UNIVERSITE DE BOURGOGNE

Le2i – Laboratoire Electronique, Informatique et Image

Ecole Doctorale E2S

THÈSE

Pour obtenir le grade de

Docteur de l'Université de Bourgogne

Discipline: Informatique

par

Ashish Karmacharya

Le 30 juin 2011

Intégration d'une couche spatiale dans l´architecture du Web Sémantique:
Une proposition via la plateforme ArchaeoKM

Introduction of a spatial layer in the Semantic Web framework: a proposition
through the Web platform ArchaeoKM

M. Pr. Klaus Böhm	Université de Mayence	Examinateur
M. Pr. Jorg Klonowski	Université de Mayence	Examinateur
M. Pr. Gilles Kassel	Université de Picardie Jules Verne	Rapporteur
M. Pr. Abder Koukam	Univ. Techno. de Belfort- Montbéliard	Rapporteur
M. Pr. Christophe Nicolle	Université de Bourgogne	Examinateur
M. Pr. Frank Boochs	Université de Mayence	Co-Directeur
M. Dr. Christophe Cruz	Université de Bourgogne	Co-Encadrant
M. Pr. Franck Marzani	Université de Bourgogne	Directeur

Dedicated to my father

Acknowledgement

I am very grateful to Prof. Dr.-Ing. Frank Boochs for the trust and patience he had on me during the course of work. He was always there to support me even during the phase when it seemed that it will not be possible to pull things out. Without his faith and words of encouragements I don't think I would be able to cross this line. I will always be thankful to him.

I want to thank Prof. Dr. Franck Marzani for his support during the course of research work. He has always encouraged me. I would always feel motivated when I am down and out after reading his encouraging emails and they have been instrumental during my research work. It is a great honor to work under his guidance.

With Dr. Christophe Cruz, I have found a friend for life. He is not just a mentor for me he has pulled out of difficulties as a good friend. He has always been very kind to me. I have been very lucky to have a supervisor like him with whom I could depend so heavily and at the same time argue his opinions. I want to thank him from the bottom of my heart for everything that he has done during this period. I cannot imagine completing the research without him.

This research work would have suffered if the archaeologists from Hochschule Bochum have not collaborated as they had done. It was completely new world for them and the way they embraced it needs to be applauded. I would like to thank Ms. Bianca Khil and Ms. Elke Schneider for their cooperation in implementing something which is completely out of box for them. Likewise, I should thank Ms. Julia Petzek for her assistance in finalizing the ontological structure from the archaeological point of view.

I would like to take this opportunity to thank all the staffs of i3Mainz especially Mr. Andreas Marbs, Prof. Dr. Kai-Christian Bruhn, Mr. Nikolai Bock, Mr. Guido Heinz, Ms. Nichole Conseil, Ms. Anja Cramer, Ms. Margaritha Vogt for their support during the research period.

I want to thank all the members of the jury and especially the reviewers who have taken out their precious time to evaluate the work and give their vital suggestions. These comments are very important to undertake the future steps and will help me to position better during undertaking similar research activities.

Lastly, I want to thank my family. I must thank my wife Rubi who always stood by my side when I was losing hope. She always believed that I can pull this out when even I doubted myself. During the course of time she gave me a very beautiful gift – our son Aryan. I will always be grateful to her for raising him mostly alone. Thank you Rubi and I will always love you. Finally, this research work would not have completed without courageous words from my mother. Thank you mum and I love you.

Reasoning is the door to knowledge.

Mots-clés

Web Sémantique, pile du Web Sémantique, Ontologie de haut niveau, Gestion de la connaissance, Archéologie industrielle, Système d'information géographique, Analyse spatial, Inférence, OWL, SWRL, SPARQL

Key words

Semantic Web, Semantic Web Stack, Top Level Ontology, Knowledge Management, Industrial Archaeology, GIS, Spatial Analysis, Inference, OWL, SWRL, SPARQL

Abstract

Spatial technology has gained momentum under database systems. More specifically, the spatial operations and spatial functions are used to carry out spatial analysis which can be executed through these database systems. In addition, there has been significant amount of research in the field of the geospatial ontology domain in order to achieve the semantic interoperability between different data sources. Although, data interoperability is one of the main objectives of the Semantic Web technologies, the potentiality of the underlying knowledge tools and techniques have not been completely identified. With the growing influence of the Semantic Web technologies towards the application based on knowledge management and intelligent systems, the geospatial application benefits from this influence. This thesis emphasizes on the use of knowledge to manage spatial data within spatial information systems through the Semantic Web framework.

This research activity is carried out with the backdrop of the case study of the industrial archaeology. It sets up an ideal environment for the application of knowledge to manage the huge and heterogeneous dataset. The use of knowledge to manage the diversity of information was well executed through the application prototype named ArchaeoKM which is based on the Semantic Web. The ArchaeoKM framework follows the 4Ks processing steps: Knowledge Acquisition, Knowledge Management, Knowledge Visualization and Knowledge Analysis. The same processing principle of 4Ks was implemented during the spatial knowledge processing. A top level ontology was developed in order to serve as the background representation of the case study in order to adjust the spatial components. Keeping the custom, the spatial knowledge processing begins with acquiring spatial signatures of the identified objects. The spatial signatures are stored within the spatial database system with proper mapping to the objects in the knowledge base. The spatial knowledge of these objects is managed through executing the spatial functions at the database level and enriching the knowledge base with the results. This spatially enriched knowledge base is used again to analyze the spatial knowledge. This research thesis benefits from Semantic Web Rule Language in order to infer knowledge. In addition, the spatial built-ins proposed during the course add up spatial dimension to the SWRL for spatial inferences. Similarly, a spatial extension of the query language SPARQL is proposed in order to query spatial knowledge from the knowledge base.

Actually, this research thesis provides the initial steps in integrating spatial components within the Semantic Web framework. This integration process is important for both technologies. Regarding the Semantic Web, the integration of non-typical semantic information within this framework opens up doors to other data pattern making the transformation of technologies easier. Likewise, geospatial technologies and GIS systems benefits through the inclusion of knowledge in the analysis process making the analysis much closer and efficient to human interpretation.

Résumé

L'analyse spatiale de données géographies connaît un regain d'intérêt dans la communauté des bases de données relationnelles. Plus spécifiquement, les opérations et les fonctions spatiales utilisées comme base de l'analyse spatiale sont implémentées par les grands noms des systèmes de gestion de bases de données relationnelles limitant ainsi l'hétérogénéité structurelle des systèmes. En outre, la littérature est abondante en publications dans le domaine des ontologies spatiales afin de limiter l'hétérogénéité sémantique des sources de données tout en améliorant l'interopérabilité de ces données. Bien que l'interopérabilité des données soit l'un des objectifs du Web Sémantique, tout le potentiel de ces outils et de ces techniques basés sur la connaissance n'a pas été révélé. Avec l'influence sans cesse croissante du Web Sémantique à travers ces outils et applications en gestion de la connaissance et système intelligent, les applications utilisant des données géospatiales suivent ce phénomène en bénéficiant de son influence. Cette thèse se focalise sur l'utilisation de la connaissance métier afin de gérer des données spatiales à l'aide des technologies du Web sémantique.

