions made by others. For example, for someone who is using South African addresses and UPU addresses, extensions to Chinese addresses are not useful. An ontology model was, thus, designed with a four-tier hierarchy of ontologies which is as extendible and decentralized as possible while still allowing people to use it at any level of granularity of concepts.

The four-tier hierarchy is created based on the source and purpose of the ontologies. The granularity of the concepts of the ontologies increases as they go higher in the hierarchy. The ontologies on the lower level of the hierarchy extend the network of concepts from the upper ontologies. At the top of the hierarchy is foundation ontology which is followed by base ontology, domain ontology, and finally application ontology. Each of the ontologies on the hierarchy extends the network of concepts from the upper one by adding new concepts (resources), their instances, and the relationship among them.

At the top of the hierarchy is the Foundation-Tier, which describes high level spatial (address domain-independent) concepts, ontology language building concepts, and their relationships. Foundation ontology is a "high level formal framework constraining the meaning of high level information modeling predicates" (Gangemi 2005; Zemmouchi-Ghomari & Ghomari 2009). The choice of foundation ontology affects the extendibility of the base ontology. In this research study, a foundation vocabulary for representing geographic information in the WGS84 geodetic reference datum was used. This foundation ontology defines geographic concepts such as *geospatial thing* and *geoPoint*, among others. The details of all the foundation ontologies are described in section 4.2 of the document.

The *base* ontology is more specific than the foundation ontology, and in this research it specifically deals with addressing systems, but it is not specific to a single addressing standard. The base ontology establishes the organizational backbone of the more context specific domain ontologies (Navigli & Velardi 2004). Developing a base ontology requires a good knowledge of the domains (that would use it). Domain experts and knowledge engineers must, thus, collaborate to prepare the base ontology (De Nicola et al. 2009). The domain expert's major task is to prepare a conceptual model by listing the high level concepts, their properties (attributes), and the relationship among the concepts. In other words, they have to prepare the high level vocabulary that is shared by all domain specific ontologies. In this context the domains are the different address standards which were developed for variety of purposes and have different conceptual models.

It is very important to develop a UML conceptual model before developing ontology, because ontologies are not convenient for human communications (Parreiras et al. 2007; Staab & Studer 2009). In Information Science the *conceptual model* is a tool used to describe the semantics of software applications at a high level of abstraction in terms of structure, behaviour, and user interaction (Andersson et al. 2006). Its main aim is to find a relationship among the different concepts used by domain experts and to describe these terms and concepts. Domain experts and knowledge engineers may have to be involved in the process of developing a conceptual model

(De Nicola et al. 2009). It is easier to identify conceptual mistakes using UML rather than from OWL ontology.

An ideal universal address system would be a single address standard that represents an address anywhere in the world and for all possible purposes. Such a universal address standard would create seamless address data, tools, and software sharing and exchange. Owing, however, to the cultural attachment of addresses to societies, their purposes, and local legal regulations, it is impossible to develop such an ideal universal address standard that would replace all the existing ones. The best option for the future is, thus, to develop a standard that would facilitate interoperability among the existing address standards. A conceptual model for addresses, with common concepts from various address standards, was developed to serve as the basis for the development of a system that will facilitate address interoperability. Such a model is useful for designing interoperable software solutions that share and manipulate addresses. The Base Address Ontology (BAO) was developed from this conceptual model. The concepts of the BAO are carefully selected not to be too specific or too generic.

The domain ontology has concepts of a finer granularity that describe the knowledge within a specific domain. Two domains of addressing are chosen to illustrate address data interoperability using ontologies. These domains are the INSPIRE (data specification) and UPU-S42. The conceptual models of the standards were converted into UML models which extend the base conceptual model (of the BAO) prior to the implementation of the ontologies. A domain ontology for each model was developed based on the respective conceptual model.

Application ontology is an ontology designed for a specific use, task, or purpose (Malone & Parkinson 2010), and it usually references domain ontologies to construct its ontological classes, their relationships, and properties. Based on their purposes, ontologies are classified as reference and application ontologies (Menzel 2003). The above mentioned (foundation, base, and domain) ontologies are reference ontologies, and application ontologies are developed from them.

To make the domain ontologies useful to as many applications as possible, instances (literal property values) are avoided from the domain ontologies. This leads to the creation of the fourth tier in the hierarchies (application ontology) which extends the domain ontologies with application to specific concepts and instances. For example, the INSPIRE domain ontology can be used by many European countries where each of them can extend it with their own requirements such as the language used for addressing, code lists, and other country specific restrictions including national address data instances.

The dependence of application ontologies on common domain ontologies increases the number of inferences that can be made from them. For example, address ontologies from multiple countries across Europe that share the same domain ontology (i.e. INSPIRE ontology) can be used to make inferences about regional address data. Application ontologies that use multiple domain ontologies but share a common base ontology can, furthermore, also be compared and useful inferences can be made to complement their class extensions.

Four application ontologies were designed to illustrate the use and advantages of address ontologies. The first application ontology (UKdata) references only the INSPIRE domain ontology, and its purpose is to illustrate the use of an application ontology and to serve as a comparison ontology for the second one. The second application ontology was UPUdata ontology that references the UPU-S42 domain. The third application ontology was UKdata+UPUdata and the last one was UKpostal ontology. All four application ontologies are discussed in depth in section 4.5.

Once the application ontologies were ready, the inference engine was run which made inferences and assigned class extensions to proper classes. Queries were, furthermore, used to generate new relationships automatically. These demonstrate that semantic ontologies can be used to share and exchange address data across contexts.

3.4 Software Used

Different tools and software were investigated for developing ontologies, running SPARQL queries, making inferences, and building conceptual UML models. TopBraid (proprietary software) was used for its powerful query engine, good inferences, user-friendly ontology development environment, and its graphical ontology representing tool which can generate UML-like representation for ontologies. Other ontology development tools, such as Sesame and Protégé, were tested, but TopBraid was chosen because of its diagrammatic and graphical modeling of ontologies, good resource editor, and comparatively powerful query and inference engines. Enterprise Architect (also proprietary software) was used to develop the UML conceptual models.

TopBraid is a development tool for semantic models (ontologies). It is a complete editor for OWL models and other RDF-based components and services. It is a tool that supports OWL statements with visualization and diagramming.

Since OWL is based on RDF, an OWL statement is a set made up of *resources, properties,* and *property values* which are IRI representations of concepts, their relationships, and instances. OWL ontology is made up of sets of OWL axioms (subject, predicate, and object triples). If RDF/XML (which is the W3C recommended) representation is used to present the ontology it then becomes challenging to navigate inconsistencies manually without the support of tools. TopBraid provides graphical representation tools that support consistency analysis of OWL ontologies.

The user interface of TopBraid editor enables the describing of resources with the correct IRIs and building relationships graphically. Manually validating ontology becomes difficult as the number of concepts and ontology statements grow. The use of the different Resource Editor panels helps to reduce coding errors and so improves consistency in the ontology. For example, the annotations, axioms, and properties of classes are modified using the Form Panel in the Resource Editor of TopBraid. The various inference engines that TopBraid provides are also used for consistency checking.

TopBraid generates UML-like diagrams that show classes (concepts), their properties, and the relationship among the classes. Each class is represented as a rectangle inside which all its properties are listed. The UML-like diagrams make it easier to visualize and edit ontologies. Figure 15 shows the concept ‘Address’, its properties, and relations to other concepts.

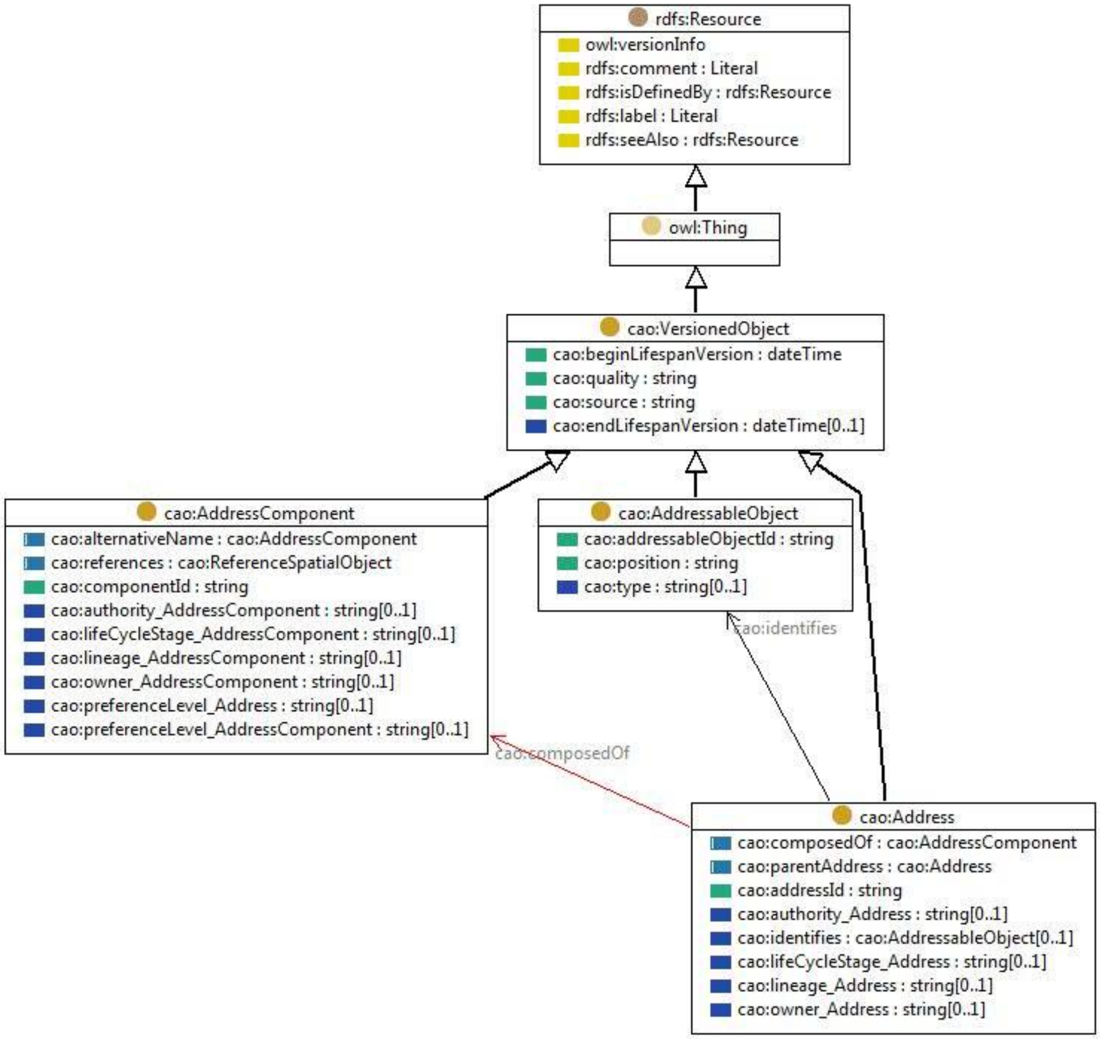


Figure 16: UML-like model for the concept 'Address'

Since OWL is RDF based ontology language, the best way to present OWL triples to humans is to use RDF-Graphs. TopBraid generates RDF-Graphs with the subject, predicate, and object triple for any selected resources. Figure 16 shows two OWL triples, with one of them having literal value for the object and the other having RDF URI references for both the subject and the object. Namespace prefixes are used to shorten the URIs and make them easier to present.

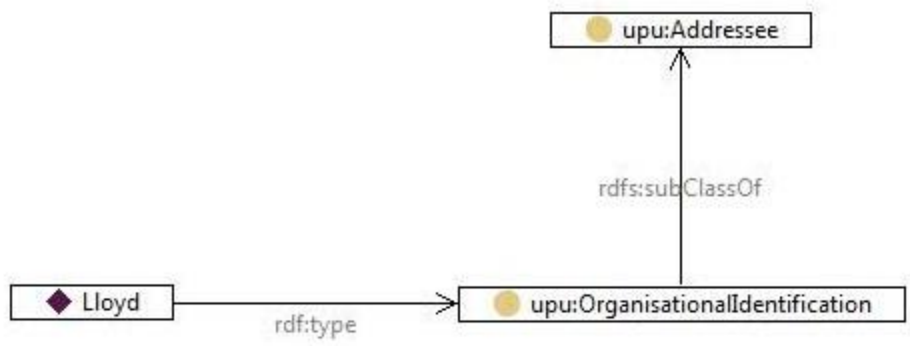


Figure 17: TopBraid generated RDF-Graph

TopBraid can also generate and convert (from and to) RDF/XML, turtle, N3, and N-Triple code.

3.5 Data and Standards Used

A number of standards (including SANS1883, INSPIRE data specification, UPU, and others) were considered during the research. To use ontologies in facilitating interoperability across standards, it is important to create a core ontology that takes all the other domains into consideration. This core ontology needs to be based on a model that is acceptable by all the representatives of the major domains. The meetings of ISO19160-1 *Conceptual Model* were a good environment to prepare such a model. Since, however, the preparation of ISO19160 takes long time and owing to time constraints, the first draft of the standard was used to prepare the core ontology which is *Base Address Ontology (BAO).*

INSPIRE and UPU-S42 domains were selected for the research because they are regional and international address standards respectively. The availability of sample address data was also an issue in the selection of the address standards. UK address data with INSPIRE data specification and UPU-S42 specification was used in the research.

3.6 Limitations

The lack of a comprehensive standardized conceptual model to facilitate address data interoperability was a challenge for the research. The ISO19160-1 standard is aimed at developing such a model. It is still work in progress, however, and the standardizing process might take years to be completed. One of the early drafts of the standard (ISO19160-1) was modified to be used in the research.

The lack of address data that is represented in two or more address standards was another challenge. To test address data interoperability there must be matching address data across domains. The option to solve this challenge was to use INSPIRE data specification and UPU S42 because there are good address samples and templates of different European countries. Access to the people who are responsible for the above-mentioned standards also made it easier to use the standards.

There is a lack of time to prepare a fully functional ontology that could facilitate address data interoperability. There is not time to wait for the ISO19160-1 which would become the standard conceptual model for address data interoperability. Time is also limited to allow for the consideration of many other standards as domains. Two standards were, thus, chosen carefully for the development of domain ontologies, and more standards were considered during the development of the core ontology (BAO). The four tier model is, thus, easily extendible to accommodate more domains in the future.

Another challenge, owing to underlying nature of OWL, was the difficulty of presenting the ontology code in the document. OWL is designed to be used by computer applications to process content of information and not for presenting information to humans. UML-like diagrams that are generated by TopBraid (ontology development software) are, thus, used to represent the ontologies in the document.

3.7 Ethical Considerations

The address data used for the experiment was selected in a way that it does not violate the privacy of any one, and the research was conducted in such a way that it does not violate or support the violation of privacy of people.

Samples that include private mailee and addressee data and queries which extract personal information were not included in the experiments.

Chapter 4

4 The Four Tier Ontological Hierarchy

4.1 Introduction

The evaluation of the semantic approach used for facilitating address data interoperability required the development of multiple ontologies organized in a four-tier hierarchy. The ontologies in the hierarchy have decreasing granularity of concepts from top to bottom (see figure 15). At the top of the hierarchy is the Foundation-Tier. Section 4.2 discusses the foundation ontologies that are required for the development of the address reference ontology. The address reference ontology that was developed as part of the research is discussed in section 4.3. The address reference ontology was used to develop two domain specific ontologies. The domain ontologies used in this research are discussed in section 4.4. Finally, four application ontologies, which are ontologies at the lowest tier of the ontological hierarchy, are discussed in section 4.5. The ontologies were processed with ontology reasoning, inferencing, and querying tools to facilitate interoperability.

4.2 Foundation Ontologies

This is the top layer of the four-tier ontology hierarchy which defines the concepts required to build ontology in general and address ontology in particular. The foundation ontology layer in this research includes concepts that describe OWL language specifications, concepts that describe metadata, data types, and geographic attributes (such as latitude and longitude). These concepts are imported from standardized (mostly W3C) sources. Table 4 shows the list of the vocabularies used in this layer and their Namespace URIs. The use of these standardized vocabularies improves the quality of the ontologies in the lower tiers of the hierarchy by providing well defined and internationally de-referenceable URIs.

Vocabulary For	**Prefix**	**Namespace URI**
Dublin Core (Metadata)	dc	http://purl.org/dc/elements/1.1/
Geographic Terms	geo	http://www.w3.org/2003/01/geo/wgs84_pos#
OWL language concepts	owl	http://www.w3.org/2002/07/owl#
RDF language concepts	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
RDFS language concepts	rdfs	http://www.w3.org/2000/01/rdf-schema#
Non-spatial (xml) data types	xsd	http://www.w3.org/2001/XMLSchema#

Table 4: Ontology vocabularies used at foundation tier

OWL ontologies are used to enable computers to understand the semantics of the data used, gain knowledge from it, and generate new knowledge based on existing knowledge. In order to create an environment where the machines totally understand the semantics of all the domain concepts, it is important to express all the concepts in the ontology (including the building concepts of the language) in a format that is semantically interpretable by machines. OWL, as web ontology language, is designed to store not only the syntax but also the semantics of data. OWL, however, is an extension of RDFS (Resource Description Framework Schema) which itself is an extension of RDF (Resource Description Framework). The vocabulary for the OWL, RDFS and RDF language concepts and rules with their respective sets of well-defined (de-referencable) URIs are, thus, necessary for the development of OWL ontologies. They form a crucial part of the foundation ontology.

In the OWL ontologies developed in the research, most of the non-spatial literal property values have primitive data types such as integer and string. For well-defined URI reference, therefore, the W3C XML vocabulary was used. This also constitutes part of the foundation ontology.

An address might have both spatial and non-spatial components. The spatial part of an address needs to be represented using spatial data types such as point, line, or polygon, and the non-spatial part needs to be represented using other data types (usually primitive data types such as integer and CharacterString). The conceptual address model used in this research has an (optional) spatial component for an Addressable Object.

The position of an 'AddressableObject' (in the conceptual address model) is a spatial attribute of `GM_OBJECT` type. The spatial data types are described in the WGS84 ontology. In the foundation ontology `geo:SpatialThing` type resource describes geometric objects with certain size, shape, or position. The resource `geo:SpatialThing` describes a spatial extent using three properties which are latitude (`geo:lat`), longitude (`geo:long`), and altitude (`geo:alt`) (see figure 18). The `geo:Point` resource represents the unique identifier of a place using a coordinate system relative to Earth, such as WGS84. `geo:Point`is a sub-class (`rdfs:subClassOf`) of `geo:SpatialThing` and it uses lat/long/alt to identify a spatial point uniquely.

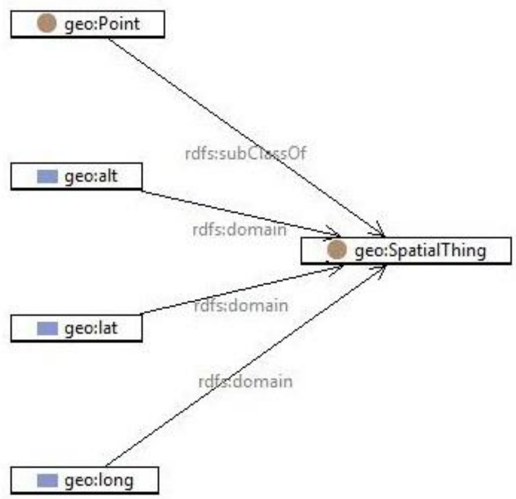


Figure 18: Foundation Ontology (Graphical Representation)

The WGS84 (sometimes called Basic Geo) vocabulary used for the foundation ontology is used by the likes of Yahoo! Maps, geocoder.us, and Locative packets (Brickley 2004) within RDF documents or RSS (as namespaces).

Other ontologies, such as foaf (Friend-Of-A-Friend) and geonames, were considered but not included in the foundation ontological layer. FOAF is an ontology that describes people, the links between them, and the things they create and do. For ethical reasons, however, all the data that describes people are avoided in this research. If there is a need to use personal information then one can add the FOAF ontology at the domain ontological layer. Similarly the geonames ontology, which describes geographical features, is avoided because it is not used for the domain considered and in order to keep the ontologies as simple and clean as possible. Once again, however, if the user wants to use geonames vocabularies then it can be imported at the domain layer.

4.3 Base Address Ontology

In the four-tier ontological hierarchy, next to the foundation layer is the Base layer (see figure 15). This layer extends the foundation layer with concepts that are carefully selected to facilitate interoperability among the different domains (which are defined in the third ontological layer). An ontology called Base Address Ontology (BAO) was developed at this layer to facilitate interoperability at lower tiers of the hierarchy.

The base address ontology was developed as reference ontology for other domain (standard) specific ontologies. As illustrated in figure 19, the process of developing a base ontology requires close consideration of the domains that derive (extend) their ontology from the base ontology.

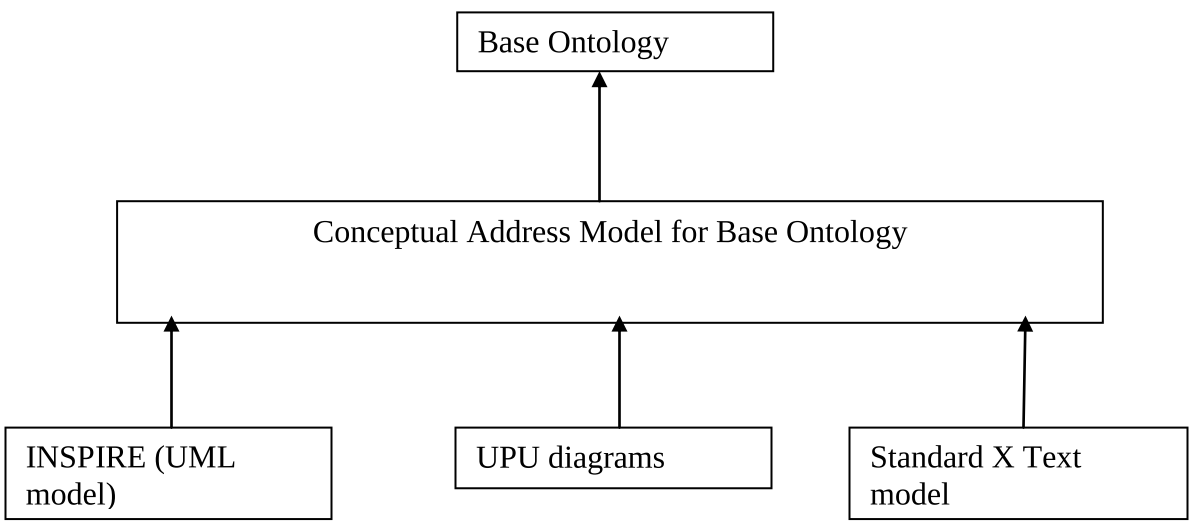


Figure 19: Base Address Ontology development stages

A conceptual model for the BAO needs to be prepared by selecting concepts that are common to most (if not all) of the address domains that can used. The concepts, thus, have to be selected carefully by considering as many and as diverse address domains as possible. The involvement of domain experts is essential in the process of developing the conceptual model.

Address standards use different notations to represent their conceptual models (see figure 19). For example, INSPIRE data specification uses UML diagrams, UPU-S42 uses tabular diagrams, and SANS1883-3 uses a combination of UML, tables, EBNF, and textual descriptions. The different conceptual modeling formats have their own advantages and limitations. Sometimes the limitations of the modeling system make it difficult to understand the relationship among different address concepts in a standard because they are prone to ambiguity. The UPU-S42 and its tabular-diagram modeling of concepts is good example of such ambiguous model. Hence, sometimes it was necessary to convert the conceptual model for the domains into a UML conceptual model to get a better understanding of the models and to avoid ambiguity of concepts. The base address conceptual model is developed in UML because it is because concepts in UML class diagram, if designed properly, are less prone to semantic ambiguity.

Different address domains were considered during the process of developing the base address conceptual model. SANS-1883, INSPIRE data specification, UPU S-42 and OASIS were chosen to be used for the preliminary development of the conceptual model, because of their diversity in purpose and geographical coverage. SANS-1883 is a South African address standard that can represent a national addressing model (with various purposes) of an African country. INSPIRE data specification is a European (regional) address standard that is designed to facilitate interoperability among European countries. UPU S-42 and OASIS are international address standards used for industry specific purposes. This list of address domains was used as a starting point to analyze what address standards have in common.

The above-mentioned address domains are, however, not representative of all the address domains in the world, especially the Asian, American and those in Oceania. There was, thus, a need to consider more domains and involve more domain experts.

The ISO/TC211 meeting was the best opportunity to work with international domain experts from all over the world. They were, furthermore, working on an international standard (ISO19160) to facilitate interoperability of addresses data. Domain experts from South Africa, the European Union, United States, Canada, Japan, South Korea, Australia, New-Zealand, and Saudi Arabia, amongst others, participated, and the domains from their respective countries were considered for the development of the ISO19160-1 conceptual model.

The early draft of the ISO 19160-1 conceptual model (see figure 5) for address information, which describes common address concepts, their properties, and relationships in different address standards, was used as the next step in the development of the base address conceptual model. Ideally, it would be very good to use the ISO19160-1 conceptual model as the conceptual model for the developing BAO. It is, however, a standard-in-progress which goes through many changes during the process, and it is expected to take some years before it is completed. A separate and more stable base address conceptual model which significantly takes the ISO19160-1 model into consideration had, thus, to be developed (see figure 20).

The aim of the base ontology is to act as a bridge to facilitate interoperability among the different address domains, and so the concepts for the base address conceptual model were selected carefully not to be too specific or too generic. If the concepts are too specific the base ontology will not be able to accommodate all the domains; on the other hand, if they are too generic then they significantly reduce the interoperability and the amount of inferences that can be done. For example, if concepts like “Addressee” and “Individual/Person”, which are too specific (used only on some domains such as UPU-S42), are included, most of the other domains which do not even consider these concepts will not be able to use it as a base ontology.

The UML conceptual model is used to identify, select, and visualize the concepts that are needed to develop the ontology. The conceptual model has six classes and two codelists. The classes that are included in the base address ontology are concepts that exist in most (if not all) address domains.

There are few universal address concepts, such as “Address”, “Addressable Object”, and “Address Component” that exist in all domains. These concepts appear with different terminologies and properties in different address domains. For example, the concept “Address” exists both in UPU-S42 and INSPIRE data specification; it is, however, referred to as “Address” in INSPIRE and “Delivery Point Specification” in UPU-S42. These two concepts have differences in their properties but they also share common properties that define an address. The common properties of the selected concepts are used to define the concepts at the second tier of the hierarchy (base layer), and the definition of these concepts is extended at the third tier (domain layer) to complete the meaning as defined in the specific domain.

The concepts at the base layer are defined with minimal mandatory attributes to widen the scope of the domains they cover. The concepts, however, have many optional attributes which increases the sharing of concepts and interoperability among the different domains. For example, an addressable object has only one mandatory attribute which is the identifier of the addressable object. The mandatory attributes of the base layer concepts must exist in all the domains and should be represented in all domain conceptual models.

Seven ontological classes (rdfs:Class) were created to represent the six classes and one code list (lifecycle stage type) on the base address conceptual model. The classes are Address, AddressableObject, AddressComponent, VersionedObject, AddressSpecification, ReferenceSpacialObjectandLifeCycleStageType. The classes Address, AddressableObject and AddressComponent are subclasses of VersionedObject class; hence they all inherit its properties. The six classes do not have instances (extensions) in this ontology because they are abstract high level concepts of a base ontology, but the class which represents the code list has three instances which are address:Active, address:Planned and address:Retired that represent the three possible values for a lifecycle stage. These extensions are used as restrictions to the values of the lifecycle stage properties.

The attributes and the relationships of the concepts from the model (fig5) become properties of the ontology, and, as a result, 29 datatype properties and 9 object properties were created as illustrated in figure 20. Datatype properties are properties that link individuals (resources) to data values (literals) while object properties link individuals to individuals (Bechhofer *et al*, 2004).

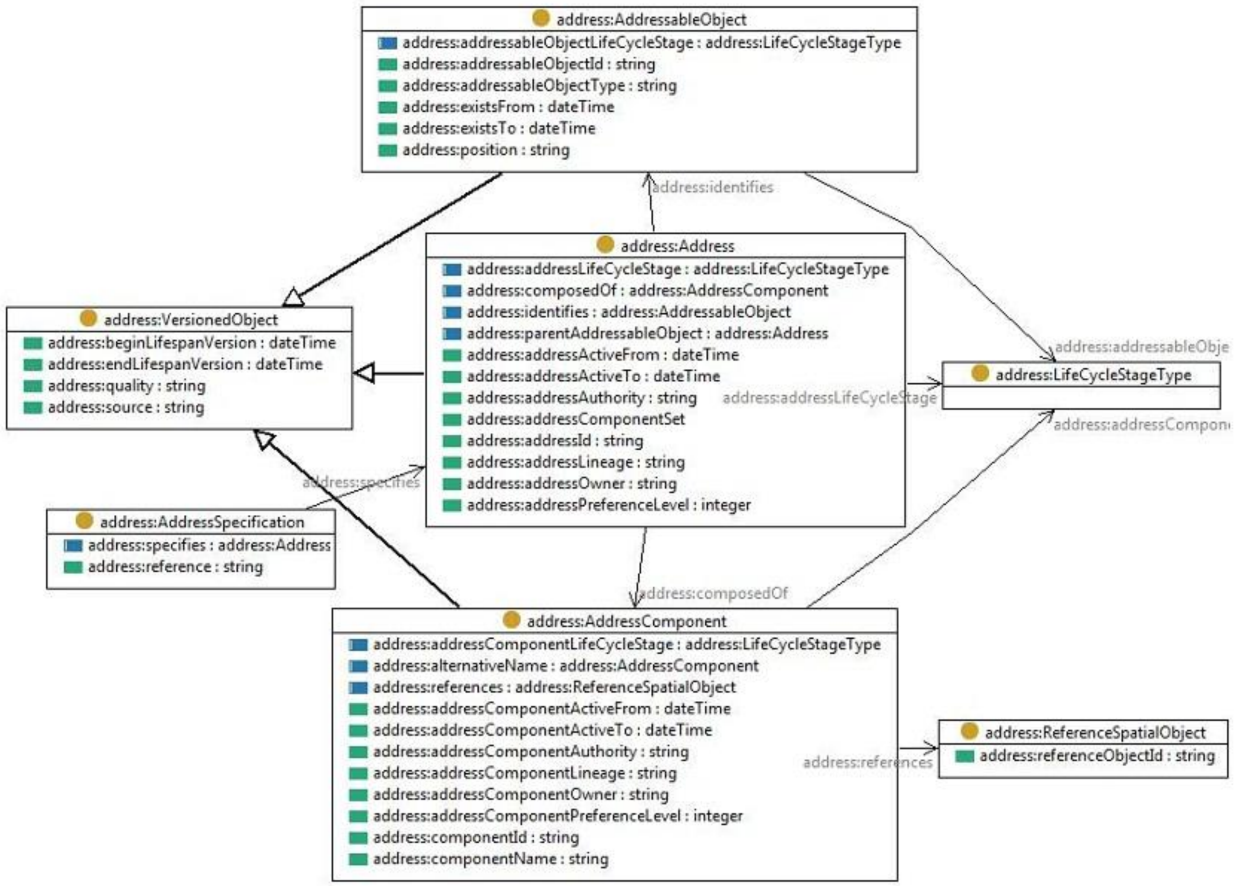


Figure 20 : UML like representation of the classes, properties and relationships and their constraints on the BAO

The domain and range of all the properties and relations are defined to restrict them from being misused (see figure 21). This is very important because, unlike object oriented programming in ontologies, properties do not necessarily belong to classes, and a property can link resources of any type unless they are restricted (Bechhofer et al. 2003).

```
address:addressableObjectLifeCycleStage
      a       owl:AsymmetricProperty ;
      rdfs:domain address:AddressableObject ;
      rdfs:range address:LifeCycleStageType ;
      owl:allValuesFrom address:LifeCycleStageType ;
      owl:maxCardinality "1"^^xsd:nonNegativeInteger ;
      owl:minCardinality "0"^^xsd:nonNegativeInteger .
```

Figure 21: Example of Restrictions Used (Turtle representation of OWL restrictions to the addressableObjectLifeCycleStage property)

Other important property restrictions, such as value constraints and cardinality constraints, were also used. For example, value constraint `owl:someValuesFrom` is used to restrict the values of the lifecycleStage properties to the values listed in the code list. Cardinality constraints, such as `owl:maxCardinality` and `owl:minCardinality`, are also used to restrict the cardinality of the properties in the ontology.

The prefix `address` is added to all the namespaces of the resources defined at the base tier of the hierarchy. The `address` prefix is for the local namespace URI "http://example.org/cao#". As stated above, all the resources in OWL are described by dereferencable URIs. The prefixes defined in table 4 are also used in the BAO. The URIs for the prefixes defined in table 4 are internationally dereferencable over the internet.

The ontology class `address:VersionedObject`is an abstract class with four datatype properties which describe the lifespan, source, and quality of an address data. These properties are also shared by three other classes (`address:Address`, `address:AddressableObject` and `address:AddressComponent`) which are sub-classes of the class. The lifespan data has XML dateTime type, and the other two properties of the class (quality and source of the versioned object) have XML string datatype.

The ontology class `address:Address` (referred as `Address` from here on) is derived from the `address` concept in the base address conceptual model which is a concept that exists in all address domains. The attributes and relationships of the `address` concept are the properties of the `Address` class in the BAO. The class `Address` has four object properties and eight datatype properties (see figure 22). The `Address` class is a sub-class of `address:VersionedObject` class and so all the four properties of `address:VersionedObject` are also properties of `Address` class. `address:addressLifeCycleStage`, `address:composedOf`, `address:identifies` and `address:parentAddressableObject` are the object properties the class `Address`. These four properties have the same domain, `Address`, and, thus, its object properties. They have different domains, however, which are `address:LifeCycleStageType`, `address:AddressComponent`, `address:AddressableObject` and `address:Address` respectively. These four object properties relate the class Address to the other objects in the BAO. The Address class has eight datatype properties which are address:addressActiveFrom, address:addressActiveTo, address:addressAuthority, address:addressComponentSet, address:addressId, address:addressLineage, address:addressOwner and address:addressPreferenceLevel. All these datatype properties, which have the domain Address, relate it to literal values with different XML datatypes.

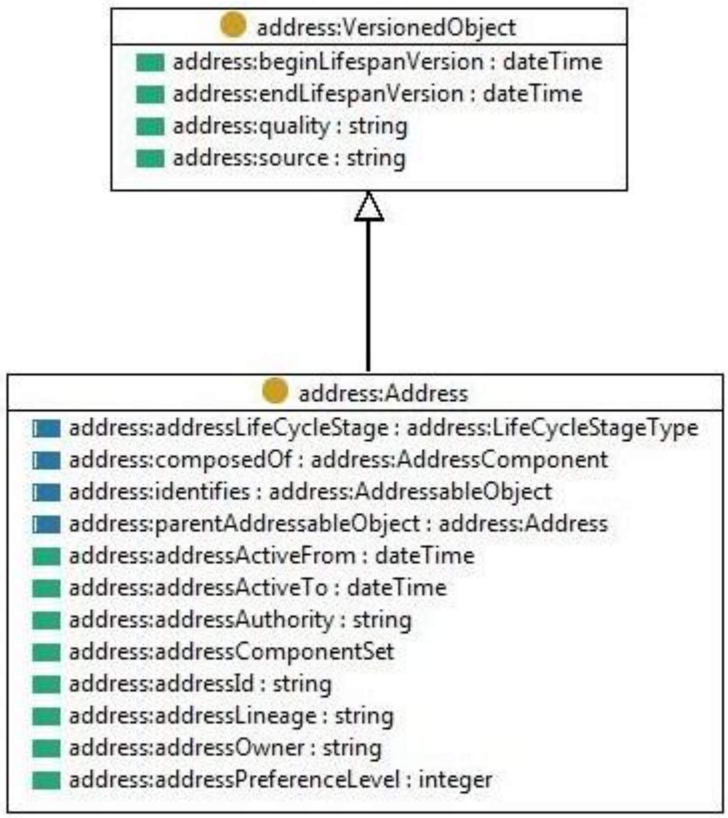


Figure 22: UML-like representation of Address class in BAO

Another sub-class of the versioned object class is address:AddressableObject class which is designed to represent the concept of different addressable objects. This class has four datatype properties and two object properties (see figure 23). An addressable object has a mandatory identifier property address:addressableObjectId, which is a datatype property with data type string and the cardinality of one. Since there are different types of addressable objects, the type of the addressable object should be specified using the optional datatype property address:addressableObjectType, which has string range. The duration of existence of an addressable object can be expressed using the optional datatype properties address:existsFrom and address:existsTo, which are both of type dateTime. The position of an addressable object can be expressed using the address:position object property with the maximum cardinality of one as geographic point type. The lifecycle stage of an addressable object can be any of the lifecycle types defined in classaddress:LifeCycleStageType, which are active, planned, or retired.

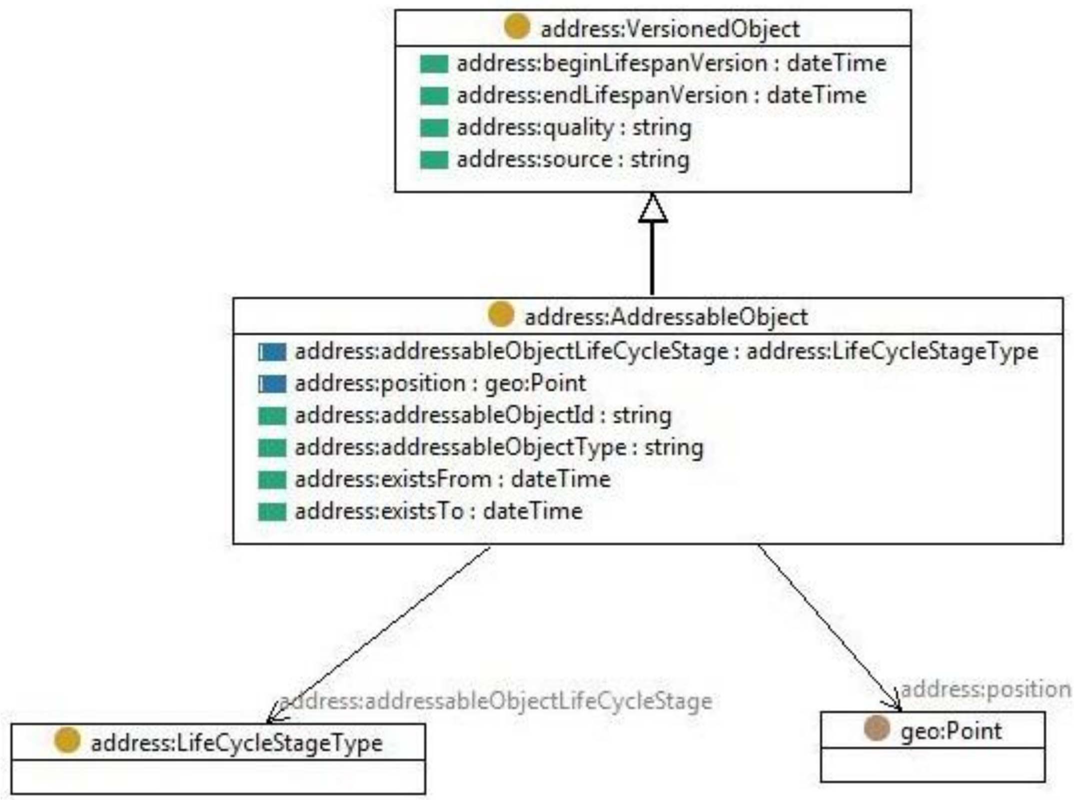


Figure 23: UML-like representation of addressable object in BAO

An address is usually composed of different components. For example, a building address in South Africa (according to SANS1883-3) is composed of a complete building unit identifier, (street identifier or intersection identifier) and locality. Another example is that postal delivery address components in the United Kingdom could be street number, district/sector identifier, town name, and postcode (UPU/POC Addressing Group 2006b). This shows that addresses in different places with different purposes are composed of different components. An address component's identification data can change over time which gives the component different versions over period of time. The `address:AddressComponent` class is a subclass of `address:VersionedObject` and the lifespan datatype properties are inherited together with the other versioned object properties (see figure 24). The validity date and time of an address component's version data is, however, determined by the datatype properties `address:addressComponentActiveFrom` and `address:addressComponentActiveTo` of the address component. The lifecycle status of a component changes based on its validity data. The value of a lifecycle stage can be one of the extensions of the `address:LifeCycleStageType` class, which are `address:Active`, `address:Planned` and `address:Retired`, based on the current date and the validity data.

An address component identifies one or more addresses and can have multiple names. It can, however, have only one unique component identifier `address:componentID` which is usually restricted at domain level with a domain specific format or pattern; hence, at the base tier, it is defined as a string type to allow the domains to put restrictions on the format.

All the ontological classes in the base tier are selected after careful observation of different addressing domains. These classes do not necessarily exit in all address domains, but they are the most basic components of most address domains. In fact only the `address:Address` class is the only mandatory class. In the following sections, different domain ontologies that extend the BAO demonstrate the usefulness of the ontology to facilitate interoperability across the domains.

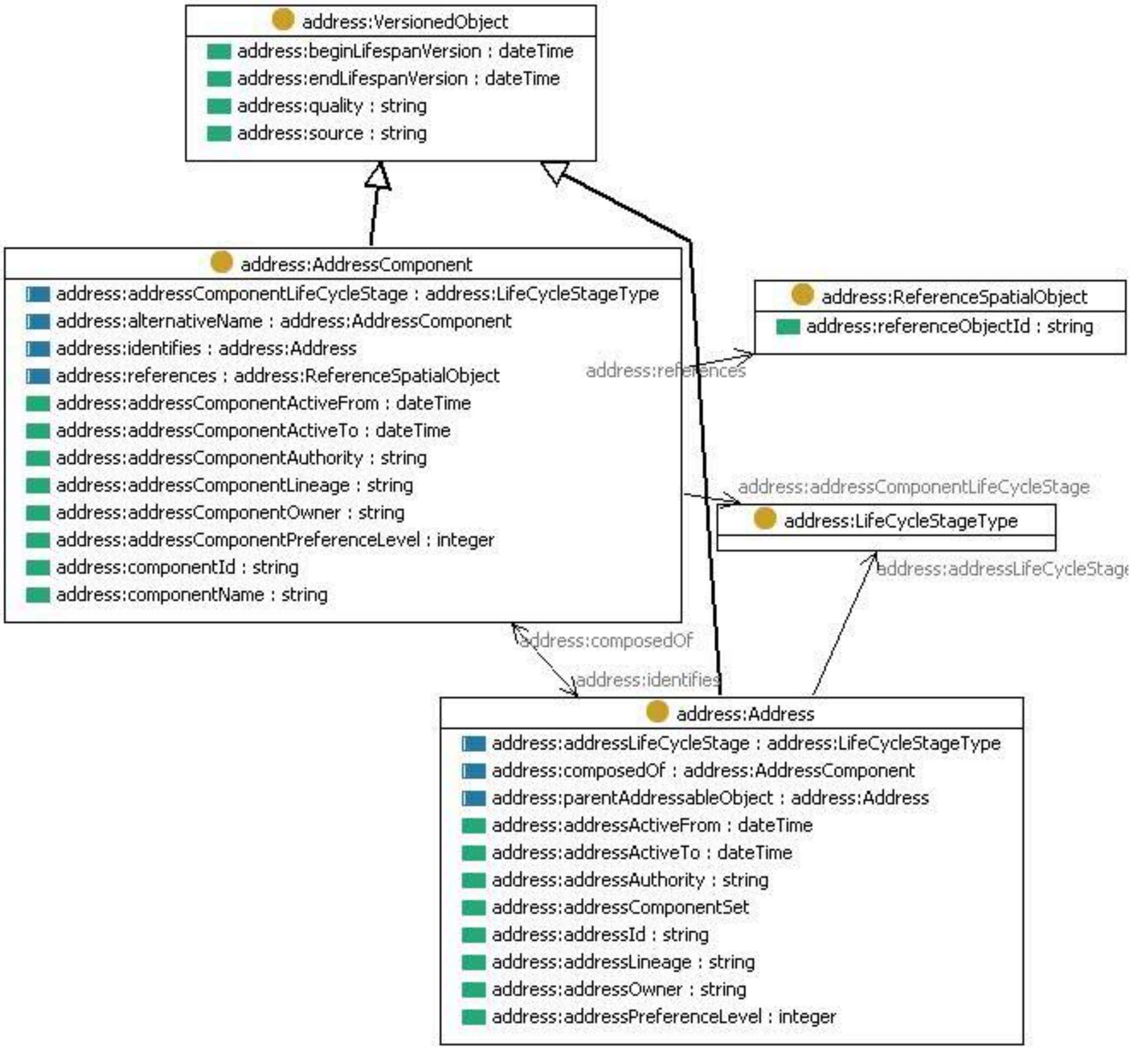


Figure 24: UML-like representation of Address Component in BAO

4.4 Domain Ontologies

The upper two tiers of the ontological hierarchy (foundation and base tiers) are a collection of concepts (and relationships) that are generic to all address domains. To use the ontologies to facilitate interoperability across different addressing domains, however, the domain specific concepts also need to be considered.

The third (domain) tier of the ontological hierarchy is where all the domain specific concepts are defined, and the concepts from the second (base) tier are extended with new concepts from domain specific vocabularies. The domains at this tier are the different addressing standards or systems that are used by different countries or organizations.

The concepts at this tier have a lower level of granularity than the ones at the base tier hence the domain ontology classes are more specific than the classes at the BAO. The BAO (discussed above) is the base ontology that describes the generic address concepts. These concepts need to be more specific in order to be used in different addressing domains (contexts). The BAO is easily extendible, so domain ontologies extend the definitions of the concepts by adding more properties, restrictions, and even new concepts as sub-concepts of the base concepts.

Developing domain ontology requires the participation of both domain experts and knowledge engineers. The domain experts of SANS-1883, INSPIRE and UPU-S42, therefore, which are addressing standards for South Africa, European local administrative systems, and international postal services, respectively, were involved.

Two domains of addressing (INSPIRE and UPU-S42) where chosen to illustrate the process of address data interoperability across contexts using ontologies. The next two sub-sections will discuss their ontologies in detail.

The ontologies at this tier extend the base ontology with domain specific concepts and add new concepts to them, but they do not add any instance extensions. The instance extensions are application specific, and they are handled at the fourth tier of the ontological hierarchy. The sole purpose of this tier is to create domain specific ontologies.

4.4.1 INSPIRE Address Ontology

The infrastructure for spatial information in the European Community (INSPIRE) is established by the directive of the European Parliament and Council to solve the challenges of sharing spatial information across various levels of public authorities in Europe (INSPIRE/TWG/ Addresses 2009). The directive has a data specification which clearly specifies address concepts in natural language and in a UML conceptual model as illustrated in figure 25. This directive aims to increase address data availability and sharing across member states by standardizing the addressing and address exchange systems.

The domain ontology that is specific to the INSPIRE addressing specifications has been developed to illustrate the role of domain ontologies in address data sharing and exchange. This domain ontology uses the BAO as the base ontology and extends it with INSPIRE specific concepts.

The vocabulary and UML conceptual models for INSPIRE were extracted from the INSPIRE specification (INSPIRE/TWG/ Addresses 2009). The structure of the conceptual model in the specification (see figure 25) shares similarities with the model of the base address conceptual model (see figure 5) but a separate profile for INSPIRE needed to be developed because of the differences. INSPIRE (UML) profile of the base address conceptual model was developed and an ontology was developed from it.

The prefix `"inspire:"` is a namespace for INSPIRE namespace URI (http://example.org/inspire#) which is attached to all the classes and properties that are INSPIRE specific and are not imported from the base address ontology. The prefix `"address:"` is the namespace for BAO namespace URI which is used to identify the base address classes and properties that are imported to the INSPIRE ontology. The resources with `"address:"` prefix does not imply that they have the same meaning (semantics) as in the base address ontology (BAO), because they could be (which usually is the case) extended to accommodate INSPIRE definitions, and the semantics could be different (extended). For example, the `"address:Address"` class (concept) that is inherited from the base ontology is extended to include properties that are specific to INSPIRE such as the date of registration of an address data in database. The attributes which already exist in the base conceptual model do not need to be redefined unless they have different or extended definitions. In that case, then either a metadata (annotation) is added to the attribute or a new attribute is created with a relationship constraint to link it to the attribute from the base ontology. New concepts, such as "INSPIRE Identifier" and "INSPIRE geographic position", are also added to the base address conceptual model components.