L'activité de recherche menée dans le cadre de cette thèse est réalisée sur des données provenant du domaine de l'archéologie industrielle. Cet environnement se caractérise par son hétérogénéité et sa grande quantité de données offrant ainsi un cadre idéal pour la réalisation d'un outil de gestion de connaissance. Cet outil basé sur les technologies du Web Sémantique a été prototypé sous le nom d'ArchaeoKM suivant le principe des 4 K, Knowledge Acquisition, Knowledge Management, Knowledge Visualization and Knowledge Analysis. Ce même principe est mis en œuvre pour les données spatiales. Une ontologie de haut niveau a été développée pour servir de cadre applicatif à la gestion des données spatiales permettant d'ajuster une ontologie de domaines sans composante spatiale. Le processus de gestion de la connaissance commence avec l'acquisition de la signature spatiale des objets identifiés. Cette signature est stockée dans un système de gestion de bases de données spatiales et est référencée par l'objet correspondant dans la base de connaissance. La connaissance spatiale de ces objets est générée à l'aide des fonctions et des opérations spatiales au niveau de la base de données spatiale et l'enrichissement de la base de connaissance est réalisé avec le résultat de ces opérations et fonctions. L'inférence de nouvelle connaissance sur la base des données existante est réalisée à l'aide de SWRL (Semantic Web Rule Language). De plus, ce langage a été étendu à l'aide de nouveaux built-ins spatiaux afin de prendre en sidération la dimension spatiale des données. De même, cette dimension spatiale a été apportée au langage SPARQL afin de réaliser des requêtes spatiales sur la base de connaissances.

En effet, l'objectif principal de cette thèse est d'initier le premier pas vers l'intégration des composantes spatiales avec les technologies du Web Sémantique. Le processus d'intégration est premier plan pour les deux technologies. D'un point de vue Web Sémantique, l'intégration de données non communes dans ce cadre applicatif ouvre la porte à l'intégration de données beaucoup plus large. D'un point de vue des systèmes d'information géographique, l'inclusion de la connaissance permet une gestion métier des données rendant l'analyse plus proche de l'interprétation humaine.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

This Introduction provides a general overview of the research thesis. It discusses its context and motivations. It highlights the aims and objectives of the thesis work. Additionally, the approach in which this research was carried out is presented within this chapter. Lastly, this chapter explains how this thesis is organized.

1.1   Context of the Thesis

The abstraction of the real world melds the semantics of its objects with the spatial characteristics seamlessly. This is visible in a way the human perceives the real world where it is often difficult to pin point the spatial characteristics of the objects from their semantics. In other words the spatial characteristics are generally hidden with the semantics of the objects. As for example, describing relations of objects the terms near, far or touching are often used which are spatial relations but in general considered as semantic properties which is not true. Hence, it is a trend to consider that the spatial behaviors of objects are parts of its semantics. Similar approaches where the spatial properties are considered as part of semantics have been translated in technical advancements made by the technologies. There is a general trend to mix up spatial components in the semantics or the semantics in the spatial components within technologies. For instance, a classic GIS ignores semantics of objects to focus on the spatial components whereas a non GIS uses spatial components as the semantic parameters of the objects. As the technology is getting matured, it is moving closer to the human perception of the real world. Today, the knowledge management is being researched in real sense to model and to manage knowledge possessed by humans which is basically the perception of the real world.

The emergence of Internet technologies has provided a strong base to share the information in a wider community. As the needs of information have grown it has become necessary to represent them in a proper and meaningful way. It involves attesting semantics to the documents. The major approach to attach semantics to documents involves first to categorize them properly and then to index them with the relevant semantics for efficient retrieval. This categorization and indexing of the Web documents have become important topic for research. These researches focus on the use of knowledge management to structure documents which involves ontologies to conceptualize knowledge of a specific domain. Then, there is knowledge representation which is a vital part of knowledge management. It consists of possibilities to represent knowledge in order to be inferred. Knowledge representations and reasonings have traditionally been a domain within Artificial Intelligence. However, the recent growth in Semantic Web technologies has added fuel to the use of knowledge explicitly in a Web environment. The XML-based knowledge languages could be inferred through different inference mechanisms in order to infer knowledge.

1.1.1 Knowledge Management and the Semantic Web

The current version of the Web could only be processed through human intelligence. Though the Internet technologies have taken a huge leap forward since it evolved but the fact is the information within the technology should be interpreted by human brain. However, in his paper (Berners-Lee, Hendler, & Lassila, 2001), Tim Berners-Lee and coauthors have envisaged the next generation of the Web which they call "the Semantic Web". In this Web the information is given with well-defined meaning, better enabling computers and people to work in cooperation. Adding on, the Semantic Web aims at machine-processable information enabling intelligent services such as information brokers, search agents and information filters, which offer greater functionality and interoperability (Decker, et al., 2000). Since then the technology has moved significantly forward and has opened the possibility of sharing and combining information in more efficient way.

The association of knowledge with Semantic Web has provided a scope for information management through the knowledge management. Since both the technologies use ontology to conceptualize the scenarios, Semantic Web technology could provide a platform for developments of knowledge management systems (Stojanovi & Handschuh, 2002). The ontologies are core to both the technologies in whichever methods they are defined. The Semantic Web defines ontologies through XML based languages and with the advancements in these languages.

The major context behind this thesis is the use of knowledge in order to manage huge sets of heterogeneous dataset in a Web based environment. It primarily focuses on the spatial dataset and its management through the available spatial technologies incorporated through the knowledge. As the Web technologies get matured through its approach in the Semantic Web, the implementation of knowledge in this domain seems even more appropriate. This research thesis puts forward the views and result of the research activities within the backdrop of the Semantic Web technologies and the knowledge management aspect within it.

1.1.2 Knowledge Representation and Ontologies

Knowledge representation has been described in five distinct roles it plays in (Davis, Shrobe, & Szolovits, 1993). Those roles are

- A surrogate for the *thing* itself used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than acting it.
- A set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- A fragmentary theory of intelligent, reasoning, expressed in terms of three components
  - The representation's fundamental conception of intelligent reasoning
  - The set of inferences the representation sanctions; and
  - The set of inferences it recommends
- A medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished.
- A medium for human to express, i.e., a language human expresses things about the world.

Semantic Web technologies use these roles to represent knowledge. The first and the last roles are primarily theoretical roles through which knowledge could be better understood. The remaining roles are conceptual roles which are being implemented within the technology. If those roles are carefully evaluated, it could be seen that knowledge representation begins with ontological commitments. That is selecting a representation means making a set of ontological commitments (Brachman, Eugene, Norton, & Martin, 1978). Thus defining ontology is a major activity with the process of the Semantic Web.

The term *Ontology* is being used for centauries to define an object philosophically. The core theme of the term remains the same in the domain of computer science however the approach in defining it has been modified to adjust the domain. Within the computer science domain, ontology is a formal representation of the knowledge through the hierarchy of concepts and the relationships between those concepts. In theory ontology is a *formal, explicit specification of shared conceptualization* (Gruber, 1993) In any case, ontology can be considered as formalization of knowledge representation and Description Logics (DLs) provide logical formalization to the Ontologies (Baader, Horrocks, & Sattler, 2003).

Description logics (DLs) [(Calvanese, Giacomo, Lenzerini, & Nardi, 2001); (Baader & Sattler, 2000)] are a family of knowledge representation languages that can be used to represent knowledge of an application domain in a structured and formally well-understood

way. The term "Description Logics" can be broken down into the terms *description* and *logic*. The former would describe the real world scenario with the real world objects and the relationships between those concepts. More formally these objects are grouped together through unary predicates defined by atomic concepts within description logics and the relationships through binary predicates defined by atomic roles. The term *logic* adds the fragrance of logical interpretations to the description. Through these logics one could reason the description for generating new knowledge from the existing one.

As the Semantic Web technologies matured, the need of incorporating the concepts behind description logic within the ontology languages was realized. It took few generations for the ontology languages defined within Web environment to implement the description language completely. The Web Ontology Language (OWL) [(Bechhofer, et al., 2004); (Patel-Schneider, Hayes, & Horrocks, 2004)] is intended to be used when the information contained in documents needs to be processed by applications and not by human (McGuinness & Harmelen, 2004). The OWL language has direct influence from the researches in Description Logics and insights from Description Logics particularly on the formalization of the semantics (Horrocks, Pater-Schneider, McGuinness, & Welty, 2004). In addition, the OWL language has its correspondence to the description logics with its sublanguages as OWL DL and OWL Lite.

The horn logic more commonly known the Horn clauses is a clause with at most one positive literal. It has been used as the base of logic programming and Prolog languages (Sterling & Shapiro, 1994) for years. These languages allow the description of knowledge with predicates. Extensional knowledge is expressed as facts, while intentional knowledge is defined through rules (Spaccapietra, Cullot, Parent, & Vangenot, 2004). These rules are used through different Rule Languages to enhance the knowledge possess in ontology. The Horn logic has given a platform to define Horn-like rules through sub-languages of RuleML (Boley, 2009). Summarizing, it could be said that ontology defines the data structure of a knowledge base and this knowledge base could be inferred through various inference engines. These inference engines can be perform under Horn logic through Horn-like rules languages.

Semantic Web technology is slowly revolutionizing the application of knowledge technologies and though they existed before Semantic Web, the implementation in their fullness is just being realized. This research benefits from the existing inference engines

through the inference rules and reasoning engines to reason the knowledge. However, the research works moves beyond semantic reasoning and semantic rule processing and attempts to integrate the spatial reasoning and spatial rule inference integrating spatial components in its structure. This research thesis introduces the approach on achieving the spatial functionalities within those inference engines.

1.1.3   Spatial Components in Semantic Web

The Semantic Web technologies is slowly gaining acceptance in the wider community. It is thus paramount to include every type of information within the technology. The core within Semantic Web technologies is the semantics of the resources. These semantics may be the spatial or non-spatial. However, the focus of the technology is mainly on utilizing the non-spatial semantics for managing the information. Thus, the spatial information is widely neglected. Nevertheless, it has been realized inclusion of spatial components within Semantic Web framework. Those researches mainly focus on semantic interoperability of spatial data for efficient exchange of spatial data over heterogeneous platforms or efficient data integration. In cases like [(Cruz, Sunna, & Chaudhry, 2004); (Cruz, Geospatial Data Integration, 2004)], the ontologies are used to map their concepts to a global concept within a global ontology and thus providing a common platform for data integration. This is a common trend of practice for managing heterogeneous data source through Semantic Web technologies. The same practice is applied for geospatial data sources. In other cases like (Tanasescu, et al., 2006), ontologies are used to manage the semantics within different data sources to maintain the semantic interoperability of spatial data within different platforms.

In the realm of geospatial and temporal concepts and relationships, the work has not yet reached a level of either consensus or *actionability* which would allow it to be basis of knowledge interoperability (Lieberman, 2007). The Open Geospatial Consortium (OGC) is playing a major role to develop a consensus among different stakeholder on various aspect of geospatial technologies. The data interoperability is a major area in which OGC is concerned upon and it has developed different standards for this. Groups like Geospatial Incubator have taken the works of OGC to formulate steps in updating the W3C Geo vocabulary and preparing the groundwork to develop comprehensive geospatial ontology. In the process it has reported different spatial ontologies that exist in the Web (Lieberman, Singh, & Goad, W3C Geospatial Ontologies - W3C Incubator Report, 2007).

It is evident that the geospatial ontologies are developed to solve individual spatial problem and are not being used to be effective for knowledge formulation within the Semantic Web framework. Existing ontologies or in the process of creation are mostly targeting the usage of vocabularies for the proper data management and not the knowledge management. One implication of such approach is that there is no possibility of geospatial reasoning to enhance the knowledge base. It is widely noticed there is the lack of a known, robust geospatial reasoners. Furthermore, it has been argued that while geospatial reasoning is an ever-evolving field of research, spatial data constructs are not yet accommodated within most current Semantic Web languages as the OWL language (Reitsma & Hiramatsu, 2006).

The seamless integration of spatial components within Semantic Web technologies is the major topic of this research thesis. Hence, the approach in which this component is integrated within the global framework of spatial Web technology is covered extensively within this research thesis. Additionally, it discusses different components involved in spatial activities within the framework.

1.1.4 Spatial Components on Database Systems

It has been seen that in the previous section that the ontology engineering has not gained enough momentum to assist spatial activities only through ontology. Hence, this thesis work utilizes the existing potentiality of spatial extensions within the current database system be exploited to carry out the spatial activities within the ontology.

Most of the database systems support spatial operations and functions through their spatial extensions. Over the past decade, as Relational DataBase Management System (RDBMS) has seen a huge growth in the database technology, the spatial components within them also seen a tremendous improvement in their functionalities. In early days, spatial data were organized in dual architectures which consist of separate administrative data for data management in a RDBMS and spatial data for a GIS system. This could easily result in data inconsistency hence all the database systems today maintain the spatial component in a single RDBMS.

In order to have a common standard among different database systems, they implement their spatial performance accordance to the Open Geospatial Consortium (OGC 1998) Simple Features Specifications for SQL (OGC 1999). Since OGC Simple Feature Specifications are built within simple spatial features in 2D space, most of the spatial operations are restricted to

2D spatial data. It is also possible to store, retrieve and visualize 3D data but it does not follow OGC simple feature specifications. Some RDBMS system today also supports certain 3D spatial queries as well.

According to OGC specification any object is represented spatially following two structures – geometrical and topological. The geometrical structure is the simple feature specification providing the direct access to the coordinates of the objects. The topological structure provides the information about the spatial relationships of the objects. The database systems store the geometrical information of the objects and not their topology. They then use their spatial operations to retrieve topological relationships between these geometries (Hellerstein, Naughton, & Preffer, 1995).

This research thesis discusses the recent growth in spatial technology in the current database system. Additionally, it covers how the spatial components are managed within database systems with reference to the three prominent database system that are widely used today. Since, the research work utilizes PostgreSQL with its spatial extension PostGIS for the spatial activities within it, the approach of implementations of spatial components converges to the functionalities provided by PostGIS.

1.2 Aims and the motivation of the thesis

It is a general fact that technologies always shift for the betterment and the components of the previous technologies must be upgraded to the shifting technology. The world is experiencing a shift in technology from the database oriented Information technology to ontology oriented knowledge base technology and thus each individual technology that have matured under previous technology requires to be shifted to this emerging technology. The tasks of shifting these components have always presented challenges as the principle foundations between the two technologies are mostly entirely different.

One of the major technical components in the database oriented technologies is the spatial technology. The immense strength of spatial technology was realized long before the emergence of database or even the computers. Maps were used to analyze the problems and derive solutions spatially (Berry, 1999). With the evolution of computers, a new discipline emerged to analyze the problems spatially, which is termed as Geographic Information System (GIS). GIS technology was one of the first to use the spatial technology for the

analysis of the geographic locations. However, it would be incorrect to consider geographic analysis as complete spatial analysis. Spatial analysis is used in other domains as well. Before the emergence of sophisticated database systems, GIS technologies used files to store the spatial data. Each vendors of the technology had their own algorithms for spatial operations and functions. This in turn provided lots of inconsistency in the analysis process. As the database technology matured, it started to include those spatial components into it. In this manner, the spatial technology got immersed within the database technology. As previously mentioned they followed the specifications provided by OGC to maintain a common standard and hence most of these inconsistencies were revolved. With the advancement in database systems the spatial technology also got matured and today it is not necessary to depend on a GIS to perform spatial analysis. This has clear advantages for the other domains which use spatial analysis as part of their analysis process.

When viewed from the Semantic Web point of view, the integration of spatial component will trigger the integration process of other data component adding an open layer for data type which could be argued as non-typically semantic within its framework. This data could be spatial or temporal data or even process data. Such level within the technical framework of Semantic Web will give clear advantages for the technology to grow.

The main aim of this research thesis is to initiate the process of setting up a layer in the Semantic Web framework for the non-typical semantic information that is not covered through the semantics. In order to illustrate its applicability, this research centers on integration of spatial component within the Semantic Web technologies. This work focuses beyond data interoperability and addresses the spatial processing through knowledge querying and inferring. In addition, the work attempts to change and to improve the ongoing data management process of archiving documents in the industrial archaeology domain into knowledge management process.