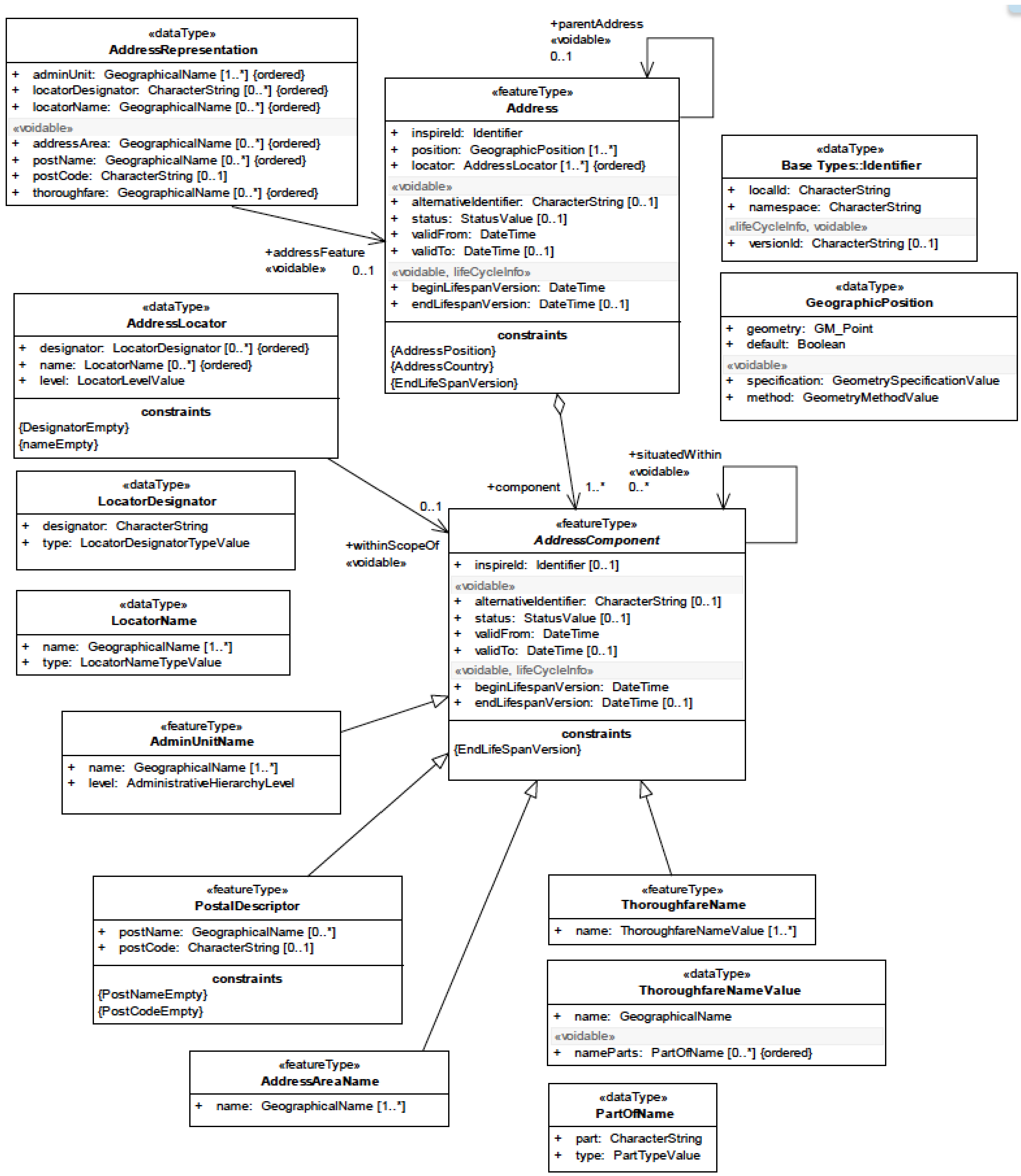


Figure 25: INSPIRE address application schema (INSPIRE Thematic Working Group-Addresses, 2009)

The address concept in INSPIRE data specification has a different definition from the one used in defining in the base tier. In fact it extends the base tier address concept with new attributes and puts in some INSPIRE specific specifications such as an address identifier. Address is a feature type in the INSPIRE data specification. Every address has a unique identifier which is of a special INPIRE address identifier data type. The "address:Address" class from the base tier has an identifier data type property with string data type but it is not good enough to represent an INSPIRE address identifier which is a combination of three different strings (figure 26). A new INSPIRE address identifier object property ("inspire:inspireAddressId") is, therefore, added to the class "address:Address". The INSPIRE address identifier has a unique data type which is adopted as a concept ("inspire:Identifier") at the domain tier. The address identifier data type in INSPIRE data specification uniquely identifies every address with its "localId", "namespace" and "versionId" attributes (of CharacterString type).

In the INPIRE ontology "inspire:inspireAddressId" is an object property with "address:Address" domain and "inspire:Identifier" range. It is equivalent to the data type property "address:addressId" which means that they share property extensions (i.e. have same instance values). Equivalent properties (related with owl:equivalentProperty) share extensions, but that does not mean that they are equal or the same. Hence, every instance of "inspire:inspireAddressId" is also an instance of "address:addressId" but the properties themselves are not equal (see figure 27).

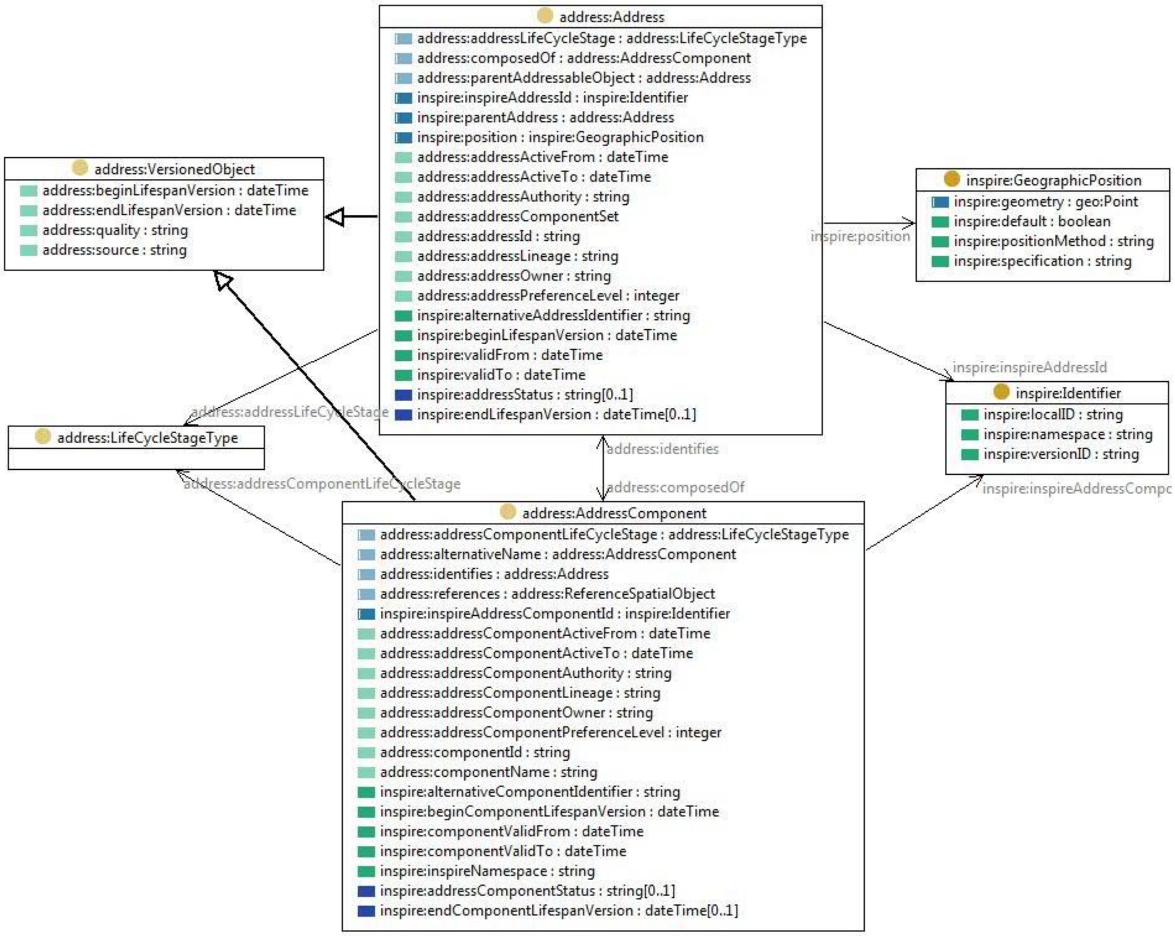


Figure 26: INSPIRE Address concept and its related concepts

```
inspire:inspireAddressId
      a        owl:ObjectProperty ;
      rdfs:domain address:Address ;
      rdfs:range inspire:Identifier ;
      owl:equivalentProperty
               address:addressId .
```

Figure 27: OWL code in turtle format for `inspire:inspireAddressId` *property*

An address in the domain of INSPIRE data specification must have a position attribute with "GeographicPosition" data type (figure 25) that has both spatial and non-spatial components. The object property "`inspire:position`" represents the position attribute in the INSPIRE ontology (figure 28). The domain for the property `inspire:position` is "`inspire:GeographicPosition`" which has one spatial and three non-spatial properties.

The "geometry" attribute of the class "GeographicPosition" (in the INSPIRE conceptual model) is the spatial part with "GM_Point" data type. The other three ("default", "specification" and "method") attributes are non-spatial and have Boolean and text data types. The object property "`inspire:geometry`" which has range of "`geo:Point`" is the spatial property of the position in the INSPIRE ontology. `geo:Point`is a concept adopted from the foundation tier of the ontological hierarchy which is typically described using a coordinate system relative to Earth, such as WGS84.`geo:Point.It` is a sub-class of "`geo:SpatialThing`" (figure 29).

```
inspire:position
      a       owl:ObjectProperty ;
      rdfs:comment "Geometry of the geographical position"^^xsd:string ;
      rdfs:domain address:Address ;
      rdfs:label "Position Geometry"^^xsd:string ;
      rdfs:range inspire:GeographicPosition .
```

Figure 28: Turtle code for `inspire:position` property

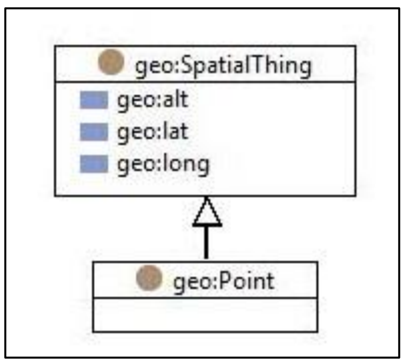


Figure 29: UML-like representation of `geo:Point` class

An address in INSPIRE data specification can have a parent address (`inspire:parentAddress`) which is of the same type (`address:Address`). It may also have an alternative identifier (`inspire:alternativeAddressIdentifier`) that is a sequence of character strings.

INSPIRE addresses have temporal properties which are similar to the ones in the base address conceptual model. The `inspire:validFrom` and `inspirevalidTo`properties are, thus, equivalent (`owl:equivalentProperty`) to the `address:addressActiveFrom` and `address:addressActiveTo` properties respectively. These properties describe the validity period of an address in the real world.

INSPIRE addresses have two temporal properties that do not exist in the base address conceptual model. These are `inspire:beginLifespanVersion` and `inspire:endLifespanVersion` which describe the date and time at which a version of the address was inserted or changed in the spatial data set (INSPIRE/TWG/ Addresses 2009).

An INSPIRE address is a structured composition of geographic names and identifiers. Just like an INSPIRE address, its components also have temporal, identifier, and status attributes. Even though INSPIRE address and address component concepts have similar attributes (for example `validFrom`, `validTo` and other temporal attributes) at the conceptual model (figure 25) they do not share a property (attribute) in the INSPIRE ontology. Sharing a property would mean that the property has more than one domain (`rdfs:domain`) which leads to the inference that the two domain concepts are equal (`owl:sameAs`) unless the `owl:unionOf` class description is specifically used. For example, if the datatype property `inspire:validFrom` is used to represent the `validFrom` attribute in both the `Address` and `AddressComponent` class (see figure 25), by only setting its domain (`rdfs:domain`) to both the classes then a second stage inference (inference on inference data) makes the two classes equal. This is because multiple `rdfs:domain` axioms are interpreted as conjunctions that restrict the domain of the property to the intersection of class extensions of the indicated class descriptions (Bechhofer et al. 2003).

To get better inference results, separate properties are used even for similar attributes in both the Address and AddressComponent concepts. For example, for the `validFrom` attribute, `inspire:validFrom` and `inspire:componentValidFrom`datatype properties are used to represent the attribute in both the Address and AddressComponent classes respectively.

An "Address Component" is an abstract concept that cannot have instances. The concept, however, has four sub-concepts that can have instance values. The sub-concepts of "Address Component" are "Postal Descriptor", "Address Area Name", "Thoroughfare Name" and "Administrative Unit Name". The INSPIRE ontology, thus, has four sub-classes of `address:AddressComponent` class which are `inspire:AddressAreaName`, `inspire:AdminUnitName`, `inspie:PostalDescriptor` and `inspire:ThroughfareName` (figure 30).

Some the optional concepts from the base tier of the hierarchy are not applicable to all domains. For example, the "Addressable Object" concept is not applicable to the INSPIRE domain. Since the domain tier ontologies extend the BAO, all the classes (concepts) from BAO are adopted by INPIRE ontology. No class extensions (instance values) are, however, added to the "`address:AddressableObject`" class.

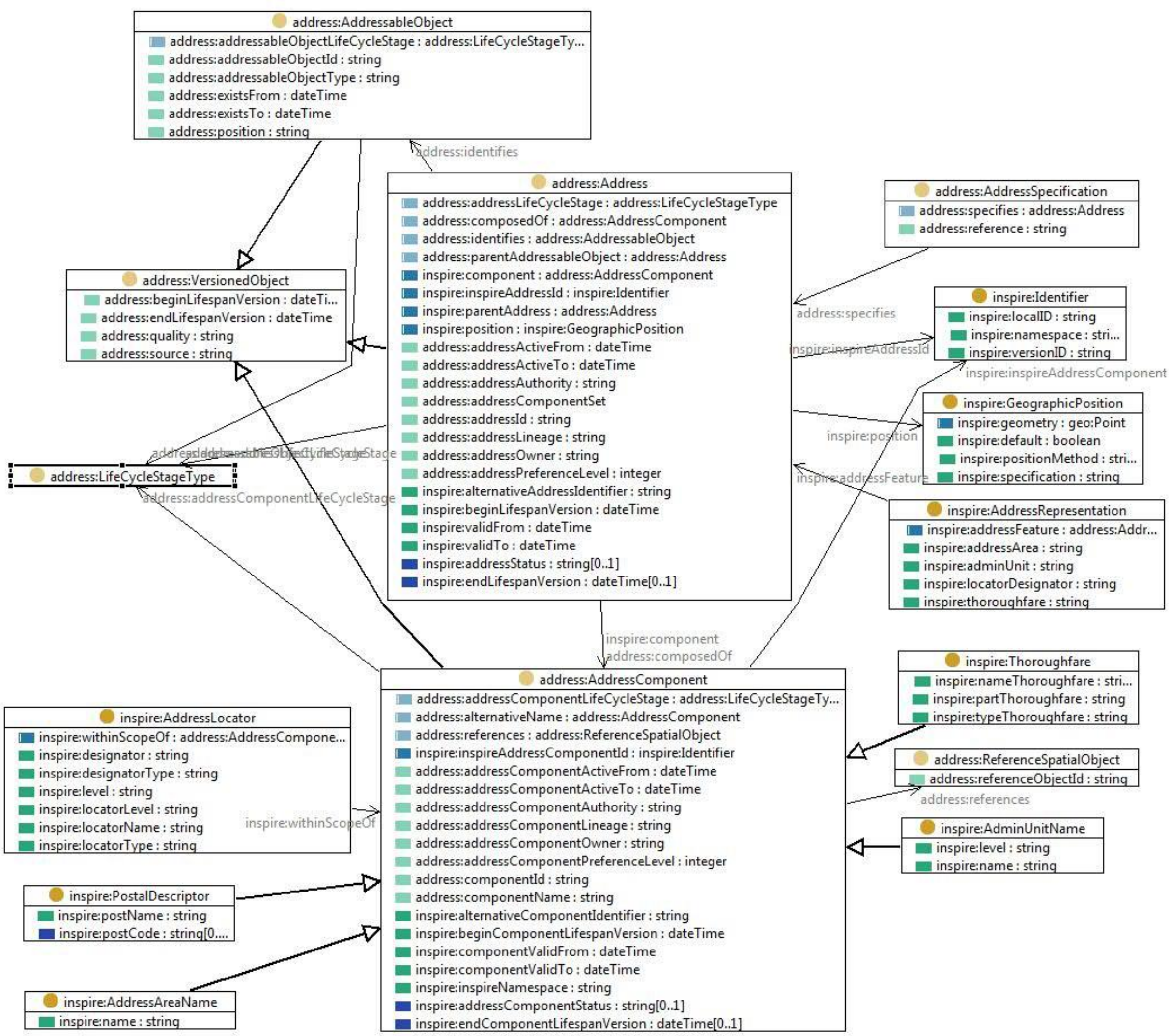


Figure 30: Partial INSPIRE conceptual model

4.4.2 UPU-S42 Ontology

Standard 42 of the Universal Postal Union (UPU-S42) is an international addressing standard that specifies postal address components and provides templates for international postal addresses (UPU/POC Addressing Group 2006b). It provides the description of postal address components with samples and instructions on how to use them.

The UPU-S42 standard specifies three hierarchical levels of postal address components, which are segment, construct, and elements, sequentially from top to bottom on the hierarchy. According to the standard, a postal address specification can have three optional and one mandatory segment. The mandatory segment is “Delivery Point Specification”, while “Addressee Specification”, “Mailee Specification”, and “Mail Recipient Despatching Information” are optional segments. As illustrated in figure 31, there are seven constructs and 38 different elements spread across the four segments.

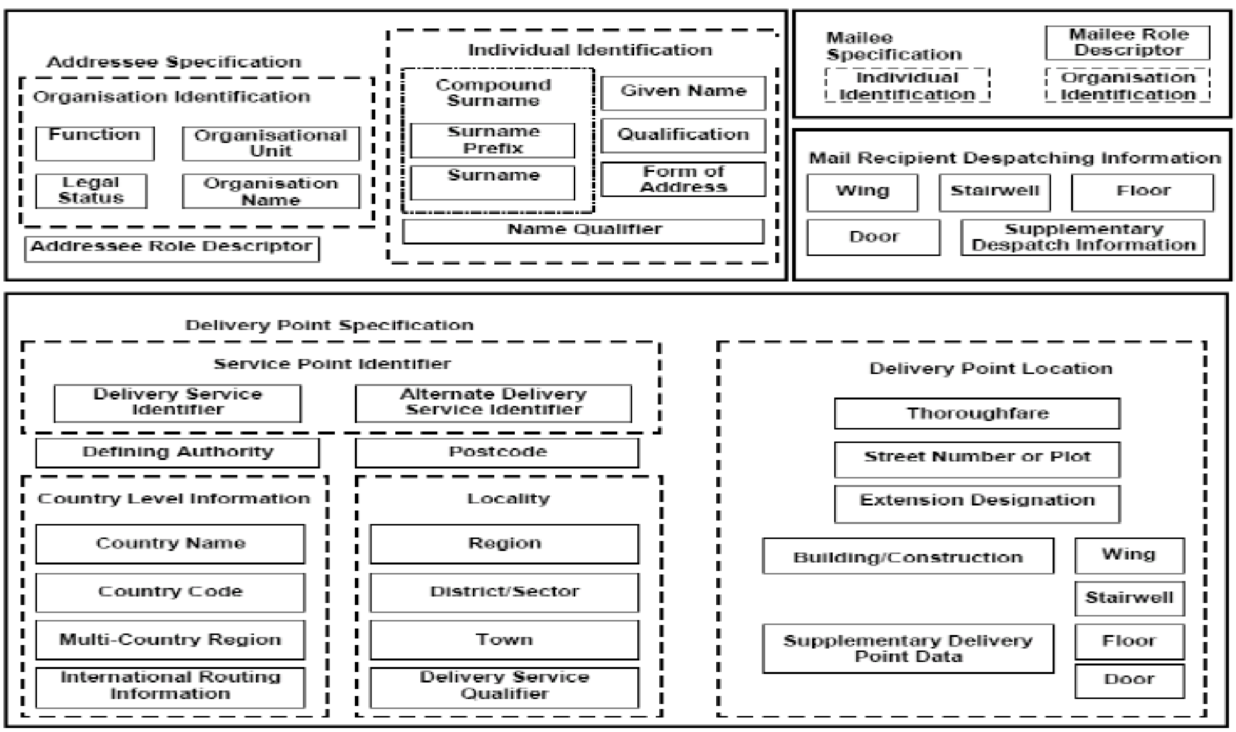


Figure 31: UPU-S42 - Postal Address Component (UPU/POC Addressing Group 2006a)

Developing UPU-S42 ontology was two stage processes which include generating UPU profile for the base address conceptual model and developing the UPU-S42 ontology from the profile. The UPU-S42 addressing conceptual model (figure 31) was converted to the UML class diagram as a UPU profile for the base address conceptual model. The mapping of segments, constructs, and elements of the UPU-S42 conceptual model in to ontology classes and properties is not directly related. The mapping of classes, attributes, and relations from a UML class diagram to ontology concepts and properties is, however, straightforward. In fact, ontology development software (TopBraid) provides tools to reverse engineer ontologies to UML-like diagrams for the easier presentation of ontologies to humans.

The UPU-S42 UML profile is developed with the help of, and comments from, domain experts. It is not, however, a UPU accepted UML equivalent of the UPU-S42 model.

The concepts that are in the base tier are not represented by the same terminologies in the UPU-S42 model, but they are generally represented in the model. For example the “Address” concept is at the centre of the base conceptual model (figure 5), and it is also represented in the UPU-S42, but it is broadly defined as the “Delivery Point Specification” construct.

The “Delivery Point Specification” construct has four segments and 21 elements that describe a postal address. The “Addressee Specification” and “Mailee Specification” constructs which describe addressable objects have three segments and eleven elements each. The last construct, “Mail Recipient Despatching Information”, has five elements (no segments) that describe an address. The segments and elements of “Delivery Point Specification” and the “Mail Despatching Information” make address components.

All the segments, constructs, and elements of the UPU model (figure 31) were represented in a UML model (figure 33) in a way that no element is left unrepresented. Some of the elements in the UPU model are big enough as concepts to be UML class on their own (for example element “Thoroughfare”) while other elements are not, and they become attributes of other classes (for example “Delivery Service Qualifier”).

Six address component sub-classes were extracted from the four constructs and 21 elements of the “Delivery Point Specification” segment and were added to the UML model. These classes (and their attributes) from the UML model are later mapped to OWL ontology concepts and properties. It is easier and more direct to map from a UML model to OWL ontology. Each class of the six “Address Component” sub-classes becomes owl:class in the UPU ontology, and the attributes become properties (owl:DatatypeProperty) of the classes (figure 32).

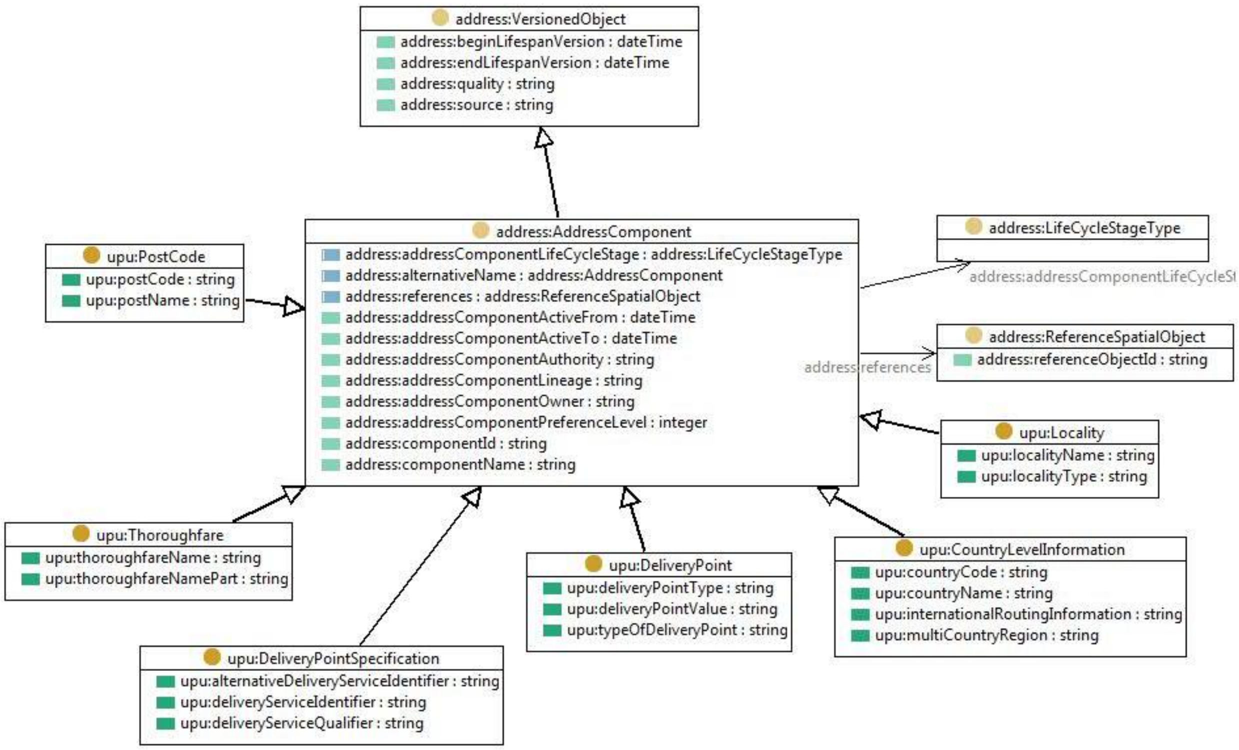


Figure 32: UML-like representation of the sub-classes of Address Component in UPU ontology

The class "Country" represents all the four elements of the "Country Level Information" construct (figure 31) which describes the elements applying to countries or groupings of countries (UPU/POC Addressing Group 2006a).

The class "Delivery Point" in the UML model (figure 33) represents the "Delivery Point Location" construct in the UPU model that identifies one or more delivery points that a postal operator can use by reference to geographical or spatial data expressed in human intelligible form (UPU/POC Addressing Group 2006a). Another UML class, "Delivery Point Specification", represents the "Service Point Identifier" construct in the UPU model that uniquely identifies a delivery point within a country without requiring a reference to its physical location (for example, post office box numbers).

The "Locality" construct of the UPU model, which identifies the geographical area in or adjacent to which a delivery point is located, is represented as a class with two attributes in the UML model.

The last two sub-classes of "Address Component" are "Thoroughfare" and "Postcode" which are just elements in the UPU model, but, since they are significant concepts, they are represented as classes in the UML model. These classes represent concepts that are important and which cannot be represented by a single attribute.