This work also aims to initiate the usage knowledge for performing spatial analyses in the existing GIS tools. It tries to draw attention towards the benefit of introducing a knowledge level in the universal GIS model. This in fact supports the relevancy theory of the need to transfer the technical component in the wake of technology change.

1.3   Industrial Archaeology: The Case Study

The research thesis is drawn around the case study of industrial archaeology. The discipline of industrial archaeology fits perfectly to demonstrate the effectiveness of the implementation of the research activities. In general the industrial archaeological sites are available for very short duration of time and the amount of information collected is huge and diverse making it impossible for the conventional technologies to manage them. This research takes on the Semantic Web and its underlying knowledge technology to manage them. The knowledge possess by archaeologists is used to identify the objects and map the data and documents to the respective objects. In this process the knowledge about the objects is acquired through first identifying the objects and defining their behavior at the ground. This knowledge can be then used during the management of these objects. In fact the research thesis is based on 4Ks processing steps: Knowledge Acquisition, Knowledge Management, Knowledge Visualization and Knowledge Analysis. In each of these 4Ks, the knowledge of archaeologists is used.

The research site lies in Krupp belt Essen. This 200 hectares site was used for steel production early nineteenth century but was later destroyed. The majority of the area was never rebuilt. The site was excavated in 2007 in order to document the findings. The area is being converted to a park of the main building of ThyssenKrupp so there was not much available time to document the finding properly due to ever changing structure of data and documents and their volume. This is hence not possible to use the traditional technology for their rigid nature and huge dependency on human manipulation of the data and documents. Possibility to engage machine to understand the information and process them through the collaboration of the knowledge possess by archaeologist was realized through an application tool – The Web platform ArchaeoKM.

The research highlights the importance of non-typical semantic information within the Semantic Web framework. It puts forward its view citing the gap present for information in the architecture of the Semantic Web framework. The research discusses the possibility of including spatial technology within the framework. The approaches it takes from the side of the Semantic Web technologies while integrating spatial technologies might prove beneficial while integrating other similar forms of information. During the course, a layer is proposed for spatial data pattern that utilizes the Semantic Web component to process spatial

knowledge. This layer could host other data patterns as well and follow the same trend of spatial integration.

The integration of spatial technology within the Semantic Web technologies adds up benefit to the geospatial community. Instead of depending on the information based on the data, the analysis process should be more efficient and less demanding through the application of knowledge. The approach of using knowledge supported by underlying spatial data to execute the analysis process was embraced by the research.

1.4   The Roadmap

The thesis is divided into three major sections with chapters discussing them. The first section constitutes the background discussion which discusses the pretexts behind the research works. They discuss basically the state of art technology and their relations to this research work. In addition there is one chapter dedicated to the discussion on the nature of case study. The next section defines the arguments that this research puts forward and its implications. The third section presents the implantation of the argument.

1.4.1   Pretext

Chapter 2	This chapter presents the case study of the industrial archaeology and argues its suitability to experimentations of the Semantic Web tools and techniques. It also introduces the ArchaeoKM platform by discussing the processing steps of 4Ks and their realization in the system architecture. It discusses the current research works in the field of information systems for cultural heritage and points out their limitations
Chapter 3	This chapter presents the general overview on Geographic Information System technologies and different components behind these technologies. It also lays out the specific areas of the spatial technology in which this research focuses on. Starting with the general overview of the technologies and then converging to specific components required for the research activities, this chapter attempts to provide background information on the GIS technology and its components in particular the ones which are provide base for the integration work.
Chapter 4	This chapter presents the overview of the Semantic Web technologies and underlying knowledge technologies. It presents the intentions behind semantic approach and how the use of semantics generates knowledge. It also discusses the tools and techniques that are vital for the technology.

The architecture of the technology and its different layers are presented through the nature of activities that makes these layers in this chapter.

1.4.2   Arguments

Chapter 5

The chapter presents the integration process of spatial technologies and the Semantic Web technologies at the backdrop of the industrial archaeology, and its associated tool called the spatial facilitator which is a query and rule engine. The technologies discussed in previous chapters are used and adjusted for processing the spatial knowledge through knowledge technologies within the Semantic Web framework in the research works. This chapter attempts to outline the methods and the processes of these adjustments and how they return the results through knowledge tools as SWRL and SPARQL.

1.4.3   Implementation

Chapter 6

The chapter presents the application prototype ArchaeoKM to demonstrate the applicability of the concepts that were presented in the earlier chapters. Moreover, the chapter covers the technical frameworks on which the ArchaeoKM frameworl is based. The realization on spatial integration through a demonstration tool is also presented in this chapter. This tool is evident that the spatial technologies could be implemented within the Semantic Web framework to carry out spatial knowledge management.

Lastly, the thesis report concludes with concluding remarks on the thesis works. It also compares itself to a research work which utilizes knowledge techniques with the Semantic Web framework to semantically annotate the spatial information. It observes the contributions that it has made with the results and proposes future course of research activities.

References

Baader, F., & Sattler, U. (2000). An overview of tableau algorithms for description logics. *Springer-Verlag. 1847.* Scotland: LNAI.

Baader, F., Horrocks, I., & Sattler, U. (2003). Description Logics as Ontology Languages for the Semantic Web. *Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence .*

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., et al. (2004, February 10). *OWL Web Ontology Language.* Retrieved November 27, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-ref/

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific AmericaN*, pp. 34-43.

Berry, J. K. (1999). GIS TECHNOLOGY IN ENVIRONMENTAL MANAGEMENT: a Brief History, Trends and Probable Future. In D. L. Soden, & B. S. Steel (Eds.), *Handbook of Global Environmental Policy and Administration* (pp. 49 - 81). Decker, Marcel Inc.

Boley, H. (2009, August 11). *RuleML: The Rule Markup Initiative*. Retrieved November 27, 2009, from RuleMLL: http://ruleml.org/

Brachman, R., Eugene, C., Norton, G., & Martin, Y. (1978). *KLONE Reference Manual.* Cambridge: Bolt Beranek And Newman Inc.

Calvanese, D., Giacomo, G. D., Lenzerini, M., & Nardi, D. (2001). Reasoning in expressive description logics. In A. Robinson, & A. Voronkov (Eds.), *Handbook of Automated Reasoning* (Vol. II, pp. 1581-1634). Elsevier Science Publishers.

Cruz, I. F. (2004). *Geospatial Data Integration.* Chicago: ADVIS Lab, Dept. of Computer Science, University of Illinois.

Cruz, I. F., Sunna, W., & Chaudhry, A. (2004). Semi-Automatic Ontology Alignment for Geospatial Data Integration. *GIScience 2004* (pp. 51 - 66). MaryLand: LNCS.

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is Knowledge Representation. *AI Magazine, 14*(1), 17 - 33.

Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Erdmann, M., et al. (2000). The Semantic Web: the roles of XML and RDF. *IEEE Internet Computing* , 63-74.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acqusition, 5*(2), 199 - 220.

Hellerstein, J. M., Naughton, J. F., & Preffer, A. (1995). Generalized Search Trees for Database Systems. *Proceedings of the 21th International Conference on Very Large Data Bases* (pp. 562 - 573). San Francisco: Morgan Kaufmann Publishers Inc.

Horrocks, I., Pater-Schneider, P. F., McGuinness, D. L., & Welty, C. A. (n.d.). *OWL: a Description Logic Based Ontology Language for the Semantic Web.*

Horrocks, I., Schneider, P. f., Boley, H., Tabelt, S., Grosof, B., & Dean, M. (2004, December 21). *SWRL - A Semantic Web Rule Language - Combining OWL and RuleML.* Retrieved May 22, 2009, from SWRL: A Semantic Web Rule Language - Combining OWL and RuleML: http://www.w3.org/Submission/SWRL/

Lieberman, J. (2007, June 23). *Geospatial Incubator Group.* Retrieved July 2010, 14, from W3C Incubator Activity: http://www.w3.org/2005/Incubator/geo/charter

Lieberman, J., Singh, R., & Goad, C. (2007). *W3C Geospatial Ontologies - W3C Incubator Report.* W3C Incubator Group.