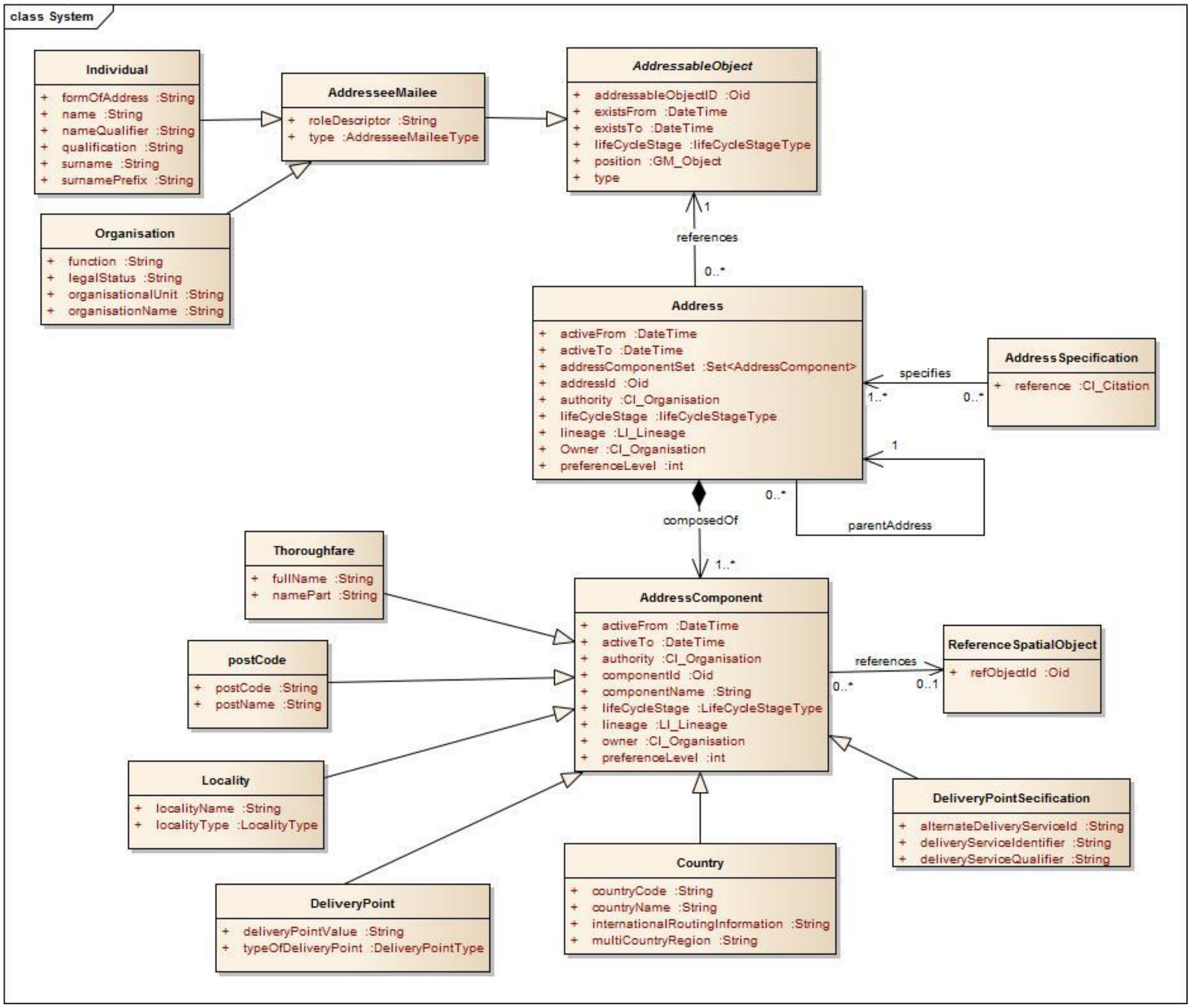


Figure 33: UML-class model for the UPU-S42 address standard

The UPU-S42 standard has two "Addressable Object" segments, the "Mailee" and "Addressee", which are the sender of the mail and the receiver of the mail respectively. These two segments have (the same) three constructs and 11 elements each. A class "AddressMailee" is, thus, included in the UML model to represent the two segments which will be identified based on the value of attribute type (which is either "Addressee" or "Mailee"). An "Addressee" or a "Mailee" can be either an individual or an organization. Two more classes (that are “Individual” and “Organization”) are, thus, added to the UML model to represent the two constructs.

The domain ontology for UPU S42 (figure 34) is developed from the conceptual model illustrated in figure 33 by mapping all the concepts in to OWL classes and their attributes and relations to OWL properties. Proper annotations and constraints were made to embed the semantics of the UPU terms in to the resources. Restrictions were made on all the properties to prevent wrong inferences, increase useful inferences, and assist proper linking of individuals to other individuals or literals. The domain and range for all resources were assigned carefully and again, just as in the previous ontologies, all properties were limited to only one domain and range. The cardinality constraints for the properties are set in accordance to the conceptual model.

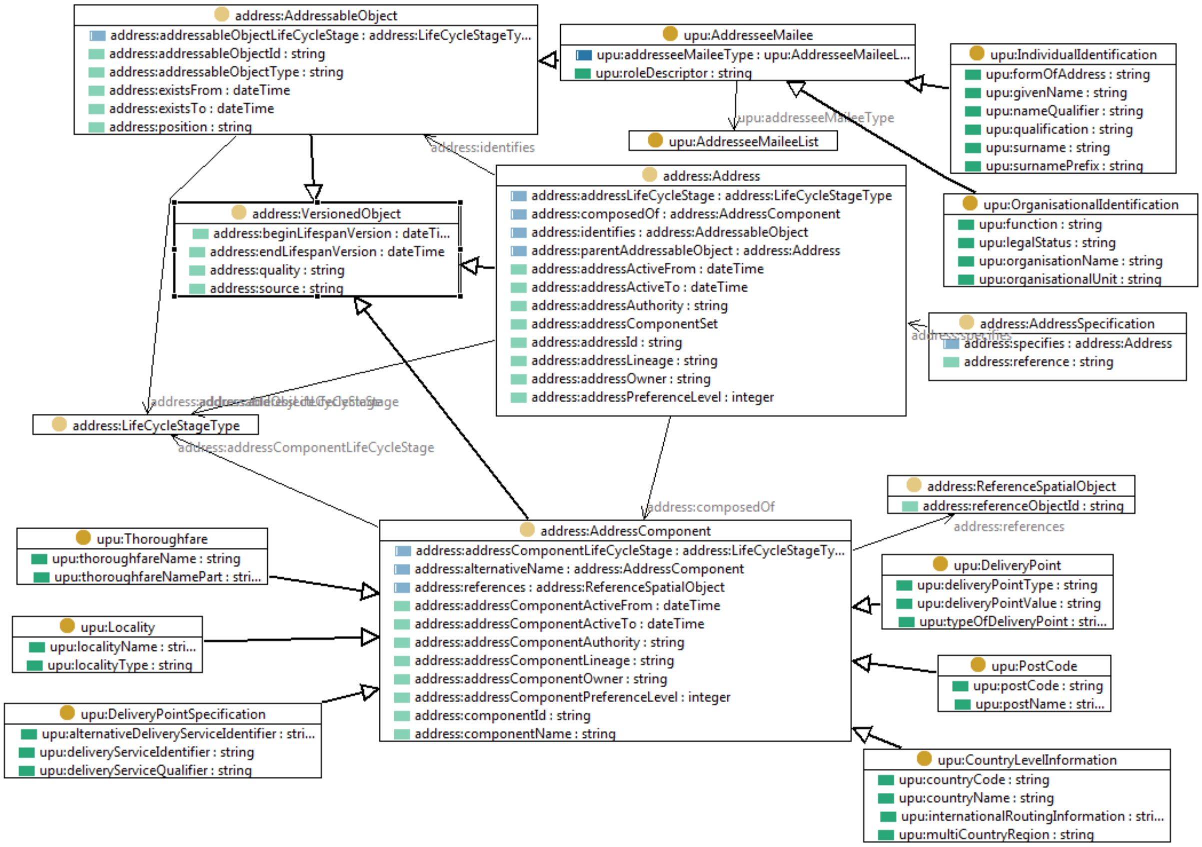


Figure 34: UML-like model of the UPU ontology

Figure 34 illustrates the UML-like model that was generated from the UPU OWL ontology. The ontology is big and is not easily understandable to humans, and so the UML-like model is presented to demonstrate the content of the ontology.

4.5 Application Ontology

The last layer of the four tier ontological hierarchy is the Application ontology. Application ontology is an ontology designed for a specific use, task, or purpose (Malone & Parkinson 2010). Application ontology extends domain specific ontology with the application of specific instances and concepts. Domain ontologies could be used for different applications, and, based on their purpose, they might have different instances. In OWL, ontology instances are values that are extensions of a class (Bechhofer et al. 2003).

For example, a simplified ontology for the domain of addresses in South African cities, which has three main concepts (as shown in figure 35), that is designed to be used by different municipalities for various purposes.

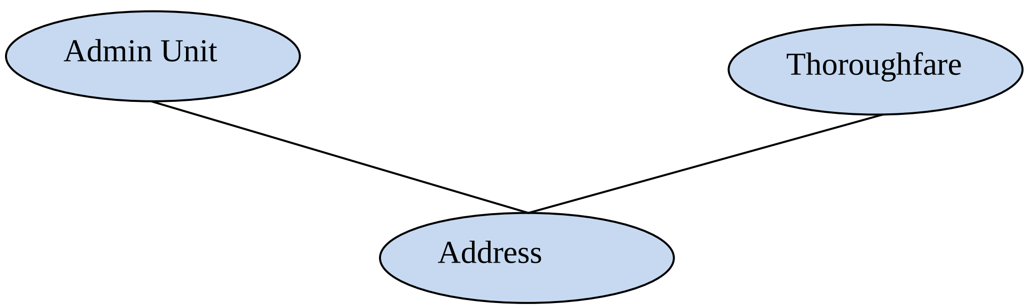


Figure 35: Simplified South African Addresses Ontology Model

Different SA municipalities can use the South African address domain ontology (figure 35) to record addresses in their jurisdiction. The instance extensions for one municipality might not be useful for another one, so every municipality can take the domain ontology and extend it with their own instances. For example, the address instances of Pretoria municipality (figure 36) are not useful for the administration of the Cape Town municipality.

The above examples are used to demonstrate (in simplified way) the necessity of the fourth tier of the ontology hierarchy which primarily adds the instance extensions to the domain ontology concepts. The domain ontology concepts can be used by many sub-domain ontologies. In fact, in the example above, the address ontologies of the different municipalities could be considered as sub-domains of the bigger South African address domain ontology. Research has shown that the use of multiple ontologies specializing in sub-domains can improve information extraction and inferences (Wimalasuriya & Dou 2009).

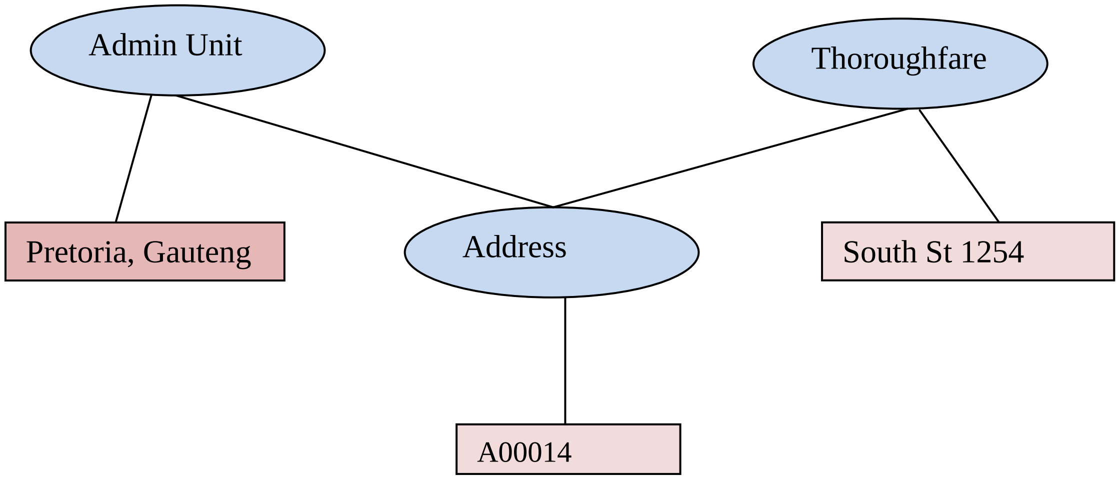


Figure 36: Sample (simplified) Address ontology model for an instance address in Pretoria

INSPIRE address ontology which is address domain ontology for 27 European countries can have multiple sub-domains. The sub-domain ontologies are basically the application ontologies that are used for special purposes, such as administrative addresses in individual European countries. These application ontologies have instance extensions of the specific sub-domain and its unique concepts.

On the other hand, application ontologies can be generated by integrating multiple sub-domain ontologies that provide different perspectives of the same domain. For example, in the domain of European Addresses, the ontologies of European Administrative addresses and European postal addresses could be integrated for better information extraction and inferences. In this case both ontologies give different perspectives to European addresses.

Three application ontologies were designed to illustrate the use and advantages of address ontologies. The first application (UKdata) extends the INSPIRE vocabulary with UK specific addresses and address concepts. In fact, the *UKdata* ontology is a specialization of the UK administrative addresses sub-domain of the INSPIRE (European) address domain. The second application ontology is *UPUdata* ontology that specializes in UK postal delivery addresses, which is a sub-domain of the UPU (global) address domain. The first two application ontologies specialize in sub-domains, while the third one (*UKdata+UPUdata* ontology) combines different perspectives of a domain. *UKdata+UPUdata*is are applicable in the domain of UK addresses.

UKdata+UPUdata ontology uses *UKdata* ontology and *UPUdata* ontology, which are both application ontologies that provide different perspectives of UK address domain. The *UKdata+UPUdata* ontology uses different axioms and restrictions to improve information extraction and inferences from the multiple ontologies with different perspectives of the same domain. The aim of application ontologies that combine different perspectives of a domain is to facilitate interoperability, sharing, and exchange of information across domains and perspectives of domains.

4.5.1 UKdata Ontology

UKdata ontology is an application ontology that references the INSPIRE domain ontology and is developed for the purpose of addressing (in accordance to the INSPIRE specifications) in the United Kingdom (UK). This application ontology extends the INSPIRE domain vocabulary with UK specific concepts.

The *UKdata* ontology can effectively be considered as a UK address sub-domain of the INSPIRE (European address) domain. It is an application ontology for UK administrative data that is obtained from the INSPIRE data specification document (INSPIRE/TWG/ Addresses 2009) among others. This application ontology is more effective in extracting UK specific address information than an ontology that is common to all European countries.

It is not easy to present the entire ontology in a document like this. The best way to present the ontology is to use a graphical representation of a carefully selected part of the ontology (see figure 37). In this, and subsequent presentations, one set of address data is consistently used to avoid confusion and for an easier explanation of the purpose and the goals of the ontologies (especially interoperability across domains). All the other sets of data are almost like the one that is presented except with different values.

The UKdata application ontology holds only UK addresses so only UK specific queries and inferences can be made. SPARQL queries can be used to find out information, such as the number of streets in a city, the number of houses in a municipality, the old name of a street, and its date of validity, amongst other things.

Instances in OWL ontology are extensions to the concepts of the ontology. Figure 37 shows a simplified model for a single address instance of London. Owing to a shortage of space, the model in figure 37 does not show the properties (and their literal values) of the resources. Figure 38, however, shows some parts (resources) of the address with their properties and literal values. In figure 37 the boxes with small purple diamond icons are instances of the boxes with golden circles which are concepts of the domain vocabulary and the text in the middle of the arrows are properties that relate the resources. All the concepts (URI resources) that are UKdata specific have "uk" prefix which is a name space for http://example.org/UK#.

The graph in figure 37 shows the (subject, predicate, and object) triples that make up an address ("`uk:A01`") in the UK. This address is used throughout the document. Since OWL is a language that embeds semantics to web data, questions such as "what is uk:`A01`?" can be asked (queried) using query languages (such as SPARQL) to find its meaning. The meaning could be retrieved from the ontology. For example, one description of the `uk:A01` could be expressed using the [subject, predicate, object] triple [uk:`A01, rdf:type, address:Address]` which indicates that "`uk:A01`" is an instance of "`address:Address`". Further follow-up queries could be used to find more answers to queries such as "what is `address:Address?`", "what is `uk:A01` composed of?" and, once we know the components, we can query "what are these components?".

As shown in figure 37, the address `uk:A01` is composed of thoroughfares "`uk:HighStreet`" and "`uk:MiddleStreet`", postal descriptors "`uk:LW_18_1ED`" and "`uk:SW_18_1ED`", address area names "London" and "Fairfield" amongst other components. Queries such as "why two thoroughfares or postal descriptors for the address", could, furthermore, follow-up to describe the components. Figure 38 shows what these components are and why there are multiple components of the same type. For example: `uk:MiddleStreet` and `uk:HighStreet` are the thoroughfares for the same address; the former one, however, is a retired ("`address:Retired`") and latest one is an active ("`address:Active`") address component.

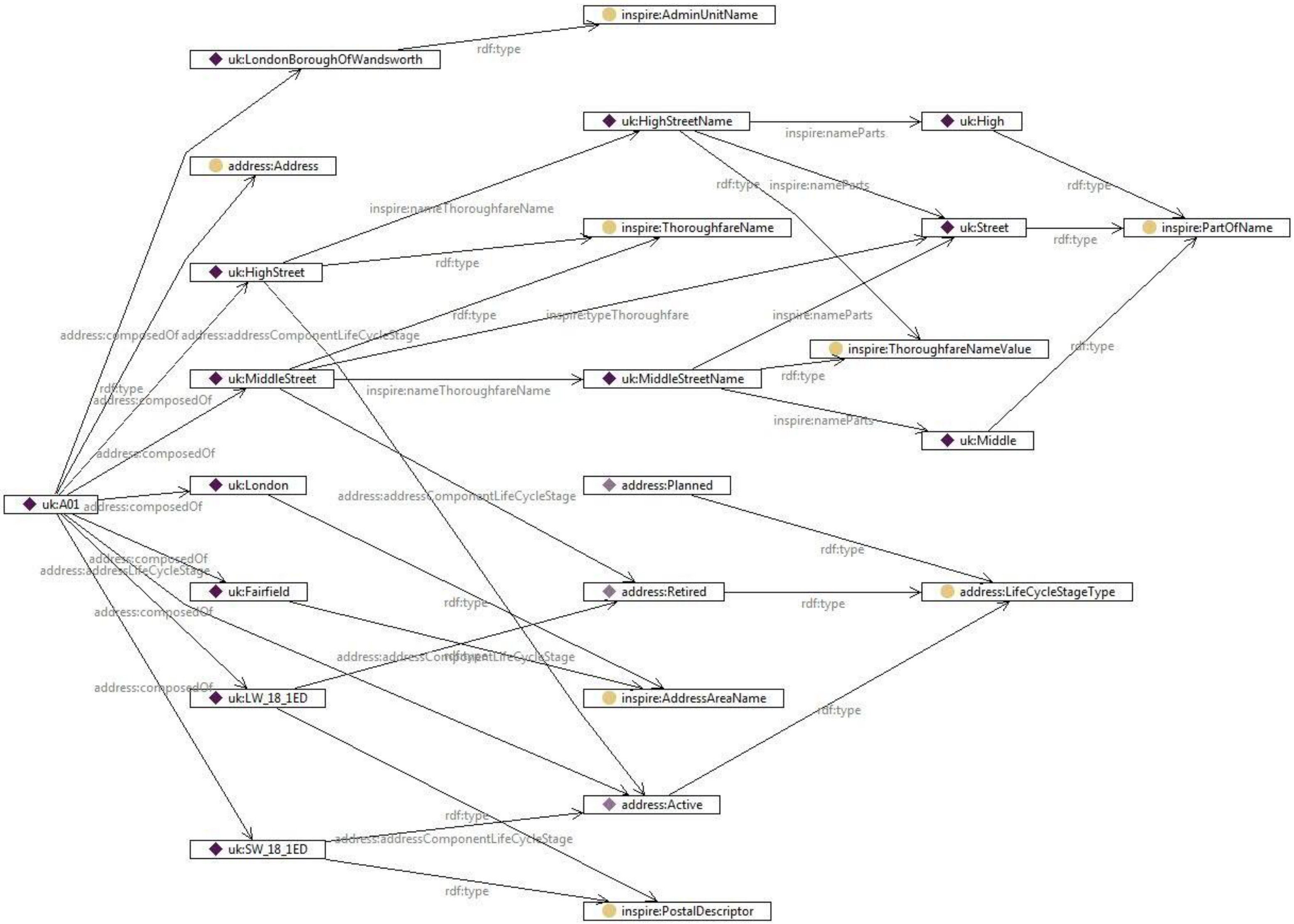


Figure 37: Simplified representation of an address in UKdata ontology

Figure 38: Partial Representation of a UK address data with literal values

4.5.2 UPUdata Ontology

UPUdata ontology is an application ontology that extends the UPU domain ontology. UPUdata ontology is developed for the purpose of addressing for postal service delivery (in accordance to UPU specifications) in the UK. This application ontology extends the UPU domain vocabulary with UK specific concepts.

UPUdata application ontology specializes in the UK postal address sub-domain of the UPU (global) postal address domain. This application ontology is designed to be more effective in extracting and making inferences of UK specific postal address information than a postal address ontology that is common to all countries of the world. UPUdata ontology uses UK specific axioms and restrictions to achieve its goals of improved information extraction and better inferences.

Once again, the best way to present the ontology in a document like this is to use graphical representation of important parts of the ontology. Figure 39 shows a simplified model for an address instance in London, UK. The same address as in the previous section (UKdata ontology) is used. In figure 39 the boxes with small purple icons are instances of the boxes with golden circle icons. In fact, in this case all the boxes with golden circle icons are concepts borrowed from the UPU domain vocabulary.

The graph in figure 39 shows the ontology triples (subject, predicate, and object) of the resources that represent a complete address ("upu:A01″). Each box represents a resource (rdfs:Resource) which is indicated by its name in the top row and its properties (and their literal values) in the second row. The graph (figure 39) models the meaning of the address "upu:A01". The ontology describes what "upu:A01" is, its components, and their properties. For example: the (subject, predicate, object) triple ("upu:A01", rdf:type, "address:address") describe that the subject resource "upu:A01" is an address. Another example of two triples ("upu:A01", "address:composedOf", "upu:UnitedKingdom") and ("upu:UnitedKingdom", "rdfs:comment", "European Country") describe that the address ("upu:A01") has a component "upu:UnitedKingdom" which is a "European Country".

The ontology that is represented by the graph in figure 39 can be summarized as: "upu:A01" is an address composed of five components and references and addressable object "upu:Lloyd". It indicates, furthermore, that "upu:Lloyd" is a banking organization. The components of address "upu:A01" show that it is in the country United Kingdom, city of London, Fairfield suburb, High street and postal code SW 18 1ED, amongst others. [N.B. the data used here do not reflect a real life address or organization.]

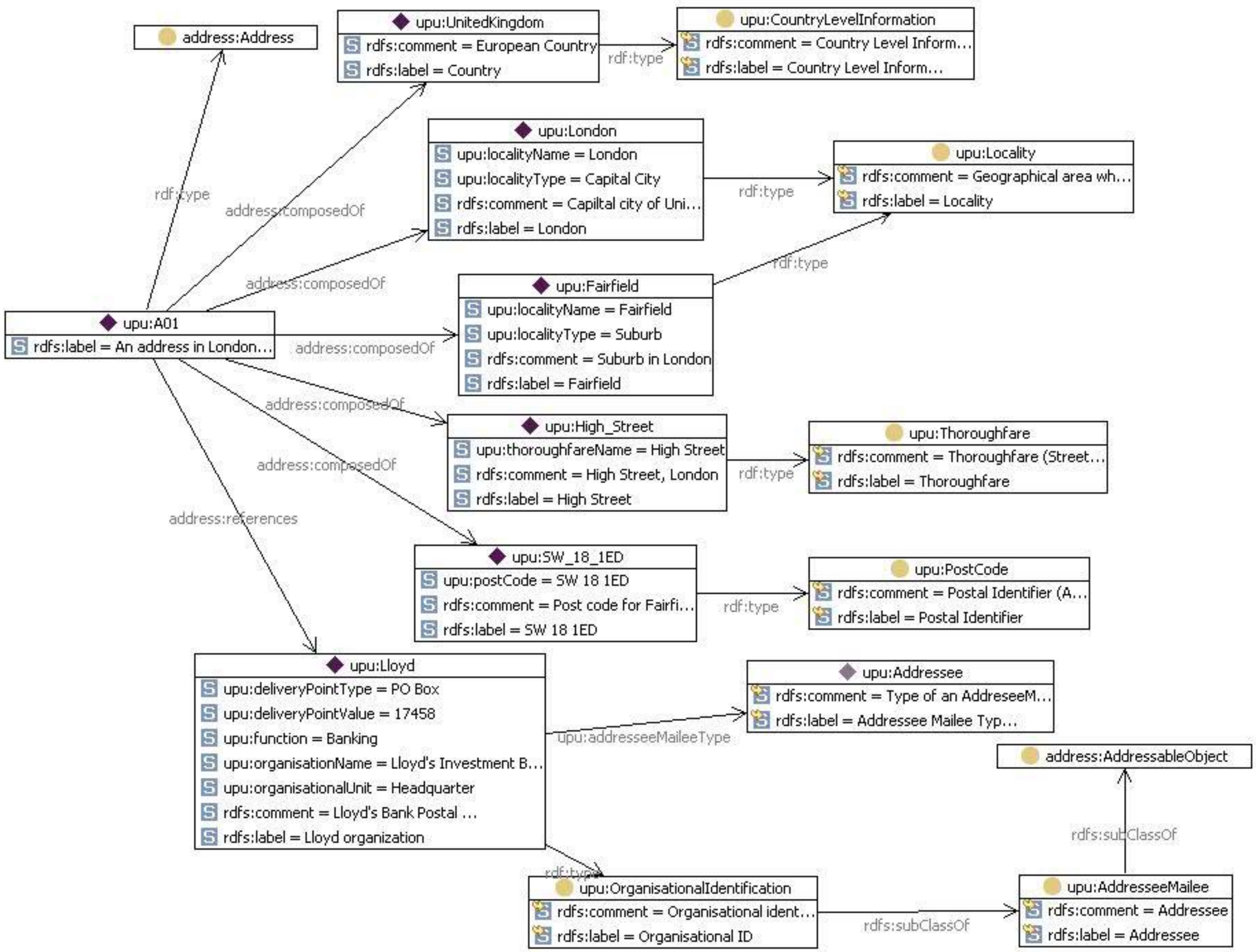


Figure 39: An address in UPUdata application ontology

4.5.3 UKdata+UPUdata Ontology

In the previous sections, two application ontologies that specialize on sub-domains of their respective domains were demonstrated. UKdata ontology (discussed in section 4.5.1) is application ontology for government administrative purposes that are applicable in the UK address domain. UPUdata ontology is also application ontology in the UK address domain but for postal service delivery purposes. Both UKdata and UPUdata ontologies are application ontologies for the UK address domain, but they provide different perspectives.