McGuinness, D. L., & Harmelen, F. v. (2004, February 10). *OWL Web Ontology Language: Overview.* Retrieved July 2010, 12, from W3C Recommendation: http://www.w3.org/TR/2004/REC-owl-features-20040210/

Patel-Schneider, P. F., Hayes, P., & Horrocks, I. (2004, February 10). *OWL Web Ontology Language.* Retrieved July 2010, 12, from W3C Recommendation: http://www.w3.org/TR/owl-semantics/

Reitsma, F., & Hiramatsu, K. (2006). Exploring GeoMarkup on the Semantic Web. *Proceedings:9th AGILE Conference on Geographic Information Science* (pp. 110 - 118). Visegrád: 9th AGILE Conference on Geographic Information Science.

Spaccapietra, S., Cullot, N., Parent, C., & Vangenot, C. (2004). On Spatial Ontologies. *GEOINFO .* Campos do Jordao.

Sterling, L., & Shapiro, S. (1994). *The Art of Prolog.* MIT Press.

Stojanovi , N., & Handschuh, S. (2002). A Framework for Knowledge Management on the Semantic Web. *Poster - the 11th International World Wide Web Conference.* Hawaii.

Tanasescu, V., Gugliotta, A., Domingue, J., Gutiérrez-villarías, L., Davis, R., Rowlatt, M., et al. (2006). A Semantic Web Services GIS based Emergency Management Application . *The 5th International Semantic Web Conference (ISWC06).* Athens, Georgia: ISWC06.

Chapter 2

THE INDUSTRIAL ARCHAEOLOGY AND THE ARCHAEOKM PROJECT

This chapter establishes a link between the case study of the industrial archaeology and the integration of spatial components to knowledge technologies of Semantic Web framework. It primarily highlights the importance of knowledge in handling the huge scale dataset that is excavated during the short span of time. It secondly goes on discussing the necessity in integrating the spatial technologies within knowledge management for better spatial data handling.

This chapter begins with a discussion about a general overview of the industrial archaeology. It presents the case study of the research site by discussing the nature of the data collected during the excavation process. It then reviews the current Information Systems that are either being implemented or researched in this domain. It includes the usages of Geographic Information Systems (GIS) in this field. Then after, the chapter continues with the introduction of the ArchaeoKM project through discussion on the principle and how it is different from the existing systems. It concludes with a discussion on the future prospective of the thesis.

2.1 The domain of the Industrial Archaeology: A case study

The domain of the Industrial archaeology is the recording, study, interpretation and preservation of the physical remains of the industrially related artifacts, sites and systems within their social and historical contexts (Clouse, 1995). During the period of 18th and 19th century the industrial revolution started from the United Kingdom and spread across the world marking a major turn of the human civilization. In the course of time the industries established during the period were abandoned and replaced with new installments. These abandoned sites however hide many important histories of modern developments which need to be preserved as historical facts. Today, the domain of the industrial archaeology has occupied its position in the archaeological community as a mainstream branch of archaeology which deals with the history of constructions, the development architecture, the history of technologies, socio-economic and cultural history (Boochs, 2009). The domain of the industrial archaeology has its own challenges. It does not involve the excavation process and just documents the standing artifacts in contrast to the conventional archaeology, the discipline was initially considered as hobby archaeology and not a mainstream archaeology. Though the branch has now been taken more seriously by its contemporary branches, it still needs acceptance by the wider community as the awareness about the importance of this field in archaeology is still minimal. The lack of acceptance has its own impact here as there is no reliable tool to document the artifact as the classical archaeology and hence there is a loss of a large scale of existing relicts. Usually the industrial archaeological sites are available for limited amount of time as they are not mostly conserved for continuous excavation and they are most often the sites for new constructions. Adding on, the advancement of current data capturing technologies made it possible to capture huge and heterogeneous datasets in this limited duration. It is absolutely not possible to manage this nature of datasets in such a

limited amount of time without the intervention of machine to assist human. It thus requires human machine collaboration to manage them which is not possible through the conventional technologies.

The thesis points out these limitations and provides a prospective solution to handle the dataset through the knowledge possessed by the archaeologists and assisted by knowledge management tools within Semantic Web technology for the intervention of machine to assist the knowledge processing. This section presents the case study site used within this research work discussing the diversity and amount of data acquired through the modern technologies.

2.1.1 The main excavation area

The main excavation area lies in Krupp area in Essen belt, Germany. The 200 hectares area was used for steel production during early 19th century. The work on steel production has a critical impact on the settlement development of Essen. In this way the history of Essen is closely related to the activities of steel production in Krupp. The site grew over the decades and formed a so-called Krupp Belt (BBSR & BBR, 2009). The site was destroyed during the Second World War. Most of the area is never rebuilt. In between 1945 to 2007, the area was basically a wasteland making it an ideal site for an industrial archaeological excavation. However, the ThyssenKrupp is returning to build its new headquarters in the site by then 2010. This has raised the problem of limitation of time period for a proper management of the excavated objects. The objects are recorded as soon as they are excavated and these records are stored in a repository in their respective data formats. Hence, there is a clear lack of well-defined structure for data management. Moreover, in contrast to the conventional archaeology where the data collection and data analysis goes side by side so in that case the data structure could be designed at the beginning, the data analysis is carried out at the end in industrial archaeology so it is not possible to perceive the structure of the data at the beginning. The first challenge consists of creating a proper data structure which helps in retrieving those data efficiently. As there was not enough time to filter the collected data concurrently, the amount of data that are collected is huge. Hence, the system that has to handle the collection of data should be able to handle this huge set.

Figure 2.1: The main excavation area Site

2.1.2 Data Pattern and Collection

Archaeologists with assistance of photogrammetric specialists were involved in data acquisition process. They were responsible to decide the methods measurement to use for acquiring data. The findings were scanned through terrestrial laser scanning instruments. Two scanners were used to acquire the scanned data. They were the *Zöller and Fröhlich* scanner (*ZF*) and the *Riegl* scanner. Those two scanners were used according to their requirement. Large objects scanning were carried out with the help of the *Riegl* scanner whereas the *ZF* scanner is used whenever some important findings are excavated. The *Riegl* scanner was installed on the roof of the *Kreuzhaus* (the building marked at the bottom of the site in figure 2.1) so that the scanner gets a good overview of the area. The findings were scanned with a resolution of 0.036 degrees (6 mm on 10 m) hence the point cloud is very dense. All the data were stored in the Gauß Krüger zone II (GK II) coordinate system. The details on coordinate system are explained in chapter 3.

An orthophoto was orthorectified from the aerial images that were taken during the course of research work. The orthophoto has 10 cm resolution and is in GK II coordinate system. Huge collections of digital pictures were taken during the research activities and they were stored in

their original formats. These photos were taken with non-calibrated digital. However, certain knowledge can be extracted from them by the archaeologists. Besides, photographs documents like the site plan of the area and some documents with relevant information of the site or the objects excavated were collected during data acquisition process. These data and documents were digitized and stored for proper mapping with the relevant objects. Archaeological notes taken by archaeologists during these excavation processes are of high importance. Hence, these notes are digitized and stored in the repository. Similarly, the site plan of the area was digitized and stored as *.shp*[1] format in ArcGIS which contains shapes.