Multiple application ontologies with different purposes can be developed for a single domain. These application ontologies could be specializations in sub-domains or different perspectives of the domain. For example, as demonstrated in the previous sections, under the domain of European addresses multiple application ontologies for each of the member nations of the European Union can be developed. This was demonstrated by the UK address sub-domain of the European addresses that is used in the previous sections. When using application ontologies that specialize in sub-domains, each ontology is more efficient in information extraction and inference making in its own sub-domain (Wimalasuriya & Dou 2009).

Not all application ontologies, however, specialize on sub-domains. Sometimes multiple application ontologies give different perspectives of a domain. These kinds of ontologies (with their different perspectives) can be used to share information across the ontologies (perspectives) and to extract information that would otherwise be difficult if not impossible to get from one perspective. In this section, two different perspectives of the UK address domain are used to demonstrate extraction, sharing, exchange, and inference of information.

Since the address ontologies are quite big and complicated, a scenario about a college, with two small ontologies, is used to demonstrate the concept of information sharing and extraction using multiple ontologies that provide different perspectives of a domain. Once the idea and methods are explained with the college scenario, the same process is applied to discuss address ontologies.

Consider two application ontologies, which are about the academic and sports activities of a college. These two simplified ontologies provide different perspectives of the college, and they are built from two separate vocabularies (see figure 40 and figure 41).

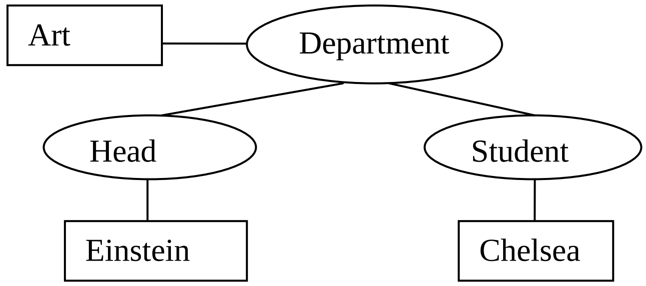


Figure 40: Simplified model of College ontology with Academic perspective

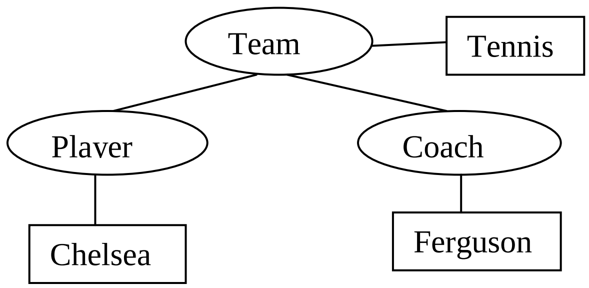


Figure 41: Simplified model of College ontology with Sports perspective

The two ontologies, represented in figure 40 and figure 41, are two networks of concepts that represent information about students in a college from two different perspectives. In order for information to flow between these two networks of concepts there needs to be a bridge (common concept) that connects them. For example, before setting the exam time-table a lecturer, who has only the academic perspective of student data, might want to know which sports teams his/her students play with. In figure 42, a simple bridge is made between the two networks of concepts.

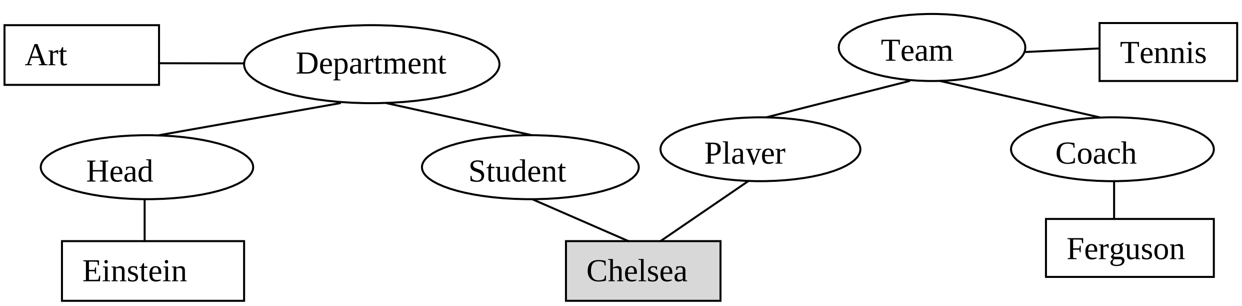


Figure 42: Simplified college ontology model with both academic and sports perspectives

The bridge between the two ontologies (that have different perspectives of a domain) is built by relating concepts across perspectives. Networks of concepts can be related (connected) using class axioms, property constraints, rules, and restrictions of/on their concepts. For example, setting equivalence between two concepts from both perspectives could create a bridge between them. In the case of the college ontology models, equality between the "Chelsea" instances of both perspectives is used (in figure 42) to demonstrated the flow of information. The resulting application ontology has a bigger network of concepts that span both perspectives. It allows better extraction and sharing as well as inference of information.

The ones who had only an academic perspective will gain sports information and can answer queries such as:

- How many sports teams are there in the college?
- How many students in department 'X' participate in college sports?
- Who is the head of department 'X'? And who is the coach of team 'Y'?

On the other side, the ones who had only a sports perspective will be able to have academic information as well and can answer queries such as:

- How many departments are there?
- To which department do the players belong?
- Who are the lecturers and heads of departments of student 'Z'?

The queries asked above need to use both perspectives to obtain the correct information. Sharing of information across the two networks of concepts is, therefore, necessary to provide a better extraction of information. Since, furthermore, ontologies enable machines to capture the semantics of data and make inferences, important information can be generated to answer questions such as:

- Which departments have the most students who participate in sports?
- Is there a relationship between departments and type of sport played by students?

The inferences can then be used in further decision-making processes.

The example used above is very simplified but it shows the idea behind use of ontologies to share, extract, and exchange data as well as to facilitate the inference of information. In the example, the instance data of only one student was used; in reality, however, there would be many students. Using the first name to set equivalence (to be used as a bridge) to connect the two networks of concepts is, thus, not practical. Unique student identifiers such as student ID numbers can be used to set equivalences. The equivalences, thus, need to be set between abstract concepts rather than instances. In the example, it would have been the Student ID concepts in both networks of concepts.

Similarly, for addresses the same principles apply when using ontologies to share, extract, exchange, and infer information from two application ontologies with the different perspectives of a domain.

UKdata+UPUdata ontology is application ontology that integrates two application ontologies with different perspectives of UK address data domain. As its name indicates, it combines both the UKdata (UK government administrative address data) perspective and UPUdata (UK postal address) perspective of the UK address domain. The purpose of the *UKdata+UPUdata* application ontology is to share and exchange address data between the government and the postal services provider, both of whom are the major sources of address data. Inferences, furthermore, could be done from the combined data to generate new information.

UKdata+UPUdata does not make further specialization on a specific sub-domain of the UK address domain. Instead it integrates multiple perspectives of the UK address domain. It builds bridges between the two networks of concepts to create a bigger network of concepts. The resulting network of concepts includes the concepts (and their semantics) from the two ontologies, and inferences are made, furthermore, to generate useful new concepts.

The *UKdata+UPUdata* ontology extends both the UKdata vocabulary (figure 37) and UPUdata vocabulary (figure 39) mostly with conditions and constraints that create and facilitate interoperability between the two application ontologies (with different perspectives). The vocabulary of *UKdata+UPUdata* (figure 44) builds on the top of the vocabularies of all the ontologies discussed in the previous sections (figure 43).

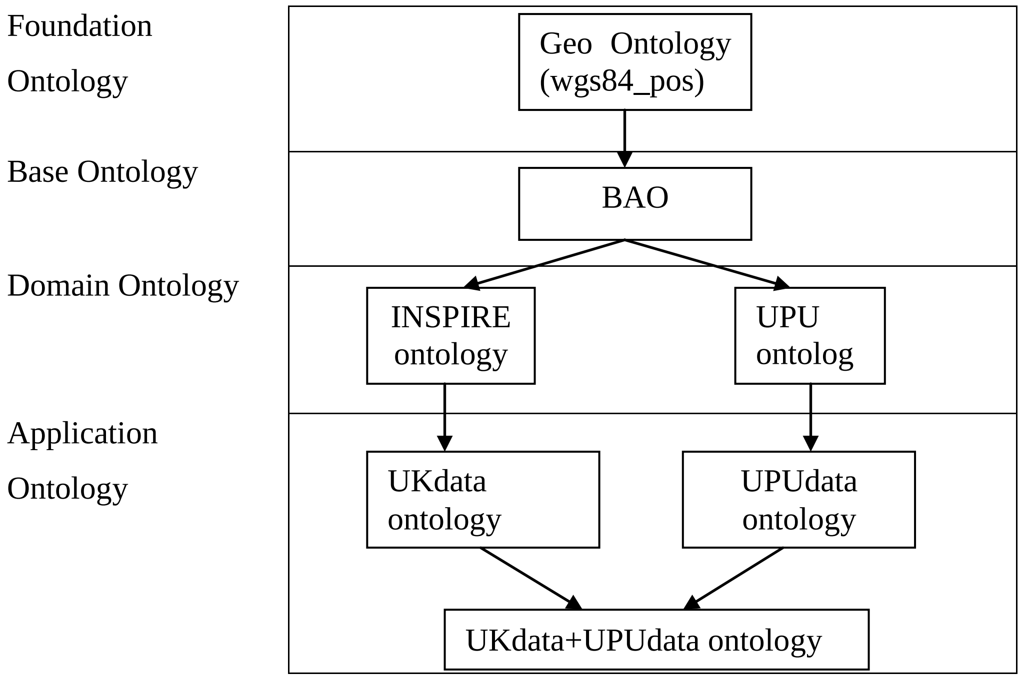


Figure 43: UKdata+UPUdata ontology's vocabulary inheritance

Both the UKdata and the UPUdata ontologies extend the BAO vocabulary. BAO is a small ontology that was developed to represent the basic address concepts that are used across all address domains. Once the two application ontologies (UKdata and UPUdata) were integrated they, thus, automatically get linked through their shared concepts from the BAO (see figure 44). The model in figure 44 is very simplified, but it shows that concepts from BAO connect the UPUdata network with the UKdata network of concepts. In reality, there are many addresses from the UK administrative database and many addresses from the postal services as well. As more address data gets represented, therefore, the model grows quickly both to the left (UKdata) side and right (UPUdata) side. Regardless of how big the model gets, however, it always remains connected by the shared concepts (from BAO).

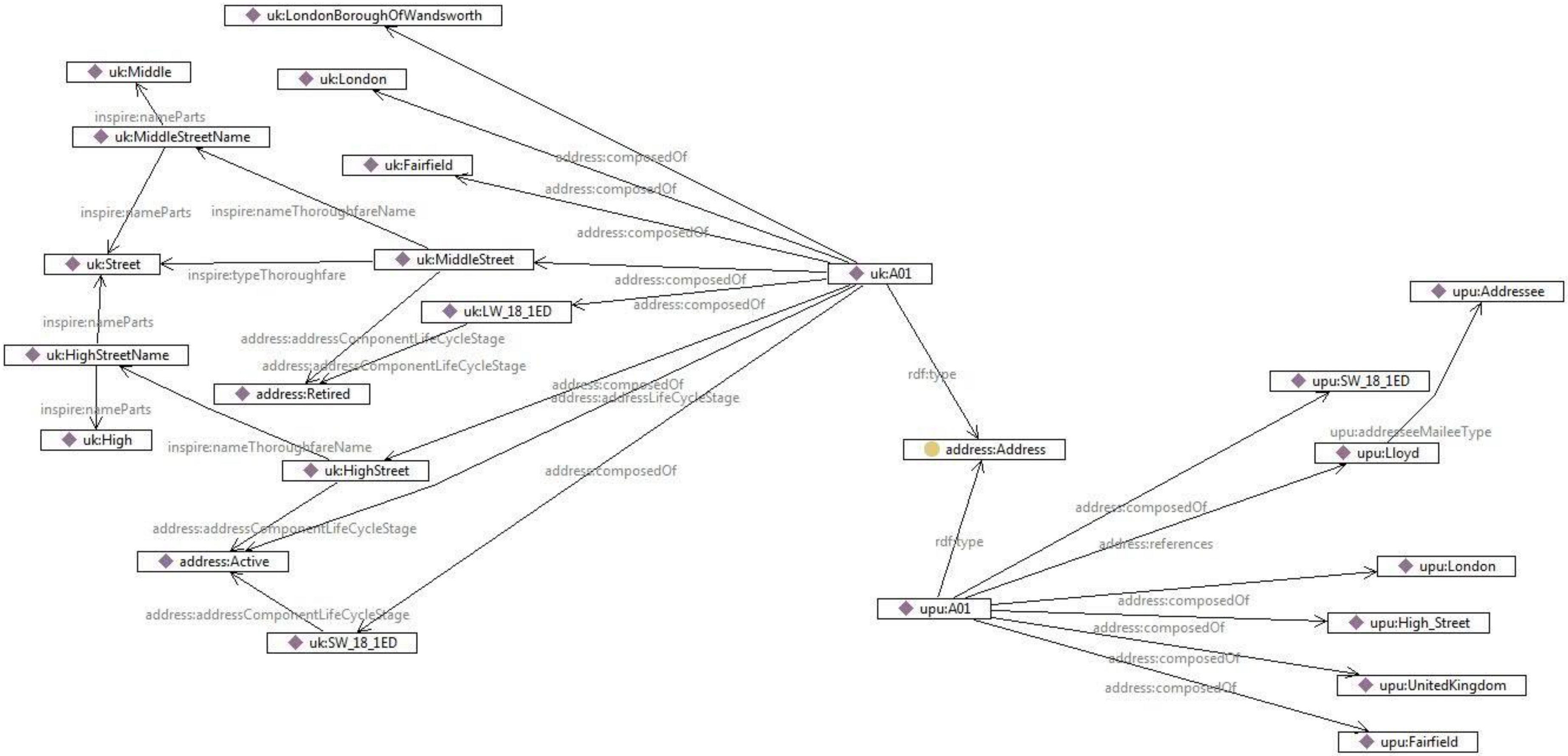


Figure 44: UKdata+UPUdata ontology before constraints and concepts are added

Two different representations of an address (based on two different standards) from the UKdata and UPUdata ontologies were used to demonstrate the process of facilitating the interoperability of address data and information extraction. Figure 44 shows that the two networks of address data concepts are connected via a single node (concept) `address:Address` from the BAO. The amount of new information that could be inferred and extracted simply from adjoining of these application ontologies is, however, small. This is because the flow of information in the network of concepts is limited to the one directional flow that points from address instances to the address concept (`address:Address`). To allow better interoperability and extraction of information, therefore, more concepts and/or instances across the perspectives need to be connected.

Consider the two addresses used to demonstrate interoperability (in figure 44). The two networks of concepts are connected with the two triplets, [`uk:A01, rdf:type, address:Address`] and [`upu:A01, rdf:type, address:Address`]. The only new information the users of UKdata (administrative data) gain from UKdata+UPUdata ontology at the initial stage (before constraints are added to the ontology) is that `upu:A01` is an address. On the other side, UPUdata users gain only the new informationthat`uk:A01` is an address. These two pieces of information do not tell UK administrative officer information about addressable object that could be found from the UPU data. Neither do they give any temporal information to UPU officer who should get it from the UK admin data.

To create a better flow of information class axioms, individual identity restrictions and property restrictions and relations need to be added to the ontology. Axioms and restrictions were added to infer answers to specific queries that are important to users of both perspectives of the address data.

Some of the address information is visible only from one perspective. For such information to be available for users with a different perspective, it is important to connect (by mapping) the concepts and instances across perspectives. As a result, a broader view (network of concepts) of the address data is generated. Users with the UK administrative data perspective need to get information about the object identified by the address, which can be obtained from address data that is collected for postal services. Local administrators might have queries such as:

- What does an address identify?
- Does an address identify an organization? If so, then what kind of organization?
- Is it residential address?
- Is it a landmark?
- Is it an abstract delivery or postal address?
- How many residential addresses in suburb X?
- How many Banks in London?

The answer to such queries might be available from data collected with perspectives that differ from the administrative data. The answers to the above queries, however, are useful to the administrators.

Postal address data, on the other hand, has all the information about the addressable object identified by the address. Postal address data, however, lacks the temporal perspective of an address. Postal service agents can get temporal information about an address and answer queries such as:

- When was an address identifier assigned?
- Is the address identifier still applicable?
- Is there an old identifier for an address?
- Is the current identifier going to expire? If so, then when will it expire? And what would be the new identifier?

The answer to all the above questions requires information from both perspectives of the addresses. There should be interoperability between the administrative addresses and the postal service addresses. It is, thus, necessary to create more flow of information between the two sides of the `UKdata+UPUdata` ontology's network of concepts (figure 44). The two address instances (`uk:A01` and `upu:A01`) are URI references that actually refer to the same address in London, UK. To create interoperability between the two perspectives of addresses within the domain of UK addresses, `uk:A01` (an address with the administrative perspective) is, therefore, mapped to its equivalent `upu:A01` (an address with postal services perspective) with `owl:sameAs` individual identity restriction property of owl. Hence the triplet [`uk:A01, owl:sameAs, upu:A01`] is added to the `UKdata+UPUdata` ontology. The two representations are, thus, linked as identifiers of the same address. After adding the individual constraint, one round of inference was run, and it generated seven new instances and 28 properties to the ontology. Figure 45 shows part of the ontology after making the inferences. Compare figure 45 with figure 44 to see some of the new properties and classes added to the ontology.

The built-in owl property `owl:sameAs` indicates that the URI of two individuals (or even classes in the case of owl Full) identify the same entity (Bechhofer et al. 2003). Hence they share all their properties and extensions (instances). This basically indicates that the two identifiers of the address have the same intentional meaning.

Figure 45 shows the two address URIs (that identify the same entity) are linked with the `owl:sameAs` attribute. The inference engine was run once after the individual identity restriction was set. As a result of the inference, the two address identifiers shared each other's properties and relationships. All the information that identified the UK admin address, thus, also identifies the UK postal address and *vice versa*.

Finally, after extending the network of concepts, with the constraints and subsequent inferences, the answers to all the above stated questions that require both perspectives of the address can be obtained with the use of SPARQL queries.

Since the ontology is quite large to be presented (either as RDF/XML or graphically) in this document, only a selected part of it is shown in figure 45. It, however, presents the most important part of the discussion. The red highlighted part shows the temporal data that is obtained from the UK administrative perspective of the address. These instances (under the red highlight) were not connected to the postal perspective of the address (`upu:A01`) before the inference. Since the two address identifiers were established as the same using the `owl:sameAs` property, the inference engine inferred that the temporal data for the UK admin address data also identifies the UK postal equivalents. Finally, the postal service providers can make queries to get information that is provided by the UK admin data.

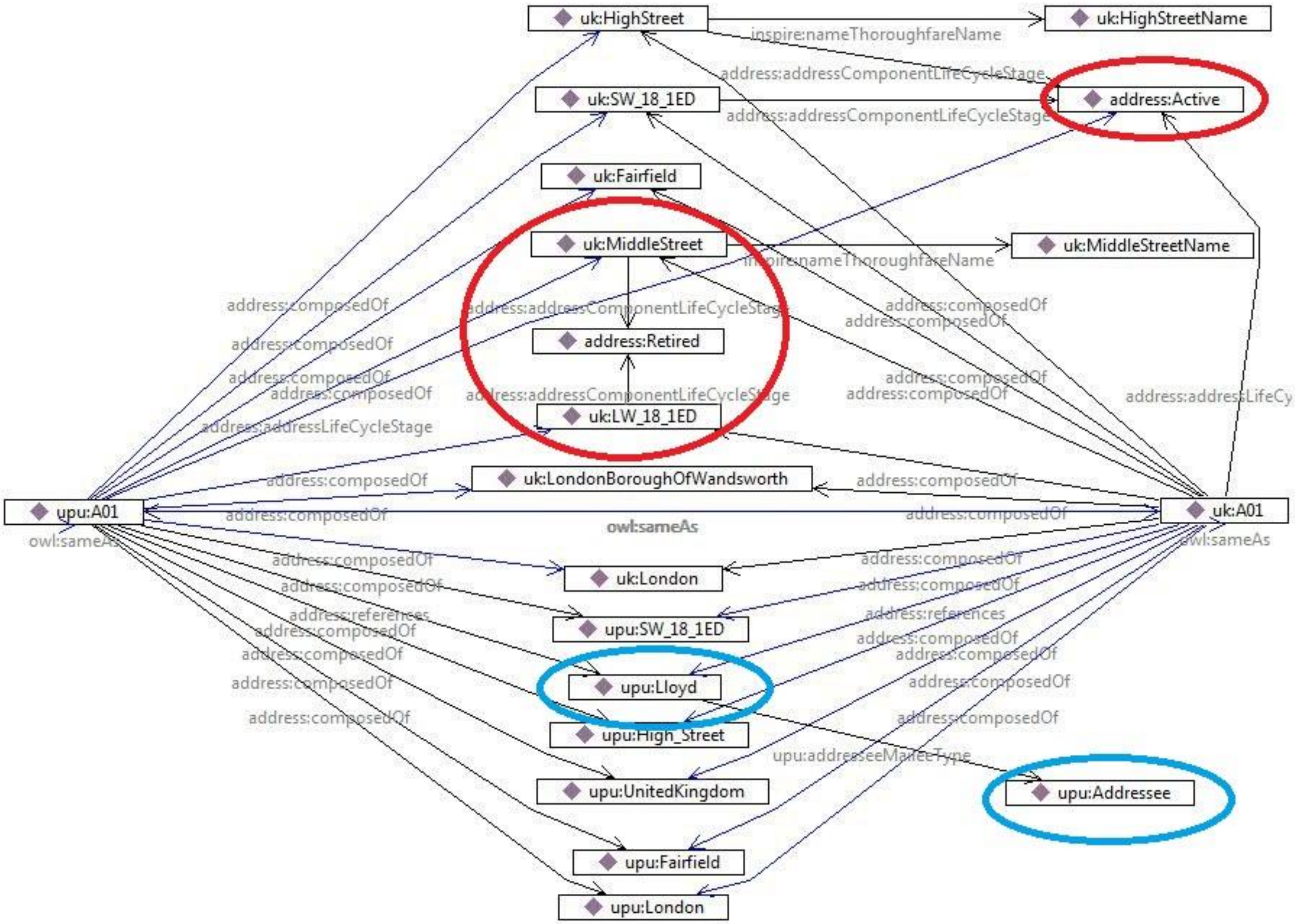


Figure 45: UKdata+UPUdata ontology after identity restriction and subsequent inferences

Similarly, the blue highlight in figure 45 shows the information that is gained from the postal services address by the UK admins. The UK admin can also make queries about the addressable objects in an address. For example, the SPARQL query below was used to find the addressable object at address `uk:A01`.

```
SELECT ?addressableObject
WHERE {
uk:A01 address:references ?addressableObject .
}
```

ANS: `upu:Lloyd`

Even though the addressable object information is not in UK admin data, since `uk:A01` is established to be the same address as `upu:A01`, the addressable object data of the `upu:A01` can be used to answer the query. Information about the addressable object can, furthermore, be extracted using SPARQL queries such as:

```
SELECT *
WHERE {
upu:Lloyd ?predicate ?object .
}
```

ANS: The details of Lloyd.

The result of the above query is the detailed information about the addressable object `upu:Lloyd`, which can be summarized as in figure 46. The information shows that the address (hypothetical and not real) belongs to a bank's headquarters in London.

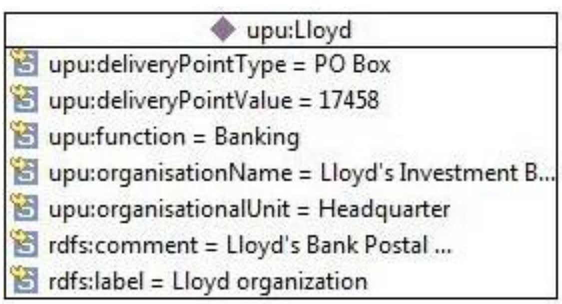


Figure 46: upu:Lloyd Addressable object and its properties

Similarly SPARQL queries can be used to extract information, which can be used by postal services providers, from the UK admin address data. The postal services providers, thus, get temporal information of their address data.