To summarize there are four kinds of data as illustrated in table 2.1:

Laser Scanning	Images	Old Archives	ArcGIS database
Resolution: 0.036 degrees (6 mm on 10 m)	Aerial Image of the site with a resolution of 10 cm.	Floor plans	Buildings archaeologists are interested on
Bunker, Oven and Wall scanned	Images taken of findings (Mostly without reference system)	Notes and sketches (old and during excavation)	

Table 2.1: The nature of data pattern

Figure 2.2 demonstrate the nature of the dataset that was collected during the research work. It is clear there are four distinct kinds of data which ranges from textual documents as the archaeological notes to multimedia documents as images. The heterogeneity of dataset is evident through the nature of each type of dataset varying completely from others in terms of their storages, presentations and implementations.

---

[1] http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
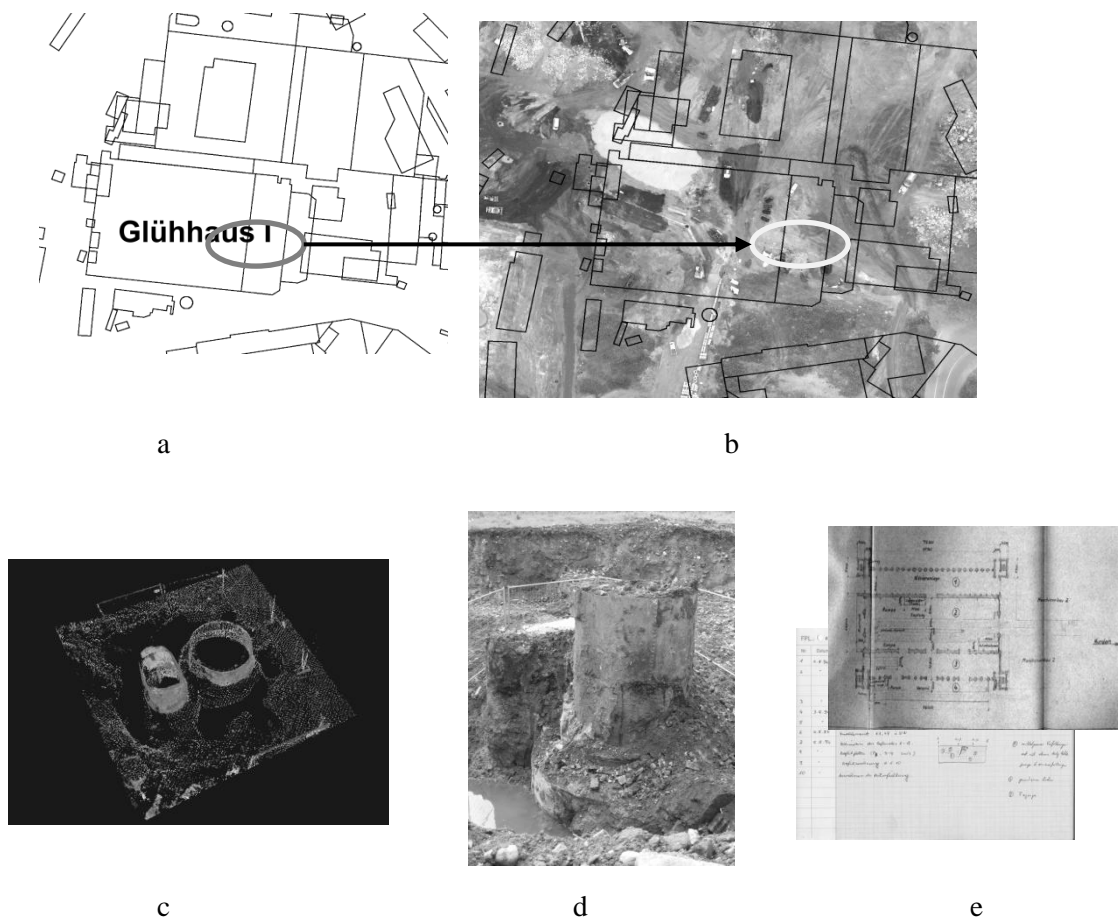
Figure 2.2: Existing data types a. the site plan of the area b. an orthophoto of the area overlayed with the site plan – the arrow pointing the area of the object within the area c. the point cloud of the scanned object d. the image of an object e. Pieces of information provided

2.1.3 GIS for archaeology

What does a GIS do? Basically providing a definition of GIS and referring to its abilities to capture and manipulate spatial data doesn't provide much insight into its functionality. The basic tasks of a GIS system can be broken down into five groups, **data acquisition**, **spatial data management**, **database management**, **data visualization** and **spatial data analysis** (Jones, 1997). Most archaeological data such as artifacts, features, buildings, sites or landscapes, have spatial and aspatial attributes that can be explored by GIS. These attributes include the spatial location that informs about the local or global context concerning the pieces of information, and the morphology that defines the shape and the size of an object.

The **acquisition** of spatial data is undertaken with the help of existing digitizing functionalities within the application software proving them. They are responsible for the

acquisition of data and integrating it to the existing spatial sets. Spatial data include, but are not limited to, topographic maps, site locations and morphology, archaeological plans, artifacts distribution, aerial photography, geophysical data and satellite imagery.

The **spatial data management** process uses sophisticated database management systems in order to store and retrieve spatial data and their attributes. Data collected from difference sources have to be transformed in the same coordinate system in order to integrate them. This involves also the building of vector topologies, the cleaning of digitalized spatial datasets and the definition of geospatial metadata.

The **database management** system, involving conceptual and logical data modeling, is an important part of GIS because it ensures that the construction and the maintenance of database is done and that the spatial and aspatial datasets and components are correctly linked.

The **spatial data analyses** part of a GIS provides the ability to undertake locational and spatial analysis of archaeological data. Much work in GIS involves the mathematical combination of spatial datasets in order to produce new data that provides insight into phenomena.

The **spatial data visualization** provides powerful visualization capabilities used for viewing spatial datasets. Many GIS provides software packages that facilitate the visualization of maps through Web or non-Web based environments.

Data Acquisition	Spatial Data Management	Database Management	Spatial Data Anlyses	Spatial Data Visualisation
• Primary data collection • Digitise maps • remote sensing • Data entry	• Coordinate transformation • Georetification • Metadata construction • Building topologies • spatial data cleaning	• Data modelling • Database construction • Metadata construction • Updating data • Creating/maintaining dtata relations	• Query by location • Query by attribute • Location analysis • Spatial analysis • Analysis of association • Visibililty modelling • Modelling of movement • Simulation of behavior • Predictive modelling • Geostatistical modelling • Surface modelling	• Digital cartography • Thematic mapping • Explore data patterns • 3DE visualisation

Figure 2.3: The five main groups of tasks performed by GIS (Jones, 1997)

Some limitations appear visible in currents GIS system in the context of the Industrial Archaeology. The lack of GIS platforms that uses data like point cloud is one of such visible limitations. Though having this limitation it is however a fact that conventionally an Information System for archaeologists is a Geographic Information System or a 3D object modeling system. The statement has been supported by the current commercial applications for the archaeologists. Applications like ArchaeoCAD from ArcTron[2] and PointCloud from Kubit[3] rely heavily on the geometry of the objects excavated. The applications are thus used primarily to represent objects excavated in a 3D space. Similarly, GIS vendors like ESRI[4] uses the spatial information of the objects to analyze them spatially. Meanwhile, the data collection process has seen a tremendous change in the last few years. Today, it is not only the amount of data that needs consideration, the diversity of data should also be taken into account. It is becoming increasingly difficult to manage them solely with the current database system due to the size and diversity of the data. In addition, information systems in archaeological projects or cultural heritage projects is lacking from a complete package. There have been lots of researches going on but they are on the independent components. However, research projects like 3D MURALE (Cosmas et al., 2001) and GIS DILAS (Wüst, Nebiker, & Landolt, 2004) contains most of the elements needed for a complete package and hence could be considered as comprehensive Information System. The 3D MURALE system is composed of a recording component, a reconstruction component, a visualization

---

[2] http://www.arctron.com
[3] http://www.kubit.de
[4] http://www.esri.com

component and database components. The findings are managed through a database management system. Once the findings are stored in the database with a proper data structure, the objects are reconstructed through the reconstruction component. This is done by modeling the objects in the 3D space. These 3D models are displayed in the visualization component. The DILAS is generic software, fully object oriented model for 3D geo-objects. The 3D geometry model is based on a topologically boundary representation and supports most basic geometry types. It also incorporates the concept of multiple levels of detail (LOD) (Balletti, Guerra , & Adami, 2005) as well as texture information. It is clear that the existing systems rely heavily on the geometries of excavated objects for their representations, but the interoperability of these systems and the knowledge sharing remains a gap.