4.5.4 UKpostal Ontology

The above-used scenario assumes the two ontologies have corresponding values for the address extensions on each side. Sometimes, however, when attempting to achieve interoperability across domains, one of the participating ontologies might not have a corresponding address instance (or even it might not have any address extension at all). In such situations, facilitating interoperability between ontologies by linking the instances (extensions) is not effective if not impossible.

For example, consider a situation where the local administration has addresses for a town that does not have any postal services. Say the postal service providers do not have any data about this town. Hence the local administrators cannot get the addressable object information about their address that they could have extracted from postal services data. The postal services could, however, benefit from the local administration's address data. Even though address data from the local administration does not have all the attributes required by the postal services, at least it has many attributes that are useful.

Setting equivalences and links between address instances does not always solve the problem, because sometimes there might not be an equivalent address to deal with. In such situations it is important to establish links between concepts (i.e. classes and properties) rather than instances in order to facilitate interoperability. To demonstrate the theory, an ontology called *UKpostal* ontology that incorporates the UKdata and UPU ontologies is developed (See figure 47). The UKdata ontology is an application ontology that is based on the INSPIRE domain ontology and specializes it with UK specific address extensions. The *UKpostal* ontology does not have any postal services specific address extensions. The aim of *UKpostal* ontology is initially to get address extensions from the *UKdata* ontology and complete them with UPU specific extensions.

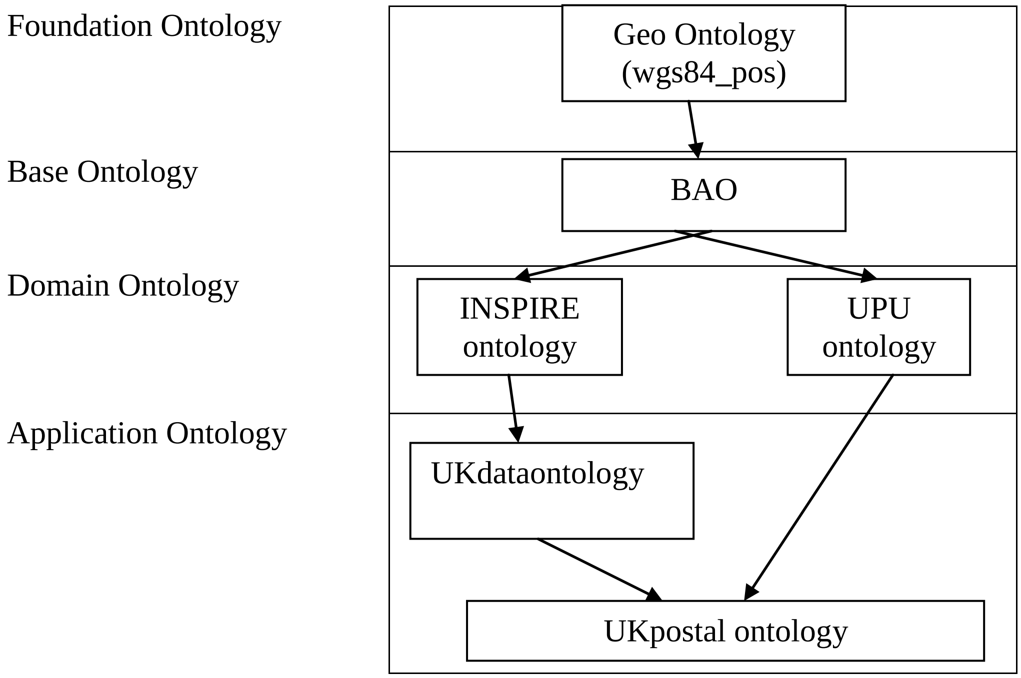


Figure 47: UKpostal vocabulary inheritance

The vocabulary of UKpostal ontology extends UKdata and UPU vocabularies. The UKpostal ontology has all extensions of the UKdata ontology. The address instances of UKdata are used for postal services by the UKpostal ontology. The address (instances of) UK administration are gathered based on the INSPIRE specification and, hence, they do not have some of the address attributes required by UPU-S42 specification.

In order for the address extensions of UKdata ontology to be useful for postal services, the concepts across the two domains (INSPIRE and UPU) need to be linked (with class axioms, property equivalences, and individual identity restrictions). The concepts to be linked should be carefully selected, and the type of restriction used must also be appropriate. Mistakes in relating concepts could lead to wrong inferences and information. Equivalent concepts (classes and properties) from the two domains were, therefore, selected carefully (as highlighted in figure 48).

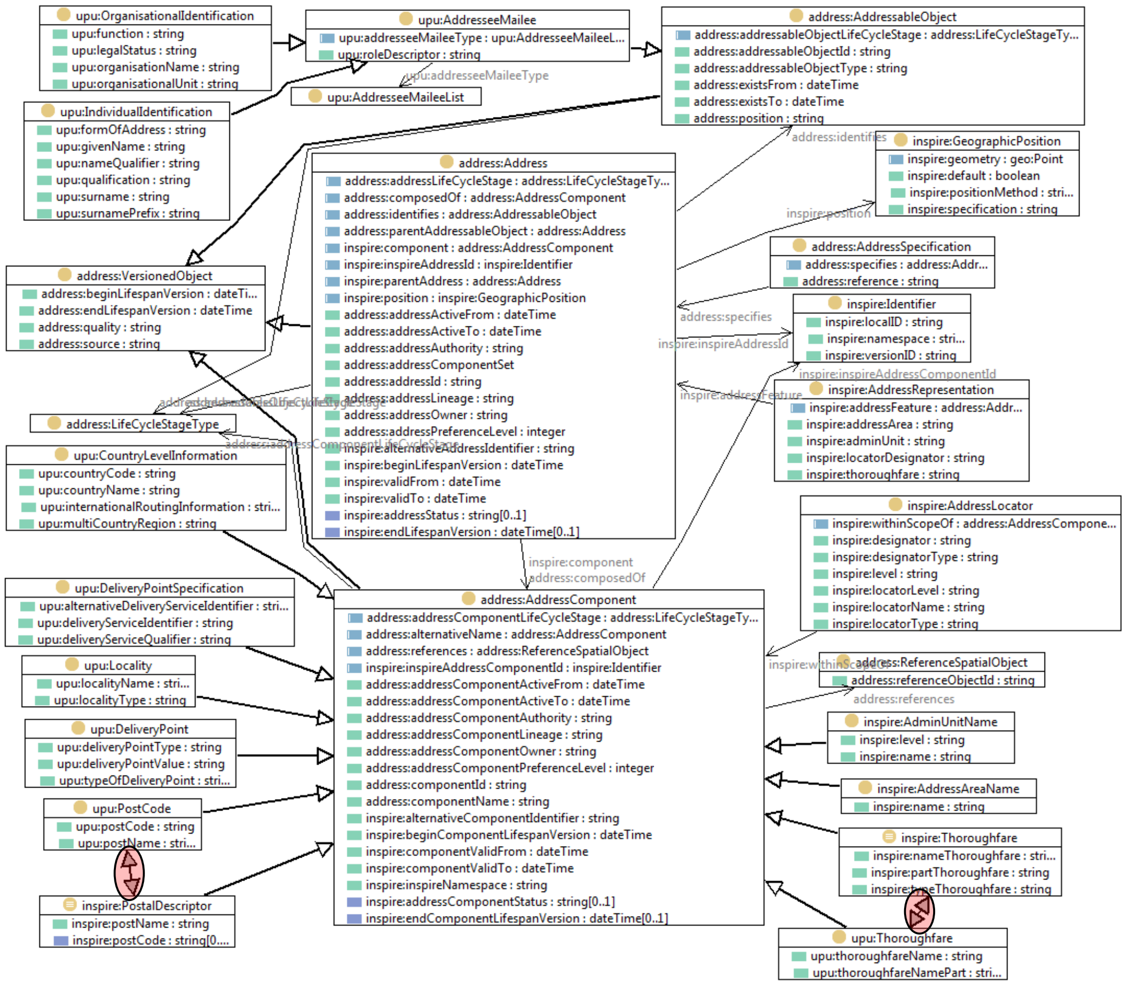


Figure 48: UKpostal ontology concepts

The classes that were identified to be equivalent in both domains are `upu:PostCode` and `inspire:PostalDescriptor`, as well as, `upu:Thoroughfare` and `inspire:Thoroughfare`. These classes are equivalent but not necessarily the same. The equivalent classes were linked using the `owl:equivalentClass` class axiom property. The equivalent classes share the same extensions, but they do not necessarily have the same meaning. For example, the extensions of class `inspire:Thoroughfare` will also be extensions of `upu:Thoroughfare`.

Not only class but also property equivalences were established between properties across the two domains. For example, the "`upu:postCode`" is set equivalent to "`inspire:postcode`" which leads the inference engine to infer that the post code of an individual in INSPIRE is the same as the post code in UPU. Very close care should, however, be taken when setting property equivalences, because wrong property equivalence leads to wrong inferences, which exacerbates quickly in multiple inferences, damaging the integrity of the ontology.

After the class axioms, property restrictions were added to the ontology and the inference engine was run. It inferred correctly that all the extensions of UKdata ontology could also be used for postal services as extensions of UPU attributes. The extensions were not duplicated and the same URIs are used for both domains. For the postal service address data to be fully operational, however, they need to get extensions for the attributes that are UPU specific.

The above two ontologies (UKdata+UPUdata and UKpostal) show that address data can be shared, exchanged, and extracted across domains. This creates interoperability across the domains.

Chapter 5

5 Discussion of Results

5.1 Introduction

The experiments conducted in this research study are concerned mainly with the knowledge representation aspect of the semantic programming approach used to facilitate address data interoperability. The major findings of the research are listed and briefly discussed in section 5.2. The ontology architecture which was evaluated to be optimal for the facilitation of address data interoperability is discussed in section 5.3. Section 5.4 discusses the foundation ontologies used in the research to develop efficient address reference ontology. One of the major objectives of the research was to develop a reference ontology for addresses, which is discussed in detail in section 5.5. Demonstrating the process of facilitating the semantic approach for address data interoperability requires domain ontologies and application ontologies to be used. Prototypes of two domain ontologies were developed (discussed in section 4.4) and they are evaluated in section 5.6. Finally section 5.7 of the chapter discusses the prototype application ontologies and evaluates the semantic approach by considering the different scenarios used to test the applications.

5.2 Research Results

The major goals of the research were to find an optimal address knowledge representation model and address reference ontology that would facilitate interoperability effectively and easily. The research has identified that four-tier address ontology architecture gives the optimal knowledge representation model. Section 5.3 argues the case for four-tier architecture being identified as optimal and how the ontologies are sorted into hierarchies.

Facilitating across domain interoperability using the semantic approach is best served by the use of efficient reference ontology. Hence, one of the major goals of the research was to develop optimal address reference ontology. The concepts of the address reference ontology do not have to be too specific or too generic in order for the reference ontology to be optimally used by domain ontologies. Section 5.5 provides a detailed discussion relative to this finding.

Other findings of the research include the identification of foundation ontologies that would be required for the development of efficient address reference ontology and the identification of ideal scenarios or best use cases for the semantic approach. The findings on the foundation ontologies are discussed in more detail in section 5.4, and the findings of the application ontologies usage scenarios are discussed in section 5.7.

The ultimate goal of the experiments was to create an interoperable system for across domain address data sharing, exchange, and processing using a semantic approach. Hence, the results of the experiment were two interoperable address data ontologies. These interoperable environments were created using the UKdata+UPUdata ontology and UKpostal ontology, which illustrate different use cases of ontologies that facilitate interoperability. Figures 43 and 47 show the hierarchy of ontologies used and the address domains used in the interoperability process. These results are discussed in detail in section 5.7.

5.3 Address Ontology Architecture

In the research study, the semantic approach was used to facilitate the interoperability of digital address data across domains. It was important to choose optimal ontology architecture to facilitate the process. The ontology architecture must accommodate the representation of all foundation concepts and the domain specific concepts.

There are many address domains that are based on vocabularies from different address standards. Some of the concepts that are necessary for one address domain are not useful in a different domain or may even be contradictory to concepts from another domain. The solution should, thus, include multiple ontologies rather than one huge network of concepts and instances. Using multiple ontologies makes it easy to separate the domains and extend the system with new address domains.

The need for multiple ontologies was evident from the beginning because the users of address data are many and there is no centralized way of storing or representing the address data. The proposed system to facilitate interoperability was designed to operate using multiple ontologies. The ontologies were classified into four tiers, based on their context, level of granularity, and purpose.

Knowledge in OWL ontology is represented as concepts, instances, and relationships amongst them. The rules that define how ontologies operate are also represented as [subject, predicate, object] triples. These concepts are defined with standardized URIs such as the RDF, RDFS, and OWL rules. It is important to keep these foundation concepts in their own tier to separate the domain logic from the foundation logic.

After separating the foundation tier, it was important to compare and contrast two, three, four, or even more tier architectures to find the optimal one. Two-tier architecture separates the foundation ontologies from the domain ontologies, but it would put all the domain concepts in one tier without considering their level of granularity and purposes. More classification is needed to solve the problem.

To facilitate interoperability across different address domains it was necessary to choose the most common and high level concepts (such as “Address” and “Address Component”) from vocabularies across all domains in order to use them as a basis for the interoperability process. These selected concepts had, thus, to be used to create the Base Address Ontology (BAO), which needs to be identified in a separate tier on the architecture.

The third tier was created by separating the Base Tier from the Domain-Tier (see figure 49). It is necessary to separate the Base-Tier from the Domain-Tier because it contains concepts that are shared by all address domains. The concepts of the Base-Tier are of higher granularity than the Domain-Tier concepts.

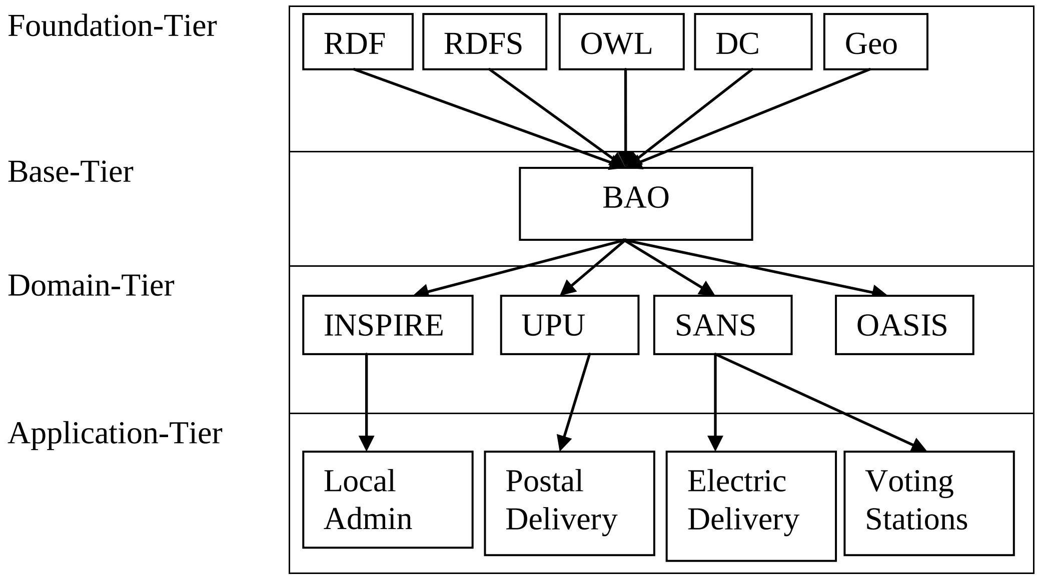


Figure 49: Address Ontology Architecture

Domain-Tier ontologies extend the Base Address Ontology (from Base-Tier) with domain specific concepts (figure 49). The concepts at the Domain-Tier are defined based on the vocabulary from the standards to which they refer. Often domain ontologies can be used for several purposes because they use concepts that are defined based on unique standards but not specific tasks. For example, INSPIRE domain ontology would define the address concepts used in the INSPIRE data specification, and it could be used for a variety of purposes. Mixing specific tasks to domain ontologies limits their usefulness. The application specific concepts and instances are, thus, separated from the domain ontology, hence creating the fourth tier of the architecture.

Separating the Application-Tier from the Domain-Tier allows for the use of concepts and instances from the Domain-Tier for many applications. Application ontologies extend domain ontologies; in fact multiple application ontologies can extend a single domain ontology and use it for separate purposes. Application ontologies add application specific concepts and (mostly) instances to domain ontologies. The application ontologies that are derived from common domain ontologies could be mutually exclusive. For example, two application ontologies designed for the administrative address data of France and Denmark could be extended from INPIRE domain ontology and are exclusively applicable to the respective countries.

Application ontologies can also be generated by extending other (one or more) application ontologies. They would, however, still be categorized under the Application-Tier because they would still be either a specialization or different perspective of an address domain. There was, thus, no need to create a fifth tier, and it was concluded that a system for address data interoperability works optimally with four-tier ontologies.

The following sections discuss each of the tiers of the address ontology architecture in detail. The next section discusses the foundation-tier and the ontologies that were used and considered in that tier. That is followed by a section that discusses the process of developing the base-tier and its contents. Discussion of domain-tier and the domains considered for this research follows the base-tier discussion. Finally, the application-tier and applicability of this research in real life is discussed in the final section of this chapter.

5.4 Foundation Tier

The Foundation-Tier is where all the foundation concepts for address interoperability process are located. In the context of the research conducted, the foundation concepts are high level concepts that are pre-defined in standardized ontologies (which are internationally accepted). These are concepts that do not need to be redefined by users for reasons of consistency, correctness of resources (i.e. URI), and efficiency of ontologies.

The Foundation-Tier is made up of multiple ontologies with concepts and instances that are needed to build the lower tier ontologies and especially the Base Address Ontology. The concepts of these ontologies include the rules and the basis of ontology languages and generic concepts that are also used for addressing purposes (such as geographic or spatial concepts).

The ontology language used to develop the ontologies was OWL and, as a result, the rules that define the proper use of OWL, and even the concepts that are required to develop an ontology, need to be included in the Foundation-Tier. These include concepts that define OWL principles, rules, and restrictions, such as Classes, Properties, and Individuals and their specific properties and interactions. The OWL concepts are found and used from the URI http://www.w3.org/2002/07/owl#.

OWL, as an ontology language, is an extension of RDFS. The RDFS concepts and rules, thus, also need to be included in the Foundation-Tier. RDFS, in turn, is an extension of RDF, and hence RDF concepts and rules were also included in the tier. RDF, RDFS, and OWL concepts establish the basic rules on how the ontologies should be used.

The Foundation-Tier also includes generic concepts that are used for spatial and non-spatial purposes which are useful in the process of creating address ontologies and making them interoperable across domains (see table 5). The generic concepts include the spatial and non-spatial type concepts which determine the behaviour of an address concept or its instances. Spatial types, such as point and polygon, are defined by W3C standard ontology at http://www.w3.org/2003/01/geo/wgs84_pos. Other spatial attributes, such as latitude, longitude, and altitude are also defined in the same ontology.

The non-spatial type concepts are also part of the foundation ontology and the types are used in the lower level ontologies. Since RDF/XML is the W3C recommended format for OWL, using XML datatypes to represent the non-spatial types was the best option.

Vocabulary For	**Namespace URI**
Dublin Core (Metadata)	http://purl.org/dc/elements/1.1/
Geographic Terms	http://www.w3.org/2003/01/geo/wgs84_pos#
OWL language concepts	http://www.w3.org/2002/07/owl#
RDF language concepts	http://www.w3.org/1999/02/22-rdf-syntax-ns#
RDFS language concepts	http://www.w3.org/2000/01/rdf-schema#
Non-spatial (xml) data types	"http://www.w3.org/2001/XMLSchema#"

Table 5: The Foundation Ontologies Used for Address Interoperability

The existence of Foundation-Tier allows the separation of "address" specific concepts from the generic concepts. The Foundation-Tier that was used in the research can easily be extended with new ontologies at any time, even after the lower tier ontologies are implemented. Concepts of the Foundation-Tier are simply imported and used by the lower tier ontologies. The six foundation ontologies listed in table 5 are necessary for the development of the Base Tier ontology for address interoperability.

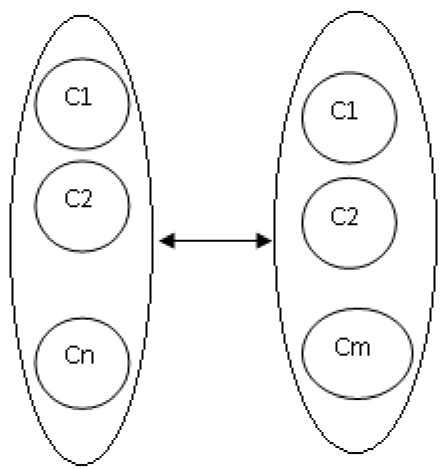
5.5 Base Tier

The Base-Tier is where all the foundation concepts and selected address-related concepts (from different address domains) are put together, and specific relations are established amongst them, to define the reference ontology for addresses. This tier contains an address reference ontology that establishes the basis for the interoperability of various address data across different address domains. The address reference ontology that was developed (to facilitate interoperability) is called "Base Address Ontology" (BAO) (see section 4.3).

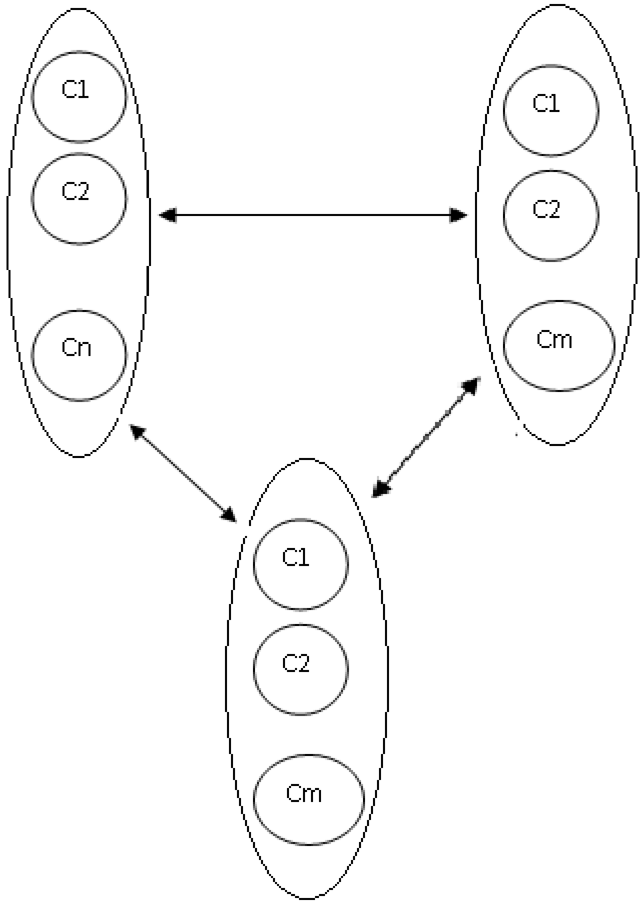
Facilitating interoperability across two address domains can be achieved by using direct mapping of concepts across the two domains and setting some cross domain restrictions, equivalences, and axioms. As the number of domains (that need to be interoperable) increase, however, this technique of direct mapping of concepts across domains becomes very inefficient and difficult to maintain.

For example, consider two address domains *D1* and *D2* (figure 50 a) that are established, based on two separate address standards. These two domains have similar (but not exactly the same) address concepts with different terminologies. The corresponding concepts are, thus, mapped across domains, and restrictions are established to define the way a concept is interpreted in a specific domain. If a third domain comes into play (as in figure 50 b) then the number of concept mapping has to increase by more than double.

There are, however, many address domains, and the interoperability is intended to be across all the domains. As is true in many cases, the number of address domains that needs to be interoperable might be more than two or three. Every new n^{th} domain that is added to the web of interoperable address domains will require n-1 more sets of mapping of concepts.



a. Direct mapping between two domains



b. Direct mapping amongst three domains

Figure 50: Direct Mapping of Concepts amongst Address Domains

It is difficult to maintain interoperability using the technique of the direct mapping of concepts because of the large amount of mapping required. Hence, it is important to reduce, if not eliminate, the need for repeated mapping of concepts amongst address domains. One way to reduce the mapping of concepts (without reducing interoperability) is to use reference ontology.

Reference ontology takes the most common concepts from the domain ontologies and defines them in a way that is usable by all domain ontologies. Reference ontology does not always eliminate the need for the direct mapping of concepts and restrictions, but it reduces it significantly (see figure 51). Reference ontology is also useful in developing ontologies for new address domains.

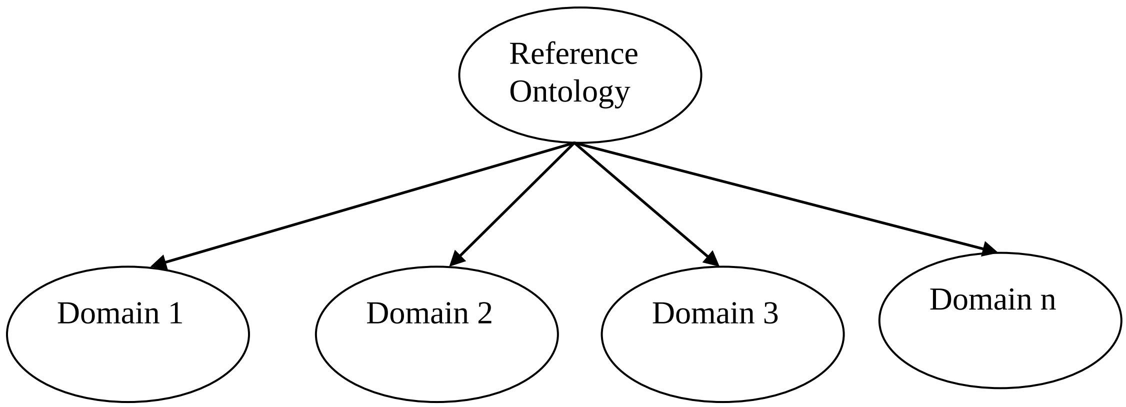


Figure 51: Mapping Concepts Across Domains Using Reference Ontology

The BAO is reference ontology for address domains, and, as such, its main goal is to:

- facilitate interoperability across address domains;
- re-use knowledge in efficient and standardized way; and
- solve problems that arise owing to heterogeneity.

As reference ontology, BAO has to describe the concepts that are common across all address domains and the concepts that serve as the basis for the establishment of all address domains. The concepts of the BAO were selected carefully not to be too specific or too generic. Finding the right balance for the contents (concepts, properties, and regulations) of the BAO as reference ontology was a challenging task. It is a task that needs continuous and very carefully designed modifications. As reference ontology, however, it should be much more stable than the domain ontologies, because any change to the reference ontology can affect the subsequent domains.

The base tier has one (reference) ontology. At the early stages of the research, owing to the diversity of the address domains and their various purposes, using multiple (two to four) ontologies was considered. The reason this idea of multiple reference ontologies was considered was to increase the number of inferences. The use of multiple address reference ontologies, however, leads to the emergence of separate poles of address domain ontologies that are distant from one another. Address data interoperability across domains in different poles would be very challenging, and the task becomes more complicated as more domains are involved in the interoperability process. Using one reference ontology avoids the polarization of the domain ontologies, and, even though it may have some drawbacks in the number of inferences, the drawbacks can be compensated for by using inter-domain restrictions and constraints when running the inference engines.

5.6 Domain Tier

The top two tiers of the ontology architecture have generic address concepts that are applicable to most, if not all, of the address domains. The top (foundation) tier contains multiple ontologies that are required to build a proper reference ontology for addresses. The second tier holds the reference ontology (BAO) for address domains. The third tier of the ontology architecture is where the reference ontology is extended with domain specific concepts that are used to describe address concepts with the exact semantics of a domain. Having a separate tier for domain ontology helps to develop and identify as many domain specific ontologies as possible from the reference ontology (BAO).

In order for a domain ontology to be used in the interoperability of address data, it has to extend the address reference ontology (BAO) with domain specific concepts, properties, axioms, and constraints. The use of common address reference ontology (BAO) allows the domain ontology developers to be given an easy start to an interoperable system. The BAO has a good balance between representation for maximum interoperability and extendibility.

The domains refer to the different address standards (both *de facto* and *de jure*) that are used to assign and collect address data. Different address terminologies and conceptual models are used owing to the variation in the purposes, locations, historical background, and other factors of the standards. Every standard has a unique set of vocabularies to express or represent addresses. Since the approach used in the research depends on capturing the semantics of the address data in order to facilitate interoperability, it was necessary to develop a separate ontology for each domain. The vocabularies and conceptual models of the address standard are usually designed with the help of systems experts in the field (for which the address standard is used). The vocabularies are used in the development of domain ontologies, and understanding them requires the involvement of the system experts. The system proposed to facilitate the interoperability of addresses depends on capturing the correct semantics of the address concepts and address data within a specific domain. The process of developing domain ontology, however, has three main challenges:

- the risk of software engineers misinterpreting address models;
- developing an intermediate model that takes the reference (BAO) model in to consideration; and

- system experts who do not understand OWL.

Software engineers need to understand and interpret the conceptual model of the address domain correctly before they start to design the domain ontology. Many conceptual models are inherently ambiguous owing to poor design or shortfalls of the modeling system used. For example, some conceptual models (like UPU-S42) use very ambiguous tabular models. Ambiguity in interpreting conceptual models could cause the total failure of semantic ontology systems because they depend on the correct meaning of the vocabularies. The best way to avoid such ambiguity is to work hand in hand with system experts of the domain who can interpret the conceptual model correctly.

In the process of developing the sample domain ontologies for the research, domain experts of various national and international address standards were involved. The participation of the SANS-1883, INSPIRE, and UPU system experts was crucial to the improvement of the quality of the domain ontologies designed for the research purposes.

The conceptual address models of the domains were designed without consideration of the reference ontology. In order for the domain ontologies to be interoperable, they need to be redesigned with the consideration of the reference ontology but without changing the semantics of the original conceptual models. Profile models for each domain should, thus, be designed to extend the conceptual model of the reference ontology. The profile models have the same semantics as the original conceptual models, even though it is possible to have syntactic differences. The reference ontology (BAO) was designed to be generic enough to be extended by many (if not all) the domain ontologies without losing their purpose or meaning.

The profile models are to be used only for the purpose of interoperability, and they do not replace the original conceptual models of the domains in any way.

Designing a profile model for a domain requires the participation of both domain experts and software engineers. This partnership allows for the avoidance of any semantic discrepancy between the original and the profile conceptual models. When designing a profile model for a domain:

- consider both the domain conceptual model and the address reference model; and
- avoid semantic ambiguity in the profile model.

As stated above, many domain conceptual models are ambiguous, but the profile model has to be unambiguous. One of the causes for ambiguity is the choice of a modeling system. UML, which is a less ambiguous modeling system that can easily be translated in to OWL, is, thus, used to represent the profile models. UML as a modeling system is easy to learn and understand because it uses few (usually easily intuitive) rules and symbols. Hence the system experts can quickly adopt it, and the software engineering can easily convert it to OWL.

Once the UML profile model for a domain is developed, the software engineers develop the domain ontology in OWL. OWL is not convenient for human communication (Gasevic et al. 2004), because it was designed to enable machines rather than humans to capture semantics. Unlike UML, OWL has a steep learning curve (and it is not intuitive at all) that makes it difficult for system experts to understand. Once the domain ontology is developed, therefore, the software engineers need to use different tools and techniques to convey the content and the results (of inferences, rules, and relationships) of the ontology to the system experts.

In this research, with the help of the TopBraid software, UML-like models and the graphical representation of relationships between extensions (instances) were used as the medium of communication with system experts and to explain the ontologies in this document. The main drawback of using RDF/XML formats of OWL ontologies is that they are not easily understood by humans.

For domain ontology to become fully applicable, it needs to be accepted by the custodians of the domain standard. The domain ontologies developed and used in this research are merely prototypes which have not gone through the long process of being accepted by the custodians. They were limited to be prototypes because of time limitations. They have, however, been used to demonstrate the role of domain ontologies in the process of address data interoperability.

5.7 Application Ontology

The last tier in the ontology hierarchy is the application tier, where all the application, task, or purpose-specific ontologies are located. The top three tiers of the hierarchy are designed to capture all the generic concepts (with decreasing granularity from top to bottom) of addresses.

The ontologies in the application tier are narrowly defined for a unique task or purpose (usually mostly with class extensions).

The application tier is necessary because, without it, the class extensions would have to be included in the domain tier, and that would limit the use of domain tier ontology to one task. The domain ontologies are designed to capture address concepts for a specific domain. These (domain) ontologies should, however, be used for various tasks. For example, the INPIRE domain ontology could be used for the task of local administration in different European states. Every European country would use its own address data for local administrative purposes, and the address data from other states might not be useful for local administration. The application tier helps, therefore, to separate the class extensions and task specific concepts from the domain specific concepts.

The addresses used to demonstrate the application tier ontologies are not real addresses. Changes have been made to the names and numbers of the addresses; the format of the address data was, however, not changed. The names used (of individuals or companies) are not real either. This was done for ethical reasons and to prevent any unintended breach of privacy.

Four application ontologies were developed to investigate different scenarios where application ontologies could be used. In OWL, instance data are represented as class extensions of ontology. As the class extensions of the address concepts (address data) are added at the application tier, the interoperability (across multiple ontologies) actually takes place at this tier.

Address application ontologies that extend the address reference ontology (BAO) could be related to one another in either one of two ways. The relationship between the application ontologies were either:

- specialization in sub-domains, in which case they have mutually exclusive tasks
- provide different perspectives of a domain

This classification seems to cover all the possible scenarios to which two or more address application ontologies could be related. The specializing in sub-domains does not necessarily mean the application ontologies specialize in sub-domains of the same domain.

Application (UKdata) ontology was developed to test the first scenario. The UKdata ontology specializes in the UK sub-domain of the INSPIRE (European) address domain. The UKdata ontology extends the INSPIRE domain ontology with UK specific address instances and concepts. Specialization in the sub-domain also helps to avoid the centralization of ontology and allows the national address data custodians (authorities) to extend their own sub-domain ontology. For example, in the case of INSPIRE (European) address domain the national address data custodians can extend the INSPIRE ontology with their national address instances and specializations. Separation of sub-domains helps to avoid the accidental assignment of relationship between instances with similar identifiers but not the same addressable objects. For example, an address component of a UK address could be related to an addressable object in France by mistake if both the address objects have similar identifiers and are managed within the same ontology. Such a wrong relationship could cause wrong inferences that could threaten the integrity of the ontology.

Sections 4.5.1 and 4.5.2 demonstrated application ontologies that specialize in sub-domains of local administrative (INSPIRE) and postal services (UPU) domain ontologies respectively. These two application ontologies were useful in understanding the process of adding class extensions (data instances), SPARQL querying, and setting restrictions on ontology properties and classes. Writing OWL ontologies without the use of editing tools can be error prone. Thus TopBraid, an ontology development tool, was used to develop the ontologies, write SPARQL queries, make inferences, and generate graphical models to show relationship among the ontology concepts.

Ontologies that specialize in sub-domains could still be used to extract information that spans across the sub-domains. For example, the ontologies that specialize in the European state sub-domains of the INSPIRE can be used to extract information such as the number of colleges, museums, or hospitals across states.

Sections 4.5.3 and 4.5.4 demonstrate interoperability across application ontologies that provide different perspectives of a domain. Users of such application ontologies benefit from interoperability because they get information of a different perspective to their domain. This extends the knowledge base and can be used to make useful inferences. In both sections (UKdata+UPUdata ontology and UKpostal ontology) two perspectives of a domain were used to demonstrate interoperability and its benefits. It is possible, however, to get an interoperable system with more than two perspectives of a domain.

Two scenarios of interoperable systems with different perspectives of an address domain were experimented with. The first one was UKdata+UPUdata ontology where two application ontologies with different perspectives of the UK address domain, each one with its own set of address data, were used. The second scenario (UKpostal ontology) demonstrates interoperability between two ontologies of a shared domain (again the UK address domain) where only one of the application ontologies has class extensions. These two scenarios were considered because different methods of facilitating interoperability are needed in each of them.

Section 4.5.3 considered the first scenario where multiple (2 or more) application ontologies of a common address domain but different application domains (different perspectives of addresses) are used to extract, share, exchange, and make inferences on address information.

An application ontology that extends the ontologies used in the interoperable system was designed. Figure 52 shows a mind map of how the ontologies relate to one another. In the figure, two application ontologies of different Application Address Domains (AAD1 and AAD2) that share a common Address Domain (K) are used to create an interoperable system via a third application ontology with Application Address Domain (AAD1+AAD2) which is a combination of the component application address domains.

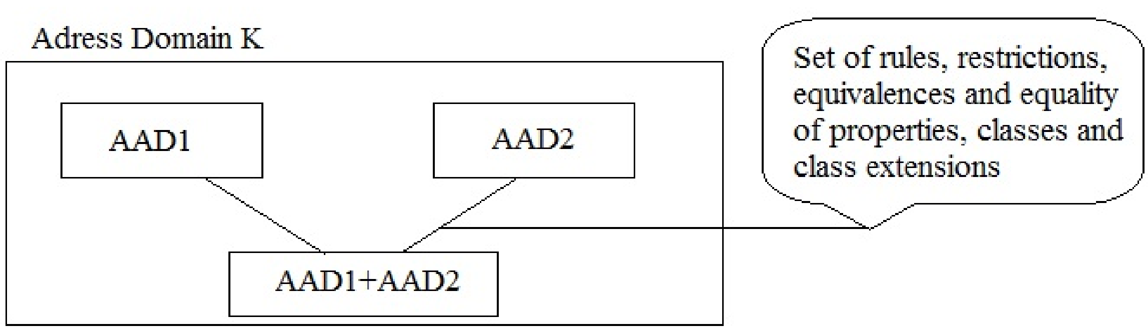


Figure 52: Scenario 1 for Application Domain interoperability

The new application ontology (AAD1+AAD2 in figure 52) extends the component ontologies with a set of rules, restrictions, property equivalences, and equality of classes and class extensions. SPARQL queries, and multi-level inferences are, furthermore, used to enable machines to gain more knowledge and make better decisions. The knowledge generated by the process or the rules used for the process of interoperability of the component ontologies does, however, not affect the content of the component ontologies.

The custodians of the component application ontologies could be different. It is, thus, crucial that the new application ontology exists to avoid tampering with the original ontologies. The set of rules, restrictions, equivalences, and equalities that are used to create interoperability must be assigned carefully. Any wrong equivalence or equality between properties, classes, or even class extensions of the component ontologies could cause wrong inferences. The information generated by such wrong inferences should be isolated and should not affect the integrity of the component ontologies. Hence, a separate application ontology that extends the component ontologies and uses a different (its own) namespace to represent the new classes, properties, and class extensions is required. The new application ontology does not duplicate the concepts of the component ontologies, i.e. there is one URI for each concept of the component ontologies. Any change that is made to the definition of concepts of the component ontologies is reflected in the interoperable system via the new (third) application ontology, but the reverse is not true. For example, changes made to concepts of AAD1 (in figure 52) affect ADD1+ADD2; the changes made in ADD1+ADD2 do not, however, affect AAD1 or AAD2.

In the research experiment, UKdata and UPUdata ontologies were the two application ontologies with a shared (common) address domain, which is the UK address domain. UKdata ontology has the UK local government (administrative) application address domain, while the UPUdata ontology has the UK postal services application address domain. The UKdata+UPUdata application ontology was designed and used to facilitate interoperability between the two component (UKdata and UPUdata) ontologies.

The UKdata+UPUdata ontology extends the application ontologies discussed in section 4.5.1 (UKdata ontology) and section 4.5.2 (UPUdata ontology). It, thus, demonstrates the first scenario where interoperability between two address systems that have class extensions of address data from a common domain. The purpose of interoperability between such two address systems is to share and exchange data and to complement the information of each other. The semantics of the concepts (and their class extensions) of each address system were first captured in separate application ontologies. Relationships were established between the ontologies to create an interoperable environment between the application domains.

It is possible to avoid use of the new (third) application ontology (UKdata+UPUdata ontology) in the interoperability process. To do that, the set of rules, restrictions, equivalences, and equalities of properties, classes, and class extensions that are required for the interoperability need to be included in either or both of the component ontologies. In such cases, even a minor error in setting property equivalence or equality of classes could compromise the integrity of the entire ontology.

When one of the component ontologies does not, however, have any class extensions, and if the purpose of the interoperability is to provide this component ontology with class extensions, it is then desirable to extend it with the set of rules, restrictions, equalities, and equivalences of concepts from the other component ontology. Running an inference engine on the component application ontology (where the OWL restrictions have been established) would facilitate the sharing of class extensions. Figure 53 shows the second scenario of the experiment where two application ontologies within a common address domain but with different address application domains are used for interoperability. This scenario considers the situation where an application ontology that has no address data (instances) gains information from existing application ontology.

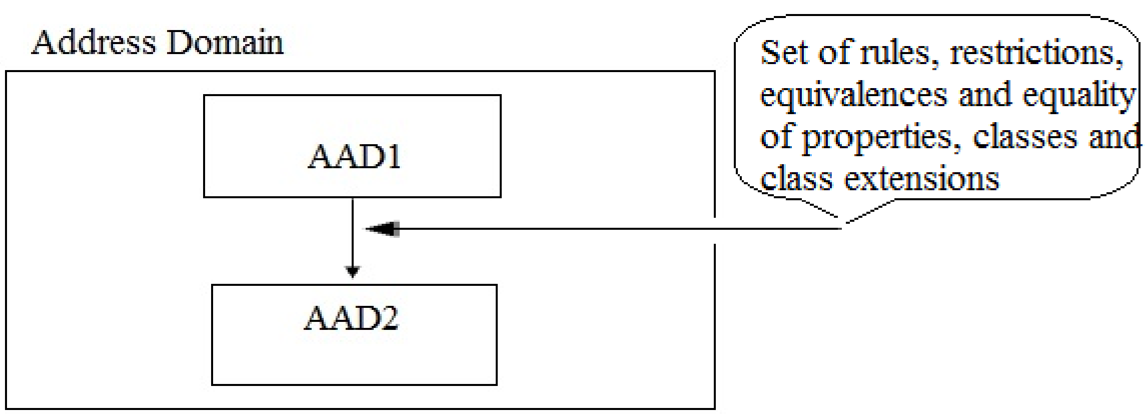


Figure 53: Scenario 2 for application domain interoperability

When two or more application ontologies share a common address domain, they can share and exchange address data directly without the need of a third (bridge) application ontology. This kind of procedure affects the integrity of one or both of the component ontologies. For example, figure 53 shows that application ontology AAD1 has address data and application ontology AAD2 needs to incorporate these address data. AAD1 and AAD2 have different application address domains (different purposes) but they share common address domain. When the aim of the interoperability is to provide one application ontology with address data (instances) from another application ontology within the same address domain, then address data from the latter needs to duplicated in the former ontology. Since both the application ontologies do not, however, have the same application domain (purpose), some of the attributes of the original address data might not be required and some new attributes might need to be added.

Section 4.5.4 demonstrated this procedure of address interoperability by employing a scenario where a new postal service office needed address data for a London suburb and used address data from the local administration authorities. Both the application ontologies had the same address domain (London, UK) but different application domains (one for postal services the other for local administration). UKdata ontology was the local government application ontology and UKpostal ontology was extended with the class extensions (address data instances) from UKdata ontology. UKpostal ontology was also extended with concepts that are specific to its application domain.

Different scenarios were covered in the research but there could be more scenarios that could be experimented with in the future. The use of ontologies for interoperability under different situations was tested. The results were positive, but more cases with larger address data need to be used to test the efficiency of the system.

Chapter 6

6 Conclusion

6.1 Introduction

Address data has been one of the most important fields of information in the day-to-day activities of individuals, private and public companies, government (local) administration authorities, and academic and scientific research institutions amongst others. The need to share, exchange, and process address data from a multitude of sources has been evident for long time. The task, however, of creating a cross domain interoperable address data system has not been easy. This research study uses semantic ontologies to deal with the challenge of address data interoperability across domains.

The previous five chapters have given the background information, problem statement, research objectives, literature review of existing technologies and knowledge base, and also a discussion of the experiment approach, tools, and results.

This final chapter contains sections dealing with:

- A brief summary of findings;
- Research conclusion;
- Contribution to science and technology; and
- Suggestions for further research.

6.2 Summary of Findings

The literature review and the research on the structure and content of address data from different countries, which are used for various purposes in different address domains, show that there are significant differences amongst them. The structure of address data depends on the standard (that specifies the conceptual address models) used to assign and collect addresses. Even though standards are used to create unified forms for address assignment and sharing, there are too many of them in different places used for different purposes. It is, however, still very important to address the issue of data sharing and processing across domains. The standards have technical, historical, and cultural aspects that are not easy to change at will at any time. The idea of developing one global address standard that replaces all the existing address standards was, thus, not even practical. Hence there was a need to find a way to create address data interoperability across domains, without disturbing the existing addressing systems, and in a way that the address data is not only shared but also processed across domains.

The challenges of address data sharing are not limited to the structure of address data; the digital storage format of the address data also plays a role in complicating the sharing process. Address data are stored in different formats, including plain text, xml, RDB, and spreadsheets amongst other formats. Even if address data are assigned and gathered based on the same standard, the format (syntax) of the storage could make it difficult to share or process. The semantic approach, thus, which takes only the semantics of the data regardless of the syntax, was taken as an option to facilitate interoperability.

The literature review and research on the use of semantic ontology for interoperability shows that there have been several successful attempts to facilitate interoperability across domains within a field of study. No such attempts or research studies on address domains have, however, been found. Owing to the technical, cultural, socio-political, and historical aspects addresses have unique importance, formats, and challenges that are not fully experienced in other fields. There was, thus, a need to come up with a novel ontology architecture (that is based on lessons learned from other ontology architectures) that would facilitate seamless interoperability of address data across multiple domains.

One of the major goals of the research was to find an optimal knowledge (address) representation model for the semantic approach that would be used to facilitate interoperability. A four-tier hierarchy of ontologies was identified as the optimal model for facilitating the interoperability process. The four-tier ontology architecture was used to facilitate interoperability by using multiple ontologies working in coherence with hierarchical dependencies that allow the re-use of ontologies and the separation of tasks.

At the top of the hierarchy are the foundation ontologies where all the ontology language concepts and axioms are defined, and where the spatial and non-spatial datatypes are defined. For consistency, correctness of resources (IRIs), and efficiency of ontologies the Foundation-Tier concepts should not be redefined. This research study has identified a number of foundation ontologies and vocabularies (see Table 5) that are essential for the development of optimal address reference ontology.

Another major goal of the research was the development of optimal address reference ontology. The second tier of the hierarchy was the Base-Tier where the address reference ontology BAO was the single ontology of the tier. The BAO is address reference ontology that was developed as part of the research to facilitate interoperability. The conceptual model for BAO was developed after carefully investigating many address domains from across the globe with some side inputs from domain experts. The concepts and axioms of the BAO were carefully defined so as not to be too generic or too specific, in order to be readily extendible by all address domains.

Ontologies in the bottom two tiers of the hierarchy extend the address reference ontology (BAO) with address domain and application domain concepts, instances, and axioms. Some prototype domain and application ontologies were developed to test the correctness of the hypothesis of the research. The UK local administrative address domain and the UK postal address domain were used to demonstrate the use of address domain specific ontologies and the benefits of separating address domain ontologies from application domain ontologies.

The last tier of the hierarchy was the Application-Tier where application specific ontologies extend one or more domain ontologies and possibly other application ontologies to create an interoperable system that gains from multiple ontologies. Some application ontologies were developed as prototypes to demonstrate different scenarios where the hypothesis can and should be tested.

The ultimate goal of the experiments was to create an interoperable system for across domain address data sharing, exchange, and processing using a semantic approach. Hence, the results of the experiment were two interoperable address data systems. These interoperable environments were created using the UKdata+UPUdata ontology and UKpostal ontology, which illustrate different use cases of ontologies that facilitate interoperability. Figures 43 and 47 showed the hierarchy of ontologies used and the address domains used in the interoperability process. Ontology reasoners, inference, and SPARQL query tools were used to share, exchange, and process address data across address domains. Ontology inferences were done to exchange address data attributes between the UK administrative address data and UK postal service address data systems in the UKdata+UPUdata ontology. SPARQL queries were, furthermore, run to extract and process information from different perspectives of an address domain and from combined perspectives of two (UK administrative and UK postal) address domains. The second interoperable system (UKpostal ontology of section 4.5.4) illustrated the use of ontology inference tools to share address data between two address data systems that provide different perspectives of a domain.

6.3 Conclusion of the Research

The objective of this research was to investigate the use of semantic ontologies to facilitate address data interoperability across address or application domains. The research problem was not whether semantic ontologies can be used to facilitate interoperability, because it has been numerously demonstrated (in other fields of studies) that it can be used. The research problem was rather, considering the special importance and characteristics of addresses:

- How could semantic ontologies be used to facilitate interoperability of address data across domains?
- What other options are there to facilitate interoperability?
- How effective would a semantic approach be?
- What are the main challenges of using semantic ontologies approach?
- What are the advantages of the approach?

After a thorough literature review and some experimental tests of feasibility, an approach with multi-tier ontology architecture was evaluated to be the feasible one. Four-tier ontology architecture was evaluated as the optimal, efficient, and effective way to facilitate interoperability with maximum interoperability without causing any constraints on the distributed nature of address data ownership and administration.

Non-semantic approaches for address data interoperability were investigated. Owing, however, to the syntax and conceptual-model differences of the address domains it was easy to create an interoperable system for multiple domains. Compared to what occurs in the semantic ontology approach, in the non-semantic approach, as the number of domains in the interoperable system increases, the difficulty of the system increases significantly.

The four-tier ontology architecture allows for the separation of the generic semantic ontology axioms, address reference ontology axioms, address domain ontology axioms, and address application domain axioms. The separation of concepts and axioms into multiple (hierarchical) ontologies allows the facilitation of the development of ontologies in an efficient, more standardized, and accurate way.

Well-designed address reference ontology optimizes the address data interoperability across domains. The quality of the reference ontology also plays a role in the amount and quality of knowledge generated with the help of reasoning, inference, and query tools. Hence, the address reference ontology (BAO) was developed after reviewing several address standards from across the world. The concepts and axioms of this reference ontology were carefully selected not to be too generic or too specific to any set of address domains. The effectiveness of an interoperable system is dependent on the quality of the address reference ontology BAO.