In addition, the sharing of knowledge in archaeology and disseminate it to the general public through wiki has been discussed in (Costa & Zanini , 2008). Likewise the use of knowledge to build up a common semantic framework has been discussed in (Kansa, 2008). Research works exist in the field of archaeology, but most of the research is carried out in other related fields. However, it could be applied in archaeology as well. The existing researches focus more on using the common language for efficient interoperability. The research project (Kollias, 2008) concerns the achieving syntactic and semantic interoperability through ontologies and the RDF framework to build a common standard. Data integration through ontologies and their relationships is discussed in (Doerr, 2008). Although the work on the Semantic Web and knowledge management in the field of Information System in Archaeology or related fields is stepping up with these research works, the fact is they are in very preliminary phases today. Additionally, these projects concentrate more on how to achieve interoperability with semantic frameworks and ontologies. However, no one focuses on the knowledge generation process and more specifically on rules defined by archaeologists in order to build up the system which should use, evaluate and represent the knowledge of the archaeologists.

Knowledge contained in documents has been traditionally managed through the use of metadata. Before going on details about knowledge management, let us first understand the perspective about the whole idea. Every activity begins with data. However data is meaningless until they are put in context of space or an event. Additionally, unless the relationship between different pieces of data is defined, simply data do not have any significance. Once the data are defined in terms of space or events and are defined through

relationships, they become Information. Information understands the nature of the data but they do not provide the reasons behind the existence of data and are relatively static and linear by nature. Information is a relationship between data and, quite simply, is what it is, with great dependence on context for its meaning and with little implication for the future (Bellinger, 2004). Beyond every relationship, arises a pattern which has capacity to embody completeness and consistency of the relations to an extent of creating its own context (Bateson, 1979). Such patterns represent knowledge on the information and consequently on data. The term Knowledge Management has wide implications. However, very precisely Knowledge Management is about the capture and reuse of knowledge at different knowledge level. In order to access the knowledge, data are annotated and indexed in the knowledge base. This is in lined with the concept proposed by Web Semantic where it proposes to annotate the document content using semantic information from domain ontologies (Berners-Lee, Hendler, & Lassila, 2001). The goal is to create annotations with well-defined semantics so they can be interpreted efficiently. Today, in the context of Semantic Web, the contents of a document can be described and annotated using RDF and OWL. The result is a set of Web documents interpretable by machine with the help of mark-ups. With such Semantic Web annotation, the efficiency of information retrieval is enhanced and the interoperability is improved. The information retrieval is improved by the ability to perform searches, which exploit the ontology in order to make inferences about data from heterogeneous resources (Welty & Ide, 1999).

2.2 The ArchaeoKM project

The project ArchaeoKM plans to complement the principle of Knowledge base where it can be used by archaeologists to develop knowledge rules from the data excavated. The knowledge stored in machine readable format then is translated into human readable format. Moreover, it moves beyond managing the concepts defined to annotate documents, which most of the research projects currently focusing on, to the instances of concepts with their own property values. In this manner, an object found in a point cloud can be linked, with the help of an instance in the ontology to other documents (a part in an image or a section of archive document) that contains the same object.

One of the main focuses on ArchaeoKM project is to determine an approach of integrating the spatial data within its overall framework of data integration. The integration process did

not only serve for the data integration but also has taken a step forward in data analysis and management through the knowledge management techniques.

2.2.1 The Web platform ArchaeoKM

The challenges possessed to document the artifacts in such a site could be handled through utilizing the knowledge of responsible archaeologists. The platform ArchaeoKM focuses on the use of the knowledge of archaeologists to document the objects with respect to the surrounding. In the process a tool based on the Semantic Web technology and its underlying knowledge technology was develop to provide the archaeologists to share their knowledge and document the information collected during the excavation process. One of the challenges is to bring all the datasets previously presented in one common platform. As a knowledge representation format, the top level ontology acts as the global schema for data integration in the platform. The application tool provides a common platform for archaeologists to share their experience and knowledge.

2.2.2 The ArchaeoKM Architecture

The GIS technology performs along a group of five tasks to execute the result. These tasks as already been mentioned are acquisition of spatial data, spatial data management, database management, spatial data analysis and the spatial data visualization. The ArchaeoKM project attempts to complement the five major processing steps of a GIS through its four processing activities which its calls the processing steps of **4Ks**: **knowledge acquisition, knowledge management**, **knowledge visualization**, **knowledge analysis**.

The **knowledge acquisition** task consists in general term to define metadata on data acquired during the survey process. The spatial data acquisition process still involved during the process, but in addition metadata on these data are defined using a knowledge representation language. Actually, an ontology, which defines the semantic of the excavated features, is defined to capture and capitalized the knowledge of archeologists on the archaeological site. Hence the schema of the ontology is defined at this level. This is done by the help of a specialist on ontologies. The relationships and there semantics are stored into the ontology. This semantic could be provided through an example of the relation of "insideOf" which is transitive relationship. In mathematics, a binary relation R over a set X is transitive if whenever an element a is related to an element b, and b is in turn related to an element c, then

a is also related to c by the same kind relation. The ArchaeoKM platform deals with this issue.

The acquisition process constitutes of generation of knowledge base through enriching the ontology. The knowledge of archaeologists is used again to identify the excavated objects and enrich them in the ontology schema formulated. In short the process consists of populating the ontology with "individual" which represent objects excavated from the archaeological site. This creates a knowledge base from the ontology schema.

The **knowledge management** task consists of storing and the retrieving data along with its semantics. Knowledge is defined through the relationships and it is the relationships between individuals that create the real knowledge in the knowledge base. These relationships not only imply the relations between objects but also relation to their spatial signatures in spatial database. A specialized tool has to be developed in order to retrieve data from the ontology and from its spatial representation stored in a GIS. The ArchaeoKM platform deals with this issue.

The **knowledge analysis** task is the ability of the system to perform inferences on datasets. This cannot be undertaken without the help of the semantic definition on the archaeological objects. Usually inference or deduction is conducted on attributive data which are defined in the ontology. Today, no tool is defined to compute inference on the individuals of an ontology and its spatial definition store in a spatial database. The ArchaeoKM platform deals with this issue.

The **knowledge visualization** task provides powerful visualization capabilities used for viewing spatial datasets and its semantics counterparts. Tools for the visualization of ontologies are of benefit to visualize the results of knowledge analysis. The ArchaeoKM platform deals with this issue.