The separation of the address domain ontologies from the application domain ontologies was found to be important because address domain ontologies can be used for many applications which do not necessarily share common instances of addresses. Hence, the four-tier approach was preferred to the three-tier approach that mixed both address and application domains.

The major challenges of the semantic ontology approach to address data interoperability are:

- a lack of an internationally standardized conceptual model for the address reference ontology;
- an ambiguity of concepts in the conceptual models of the address standards;
- the inherent difficulty of semantic ontology languages (such as OWL) to demonstrate (present) to domain experts;
- a lack of skilled people to add semantics to the Web (develop domain and application ontologies); and
- there will be too little address domain and application ontologies on the Semantic Web initially to achieve the benefits of interoperability.

The major advantages of the semantic ontology approach to address data interoperability are its ability to:

- integrate different existing address data formats and systems;
- extend an interoperable system without significant effort required when a new address domain is added to the interoperable system;
- use machines to reason and make inferences that would have been difficult to do manually on existing address data;
- use distributed sources of address data to achieve interoperability without losing any control from the data owners;
- changes (corrections or addition of new address data or model) made by the custodians of an address domain could automatically be reflected in other ontologies in the interoperable system; and
- be accessed from any part of the world because it uses the Semantic Web infrastructure.

6.4 Contribution

The findings of the research have made some contributions to science and technology by:

- providing an example of a semantic approach to facilitate address data interoperability across address domains using ontologies;
- identifying the challenges and benefits the approach used;
- developing an address reference ontology that has not been defined on the Semantic Web and the Liked Open Data (LOD) cloud in particular; and
- laying the ground work and suggestions for further research work.

6.5 Suggestions for Further Research

The research work in this dissertation is the first step in the investigation of the use of the Semantic approach in facilitating address data interoperability. The research had several findings that answered the research questions, and it has also opened the door for more new research questions that need to be answered. Some of the research questions that require further investigation are:

- What are the challenges of the application integration aspect of the approach?
- How efficient would the approach be if large datasets from distributed sources are used?
- What would be the challenges and benefits of modifying the address reference ontology (BAO) with the conceptual model from ISO19160-1 (once it is published)?

- How can the Semantic Web be made a useful component of Spatial Data Infrastructures (SDIs)?

Semantic Web programming has two major components, knowledge representation and application integration. The research of this dissertation was mainly concerned with the former component, while the latter component of Semantic Web programming (i.e. application integration) requires further investigation. The challenges of integrating different address data repository systems using semantic ontologies needs to be investigated.

Small and carefully selected datasets were used during the research to demonstrate the proof of the concept. The semantic approach, however, needs further investigation for its efficiency when large datasets from distributed sources are used.

The address reference ontology (BAO) that was developed as part of the research used a conceptual model that was inspired by conceptual models from the early drafts of ISO19160-1. Since ISO19160-1 will be an international address standard that has involved the participation of representatives from many countries and organizations once it is published, its conceptual model can be used to provide a vocabulary to modify the address reference ontology. The modified address reference ontology's benefits and challenges when it is used in address and application domain need to be investigated and evaluated. This will require repeating the experiments done in this research with similar procedures.

The role of the Semantic Web in different fields and aspects of life have been investigated. The role of the Semantic Web, its benefits and challenges, not just for address data but also for all kinds of spatial data sharing and processing, needs to be investigated.

Bibliography

Allemang, D. & Hendler, J., 2011. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL* Seconnd Ed., Elsevier.

Andersson, B. et al., 2006. Conceptual Modeling - ER 2006. In D. W. Embley, A. Olivé, & S. Ram, eds. *25th International Conference on Conceptual Modeling*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 482–496. Available at: http://www.springerlink.com/index/10.1007/11901181 [Accessed October 10, 2014].

Angrosh, M.A., Cranefield, S. & Stanger, N., 2014. Contextual information retrieval in research articles: Semantic publishing tools for the research community. *Semantic Web*, 5(4), pp.261–293.

Antoniou, G. & Van Harmelen, F., 2004. *Handbook on Ontologies* S. Staab & R. Studer, eds., Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: http://link.springer.com/10.1007/978-3-540-24750-0 [Accessed October 10, 2014].

Bechhofer, S. et al., 2003. OWL Web Ontology Language Reference. *W3C.org*. Available at: http://www.w3.org/TR/2003/PR-owl-ref-20031215/ [Accessed October 9, 2014].

Blind, K., 2004. The economics of standards. *Theory, evidence, policy. Cheltenham*.

Bock, C. et al., 2012. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). Available at: http://www.w3.org/TR/owl2-syntax/ [Accessed October 10, 2014].

Bratt, S., 2008. Semantic Web, and Other Technologies to Watch: January 2007 (37). Available at: http://www.w3.org/2008/Talks/1009-bratt-W3C-SemTech/#(37) [Accessed October 17, 2014].

Brickley, D., 2004. W3C Semantic Web Interest Group: Basic Geo (WGS84 lat/long) Vocabulary. Available at: http://www.w3.org/2003/01/geo/ [Accessed October 18, 2014].

Brickley, D. & Guha, R.V., 2014. RDF Schema 1.1. *W3C.org*. Available at: http://www.w3.org/TR/rdf-schema/ [Accessed October 9, 2014].

Cargill, C.F., 1995. A five-segment model for standardization. In B. Kahin & J. Abbate, eds. *Standards policy for information infrastructure*. Harvard Information Infrastructure Project, pp. 79–99.

Cargill, C.F., 1989. *Information technology standardization*, Digital Press.

Chen, D., Doumeingts, G. & Vernadat, F., 2008. Architectures for enterprise integration and interoperability: Past, present and future. In *Computers in Industry*. pp. 647–659.

Coetzee, S., 2008a. Address data exchange in South Africa. In *ISO Workshop on address Standards - Considering the issues related to an international address standard*. Copenhagen, Denmark: ISO Workshop on address standards, pp. 38–49.

Coetzee, S., 2010. Address for the World. In *Geospatial World*. Noida, India, pp. 48–50.

Coetzee, S., 2008b. How standardized addresses can spatially enable government A citizen at each address …. In *3rd Annual Spatially Enabled Government Conference*. Midrand, South Africa.

Coetzee, S. et al., 2010. What address standards tell us about addresses. In *ISOFocus+*. Geneva, Switzerland, pp. 1–25.

Cole, J., 2004. Raising the bar. *Computer*, 37(9), pp.112–113.

Communication Programme International Bureau Universal Postal Union, 2012. *Annual Report Setting the stage*, Berne, Switzerland. Available at: http://news.upu.int/fileadmin/user_upload/PDF/Reports/annual_report_2012.pdf.

Cyganiak, R. & Jentzsch, A., 2014. The Linking Open Data cloud diagram. Available at: http://lod-cloud.net/ [Accessed October 18, 2014].

Cyganiak, R., Wood, D. & Lanthaler, M., 2014. RDF 1.1 Concepts and Abstract Syntax. *W3C.org*. Available at: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ [Accessed October 9, 2014].

Damsgaard, J. & Truex, D., 2000. Binary trading relations and the limits of EDI standards: the Procrustean bed of standards. *European Journal of Information Systems*, 9(3), pp.173–188.

Das, S., Sundara, S. & Cyganiak, R., 2012. R2RML: RDB to RDF Mapping Language. *W3C Recommendation*. Available at: http://www.w3.org/TR/r2rml/ [Accessed February 5, 2015].

Delbru, R. et al., 2010. Hierarchical link analysis for ranking web data. In *The Semantic Web: Research and Applications*. Springer Berlin Heidelberg, pp. 225–239.

Dolin, R.H. & Alschuler, L., 2011. Approaching semantic interoperability in Health Level Seven. *Journal of the American Medical Informatics Association*, 18(1), pp.99–103.

Egyedi, T.M., 2009. Between supply and demand: Coping with the impact of standards change. *ICT Standardization for E-Business Sectors: Integrating Supply and Demand Factors, Hershey, PA: IGI*.

ESRI, 2012. Reveal More Value in Your Data with Location Analytics - reveal-more-value.pdf. Available at: http://www.esri.com/library/whitepapers/pdfs/reveal-more-value.pdf [Accessed October 9, 2014].

Florescu, D. & Kossmann, D., 1999. Storing and Querying XML Data using an RDMBS. *IEEE Data Engineering Bulletin, Special Issue on 1060*, 22(3). Available at: http://dsl.serc.iisc.ernet.in/~course/TIDS/papers/daniela.pdf.

Folmer, E. & Verhoosel, J., 2011. *State of the Art on Semantic IS Standardization, Interoperability & Quality*, UT, CTIT, TNO en NOiV. Available at: http://doc.utwente.nl/76291/.

Gangemi, A., 2005. Ontology Design Patterns for Semantic Web Content Y. Gil et al., eds. *History*, 3729(4), pp.262–276. Available at: http://www.springerlink.com/index/f513071t4477h2w2.pdf.

Gasevic, D. et al., 2004. Converting UML to OWL ontologies. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters - WWW Alt. '04*. New York, New York, USA: ACM Press, pp. 488–489. Available at: http://portal.acm.org/citation.cfm?doid=1013367.1013539 [Accessed October 9, 2014].

Görlitz, O. & Staab, S., 2011. Federated data management and query optimization for linked open data. In *New Directions in Web Data Management 1*. Springer, pp. 109–137.

Gruber, T.R., 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43(5-6), pp.907–928.

Hammond, W.E., 2008. eHealth interoperability. *Studies in health technology and informatics*, 134, pp.245–253.

Hanseth, O., Monteiro, E. & Hatling, M., 1996. Developing Information Infrastructure: The Tension Between Standardization and Flexibility. *Science, Technology & Human Values*, 21(4), pp.407–426.

Hawke, S. et al., 2013. W3C Semantic Web Activity Homepage. Available at: http://www.w3.org/2001/sw/ [Accessed October 17, 2014].

Heath, T. & Bizer, C., 2011. Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1), pp.1–136.

Hebeler, J. et al., 2009. *Semantic Web Programming* M. Dean & M. Smith, eds., Indianapolis, Indiana: John Wiley & Sons.

Hesser, W., Feilzer, A. & De Vries, H.J., 2006. *Standardisation in companies and markets* W. Hesser, ed., Hamburg, Germany: Helmut Schmidt University.

Hitzler, P. et al., 2012. OWL 2 Web Ontology Language Primer (Second Edition). *W3C.org*. Available at: http://www.w3.org/TR/2012/REC-owl2-primer-20121211/ [Accessed October 9, 2014].

Huang, M., 2010. Use geographic ontology and web mining to enhance service quality of web map. In *2010 3rd International Conference on Computer Science and Information Technology*. IEEE, pp. 625–629. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5565110 [Accessed October 9, 2014].

Ilayperuma, T., 2007. *Reference Ontology for Business Models Towards Interoperability between Business Modelling*. Stockholm University.

INSPIRE/TWG/ Addresses, 2009. *D2 . 8 . I . 5 INSPIRE Data Specification on Addresses – Guidelines*, INSPIRE Thematic Working Group Addresses.

International Organisation for Standards -ISO, 2014. ISO Standards. Available at: http://www.iso.org/iso/home/standards.htm [Accessed October 10, 2014].

ISO Central Secretariat, 2014. About ISO - ISO. Available at: http://www.iso.org/iso/home/about.htm [Accessed October 17, 2014].

ISO/TC 211, 2011. *Draft Review summary of project 19160*, *Addressing*, Available at: http://www.isotc211.org/protdoc/211n3075/.

ISO/TC211, 2005. Business plan. , pp.1–10. Available at: http://isotctest.iso.org/livelink/livelink/fetch/2000/2122/687806/ISO_TC_211__Geographic_information_Geomatics_.pdf?nodeid=1002196&vernum=-2.

ISO/TC211/WG7, 2012. Draft ISO19160-1 Addressing Conceptual Model.

Kazuhiko, A., 2008. Japanese Address System. In *ISO Workshop on address Standards - Considering the issues related to an international address standard*. Copenhagen, Denmark, pp. 1–13.

Kerski, J.J., 2008. Developing Spatial Thinking Skills in Education and Society. *ESRI ArcWatch*. Available at: http://www.esri.com/news/arcwatch/0108/spatial-thinking.html [Accessed October 9, 2014].

Kosanke, K., 2006. *Interoperability of Enterprise Software and Applications* D. Konstantas et al., eds., London: Springer London. Available at: http://www.springerlink.com/index/10.1007/1-84628-152-0 [Accessed October 9, 2014].

Kraemer, B. & Chandra, S., 2014. The role of standards in economic development. In Asmsterdam, Netherlands: IEEE Standards Association. Available at: http://www.ieee.org/societies_communities/geo_activities/sections_congress/2014/sc2014_lab_the_role_of_standards_in_economic_development.pdf [Accessed October 10, 2014].

Krechmer, K., 2006. Open Standards Requirements. *International Journal of IT Standards and Standardization Research*, 4(1), pp.43–61.

Kreger, H., 2003. Fulfilling the Web services promise. *Communications of the ACM*, 46(6), p.29.

Kubicek, H. & Cimander, R., 2009. Three dimensions of organizational interoperability. ￼ Insights from recent studies for improving interoperability frame-works. *European Journal of ePractice*, 6, pp.1–12. Available at: http://www.epractice.eu/files/6.1.pdf.

Legner, C. & Lebreton, B., 2007. Preface to the focus theme section:"business interoperability"business interoperability research: present achievements and upcoming challenges. *Electronic Markets*, 17(3), pp.176–186.

Lutz, M. & Klien, E., 2006. Ontology- based retrieval of geographic information. *International Journal of Geographical Information Science*, 20(3), pp.233–260. Available at: http://www.tandfonline.com/doi/abs/10.1080/13658810500287107 [Accessed August 21, 2014].

Maedche, A. & Staab, S., 2001. Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2), pp.72–79. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=920602 [Accessed January 14, 2015].

Mahonen, P., 2000. The Standardization Process in IT-Too Slow or Too Fast? *Information Technology Standards and Standardization: A Global Perspective*, pp.35–47.

Malone, J. & Parkinson, H., 2010. Reference and Application Ontologies Ontogenesis. Available at: http://ontogenesis.knowledgeblog.org/295 [Accessed October 10, 2014].

McGuinness, D.L. & van Harmelen, F., 2004. OWL Web Ontology Language Overview. *W3C.org*. Available at: http://www.w3.org/TR/2004/REC-owl-features-20040210/ [Accessed October 9, 2014].

Menzel, C., 2003. Reference Ontologies — Application Ontologies : Either / Or or Both / And ? In *Proceedings of the KI2003 Workshop on Reference Ontologies and Application Ontologies*. pp. 1–10.

Zur Muehlen, M., Nickerson, J. V. & Swenson, K.D., 2005. Developing web services choreography standards—the case of REST vs. SOAP. *Decision Support Systems*, 40(1), pp.9–29. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167923604000612 [Accessed February 5, 2015].

Naudet, Y. et al., 2010. Towards a systemic formalisation of interoperability. *Computers in Industry*, 61(2), pp.176–185. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0166361509002073 [Accessed October 8, 2014].

Navigli, R. & Velardi, P., 2004. Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics*, 30(2), pp.151–179. Available at:

http://www.mitpressjournals.org/doi/abs/10.1162/089120104323093276 [Accessed October 9, 2014].

De Nicola, A., Missikoff, M. & Navigli, R., 2009. A software engineering approach to ontology building. *Information Systems*, 34(2), pp.258–275. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0306437908000628 [Accessed August 26, 2014].

Obitko, M., 2007. Semantic Web Architecture - Introduction to ontologies and semantic web - tutorial. Available at: http://obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html [Accessed October 17, 2014].

Ojo, A., Janowski, T. & Estevez, E., 2009. Semantic Interoperability Architecture for Electronic Government. *Electronic Government*, pp.63–72.

Pardo, C. et al., 2012. An ontology for the harmonization of multiple standards and models. *Computer Standards & Interfaces*, 34(1), pp.48–59. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0920548911000675 [Accessed January 7, 2015].

Parreiras, F.S., Staab, S. & Winter, A., 2007. TwoUse : Integrating UML Models and OWL Ontologies. *Computer*, (16), pp.3–17.

Patel-Schneider, P.F., 2012. OWL 2 Web Ontology Language New Features and Rationale (Second Edition). *W3C.org*. Available at: http://www.w3.org/TR/2012/REC-owl2-new-features-20121211/ [Accessed October 9, 2014].

Polleres, A. et al., 2010. Can we ever catch up with the Web ? *Semantic Web*, 1(1-2), pp.45–52. Available at: http://iospress.metapress.com/content/405517qkp7h38536/fulltext.pdf.

RDF Working Group, 2014. RDF - Semantic Web Standards. *W3C.org*. Available at: http://www.w3.org/RDF/ [Accessed October 9, 2014].

Reinstaller, A., 2008. Economic Analysis of Technological Standards and Standard Setting. In *Bargaining Norms Arguing Standards*. pp. 70–85.

Rosse, C. & Mejino, J.L. V, 2003. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of biomedical informatics*, 36(6), pp.478–500. Available at: http://www.ncbi.nlm.nih.gov/pubmed/14759820 [Accessed July 22, 2014].

Rukanova, B., 2005. *Business transactions and standards : towards a system of concepts and a method for early problem identification in standard implementation projects*. University of Twente. Available at: http://doc.utwente.nl/50864/.

Samuelson, P., 2006. Copyrighting standards. *Communications of the ACM*, 49(6), pp.27–31.

Sanders, T.R.B., 1972. *The Aims and Principles of Standardization-Saunders.pdf* T. Sanders, ed., International Organization for Standardization - ISO. Available at: http://www.iso.org/sites/edumaterials/trbsaunders.pdf.

Schreiber, G. & Raimond, Y., 2014. RDF 1.1 Primer. Available at: http://www.w3.org/TR/rdf11-primer/ [Accessed October 10, 2014].

Shapiro, C. & Varian, H.R., 1999. The Art of Standards Wars. *California Management Review*, 41(2), pp.8–32. Available at: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=1671231&site=ehost-live.

Sherif, M.H., 2006. Advanced Topics in Information Technology Standards and Standardization Research, Volume 1. In K. Jakobs, ed. *Information Technology Standards and Standardization Research, Hershey: Idea Group Publishing*. IGI Global, pp. 183–205. Available at: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-938-0 [Accessed February 5, 2015].

Simcoe, T., 2007. *Delay and de jure standardization: Exploring the slowdown in Internet standards development*, Cambridge University Press, Cambridge, UK.

Söderström, E., 2004. Formulating a General Standards Life Cycle. In A. Persson & J. Stirna, eds. *Advanced Information Systems Engineering*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 263–275. Available at: http://www.springerlink.com/index/10.1007/b98058 [Accessed February 5, 2015].

South African Bureau of Standards, 2009. SANS 1883-1 Geographic Information – Addresses, Part 1: Data format of addresses.

Spivak, S.M. & Brenner, F.C., 2001. *Standardization essentials: Principles and practice*, New York, New York, USA: CRC Press.

Staab, S. & Studer, R., 2009. *Handbook on Ontologies* S. Staab & R. Studer, eds., Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: http://books.google.com/books?hl=en&lr=&id=W6ZNcAolVbwC&pgis=1 [Accessed July 9, 2014].

Stango, V., 2004. The Economics of Standards Wars. *Review of Network Economics*, 3(1). Available at: http://www.degruyter.com/view/j/rne.2004.3.1/rne.2004.3.1.1040/rne.2004.3.1.1040.xml [Accessed February 5, 2015].

Tanenbaum, A., 1988. *Computer Networks* 2nd ed., Englewood Cliffs, NJ: Prentice Hall. Available at: http://scholar.google.co.za/scholar?q=Tanenbaum++Computer+networks+++1988&hl=en&as_sdt=0,5#6 [Accessed October 10, 2014].

Tummarello, G., Delbru, R. & Oren, E., 2007. The Semantic Web. In K. Aberer et al., eds. *6th International Semantic Web Conference*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 552–565. Available at: http://www.springerlink.com/index/10.1007/978-3-540-76298-0 [Accessed February 5, 2015].

UPU/POC Addressing Group, 2006a. Universal Postal Union – Addressing standard S42-a. Available at: http://www.upu.int/en/activities/addressing/s42-standard.html [Accessed October 17, 2014].

UPU/POC Addressing Group, 2006b. Universal Postal Union – Addressing standard S42-b. Available at: http://www.upu.int/en/activities/addressing/s42-standard.html [Accessed October 17, 2014].

De Vries, H.J., 2006. *Advanced Topics in Information Technology Standards and Standardization Research, Volume 1* K. Jakobs, ed., IGI Global. Available at: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-938-0 [Accessed February 5, 2015].

Vries, H.J.D., 2006. *Advanced Topics in Information Technology Standards and Standardization Research, Volume 1* K. Jakobs, ed., IGI Global. Available at: http://0-www.igi-global.com.innopac.up.ac.za/chapter/advanced-topics-information-technology-standards/4654 [Accessed February 5, 2015].

Wang, E.T.G. & Seidmann, A., 1995. Electronic Data Interchange: Competitive Externalities and Strategic Implementation Policies. *Management Science*, 41(3), pp.401–418.

Van Wegberg, M., 2004. Standardization and competing consortia: The trade-off between speed and compatibility. *International Journal of IT Standards and Standardization Research (IJITSR)*, 2(2), pp.18–33.

Weissinger, R., 2014. Economic benefits of standards. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 37(3), pp.183–188. Available at: http://www.degruyter.com/view/j/piko.2014.37.issue-3/pik-2014-0015/pik-2014-0015.xml [Accessed February 5, 2015].

Weitzel, T. et al., 2003. Network Effects and Diffusion Theory. *International Journal of IT Standards and Standardization Research*, 1(2), pp.1–21. Available at: http://search.epnet.com/login.aspx?direct=true&db=buh&an=10077010 [Accessed February 5, 2015].

Van Wessel, R.M., 2008. *Realizing Business Benefits from Company IT Standardization: Case Study Research into the Organizational Value of IT Standards, Towards a Company IT Standardization Management Framework*. Tilburg University Press. Available at: http://www.narcis.nl/publication/RecordID/oai:wo.uvt.nl:3130603 [Accessed October 10, 2014].

West, J., 2007. The Economic Realities of Open Standards : Black , White and Many Shades of Gray. In S. Greenstein & V. Stango, eds. *Standards and Public Policy*. Cambridge University Press, Cambridge, UK, pp. 87–122.

Wimalasuriya, D.C. & Dou, D., 2009. Using multiple ontologies in information extraction. In *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*. New York, New York, USA: ACM Press, p. 235. Available at: http://portal.acm.org/citation.cfm?doid=1645953.1645985 [Accessed February 5, 2015].

Yeung, A.K.W. & Hall, G.B., 2007. *Spatial database systems: Design, implementation and project management*, Dordrecht, The Netherlands: Springer.

Zemmouchi-Ghomari, L. & Ghomari, A.R., 2009. Reference Ontology. In *2009 Fifth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, pp. 485–491. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5633930 [Accessed February 5, 2015].

Zhao, K., Xia, M. & Shaw, M., 2007. An Integrated Model of Consortium-Based E-Business Standardization: Collaborative Development and Adoption with Network Externalities. *Journal of Management Information Systems*, 23(4), pp.247–271.

Zhu, K. et al., 2006. Migration to open-standard interorganizational systems: Network effects, switching costs, and path dependency. *MIS Quarterly*, 30, pp.515–539. Available at: papers2://publication/uuid/1D2A404D-835D-4641-8D73-8F06EC37C1D2.

Appendix A – Acronyms and Abbreviations

ANSI	American National Standards Institute
ARSO	African Organization for Standardization
BAO	Base Address Ontology (Reference Ontology for addresses)
CEN	European Committee for Standardization
DB	Database
EDI	Electronic Data Interchange
ESRI	Environmental Systems Research Institute
ETSI	European Telecommunications Standards Institute
FOAF	Friend Of A Friend
GIS	Geographic Information Systems
GISc	Geographic Information Sciences
HTML	Hyper Text Meta Language
IEC	International Electro-technical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
INSPIRE	INfrastructure for SPatial InforRmation in the European Community
INSPIRE TWG	INSPIRE Thematic Working Group
IRI	Internationalized Resource Identifier
IS	Information Systems
ISO	International Organization for Standardization

ISO/TC 211	ISO Technical Committee 211, Geographical information/Geomatics
ITU	International Telecommunication Union's Telecommunication Standardization
JISC	Japan Industrial Standards Committee
ISO/IEC JTC1	ISO/IEC Joint Technical Committee 1 (for Information Technology)
LOD	Linked Open Data
NEN	Netherlands Standardization Institute
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OMG	Object Management Group
OOP	Object Oriented Programming
OWL	Web Ontology Language
RDBMS	Relational DataBase Management Systems
RDF	Resource Description Framework
RDFS	RDF Schema
SA	Standards Australia
SADCSTAN	Southern African Development Community Cooperation in Standardization
SABS	South African Bureau of Standards
SDO	Standards Development Organizations
SNZ	Standards New Zealand
SPARQL	Simple Protocol RDF Query Language

SSO	Standards Setting Organizations
TWG-AD	INSPIRE Thematic Working Group on Addresses
UML	Unified Modeling Language
UPU	Universal Postal Union
UPU-S42	Universal Postal Union – Standard 42: International Address Standard
UPU/POC	UPU Postal Operations Council
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
XML	Extendible Markup Language
XSD	XML Schema Definition
XSPARQL	Language that combines XQuery and SPARQL