As illustrated in figure 2.4 the system architecture of the ArchaeoKM platform is a three layered architecture with a structure for spatial component standing parallel against them.
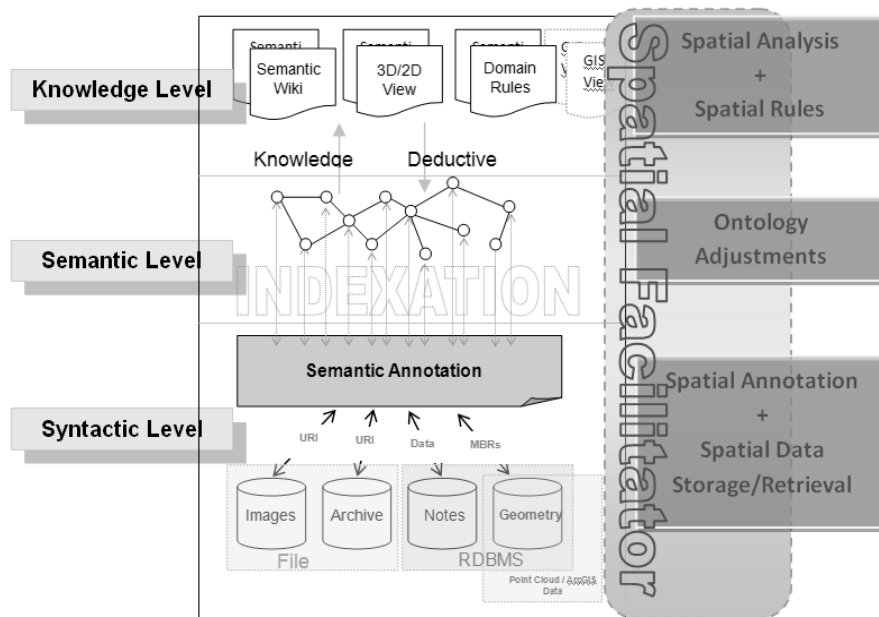
Figure 2.4: The system architecture of the ArchaeoKM

The bottom level is the Syntactic level. This level contains all the information excavated from the site. Most of the data and documents collected during the excavation process are stored in their original formats. Certain data which needs to be stored in database system such as GIS data are stored in the RDBMS. This level basically performs as the repository of the dataset. One of the main tasks of the syntactic level is to explain the data. For a proper identification, the data needs to be analyzed with reference to the objects illustrated in the index. One of the first features within the application is the identification process. A proper identification mechanism allows defining the identified objects. The ArchaeoKM platform utilizes the knowledge of archaeologists to identify the object. The identification is carried out by tagging the objects in the orthophoto of the site provided in the application. Attaching the semantic characteristics through semantic analysis on these objects generates knowledge. Different methods are used for the associating the semantic information according to the data pattern. Three distinct methods are applied to associate the semantic information which depend on the nature of the datasets with which it is associating with: Minimum Bounding Rectangles (MBRs) for the spatial data set, Uniform Resource Identifier (URI) for images and archive data and mapping to the data tables for datasets stored within RDBMS. The method is reflected by the feature Semantic Annotation within the platform. These annotations are carried out through creating individual Resource Description Framework (RDF) triplets for

each annotation process technology. RDF triplets also map the identified objects to the relevant classes in the domain ontology in the semantic level.

The next level is the semantic level, which manages the extracted knowledge. As stated, it is achieved through the ontological structure established through the descriptions, observations and rules defined by the archaeologists. These descriptions and rules are represented through different axioms in the domain ontology. Archaeologists are involved actively in this phase as they are the one best suited to provide entities and their relationships needed to build up the domain ontology. The semantic annotations from the Syntactic level will be indexed semantically to the entities of the domain ontology in this level. This semantic index through the identification process is the building block of the domain ontology and through semantic annotations provides a semantic view of the data. It also provides a global schema between various data sources making the data integration possible at certain level. This level represents a bridge between interpretative semantics in which users interpret terms and operational semantics in which computers handle symbols (Guarino, 1994). It has been discussed earlier that the acquired knowledge is managed through mapping them to the semantic information stored in different datasets through the semantic annotations. The knowledge is also managed through assigning semantic properties to the objects through proper relationships with other objects.

The top most level is the most concrete one as this level represents the organization of the knowledge on the semantic map through different visualizing tools. This level provides the user interfaces and they are visualized in form of Web pages as illustrated in figure 2.4. These Web pages represent knowledge which are generated through the knowledge management process discussed above. The pages are interrelated and can be used according to their relevance. The main representation of the knowledge is, however, demonstrated through Detail View pages. These pages are not only designed to illustrate the knowledge that has been generated and to manage it through the bottom two levels, but to also perform semantic research in order to gain new knowledge. Various techniques of the Semantic Web technology are being integrated within ArchaeoKM structure for acquiring new knowledge. Domain rules through inference engine provide one of those features in ArchaeoKM structure. In archaeology it is sometimes not possible to analyze the finding immediately and needs some properties or relationships to support them later. These inference rules provide the archaeologists such functionalities within the application.

In addition to the three levels, the system architecture contains components that facilitate the acquisition, validation, upgrade, management and analysis of the spatial knowledge. These components are packaged into the Spatial Facilitator as illustrated in figure 2.4. This component is responsible for analyzing the spatial data and providing results; either to update the current ontological structure in the semantic level or to populate the knowledge base. Through the inference capabilities in Semantic Web technology, this can then be used to explore new theories. This structure discussing the spatial components will be discussed in details later in chapter 5.

2.3 Discussion

This chapter has presented the case study of industrial archaeology for implementation of the arguments the research proposes. Industrial archaeology is the best suited for the research for the nature of the domain. The discipline of industrial archaeology generates huge and diverse data in very short duration of time and amount of time for the sites to be available for excavation are short making it not possible to manage information through the conventional technologies. It is thus apprehending that this huge and diverse information could only be managed through active involvement of the archaeologists and the knowledge possess by them.

The ArchaeoKM project uses the knowledge possessed by the archaeologists to manage the information they gathered during the excavation. It is handled through a platform based on Semantic Web technologies and knowledge management and is termed as ArchaeoKM itself abbreviating Archaeological Knowledge Management. It is based on Semantic Web technologies and its underlying knowledge technology. It uses the processing steps of 4Ks representing knowledge acquisition, knowledge management, knowledge analysis and knowledge visualization complementing the fiver steps of a GIS process. These 4Ks processing steps use the knowledge of the archaeologists in manipulating the data to manage them.

This chapter establishes a relation between the case study of industrial archaeology and the spatial knowledge modeling through highlighting the direction of the research. Primarily based on knowledge management of Semantic Web framework, it uses the spatial nature of case study to implement the spatial tools provided by the current spatial technology within the framework. The capabilities in existing tools to use the current database systems and their

spatial extension are evident of the ability of database systems to manage spatial data. It however lacks the flexibility to adapt itself into new scenarios that might arise through generation of new information or changes in the contexts due to the fact that it is still based on table based data structure. This research carries these capabilities forward by using the spatial knowledge processing through knowledge tools which provides the proper data management in archaeology that addresses the limitation in adaptation of the conventional technologies.

This chapter has presented the concept of the inclusion of spatial knowledge in handling the spatial nature of data excavated. This is new domain of research and probably one of its kinds. Hence it is important to understand the current state of art in both spatial and Semantic Web technologies. The next chapter thus discusses the state of art in spatial technology through Geographic Information System.

References

Balletti, C., Guerra , F., & Adami, A. (2005). 3D Multiresolution Representations in Archaeological Sites. *ISPRS* (pp. 654-659). ISPRS XXXVI-5/C34.

Bateson, G. (1979). *Mind and Nature: A Necessary Unity (Advances in Systems Theory, Complexity, and the Human Sciences).* Hampton Press.

Bellinger, G. (2004). *Knowledge Management—Emerging Perspectives*. Retrieved December 13, 2010, from Systems thinking : http://www.systems-thinking.org/kmgmt/kmgmt.htm

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific AmericaN*, pp. 34-43.

Boochs, F. (2009). Spatial information system for acquisition, documentation and analysis of industrial archaeological objects. *The International Committee for the Conservation of the Industrial Heritage Conference.* Freiberg: TICCIH 2009.

Clouse, R. A. (1995). *Industrial Archaeology*. Retrieved September 23, 2010, from Minnesota Historical Society of Archaeology: http://www1.umn.edu/marp/indus/indus.html

Cosmas, J., Itagaki, T., Green, D., Grabczewski, E., Waelkens, M., & Degeest, R. (2001). 3D MURALE: A Multimedia System for Archaeology. *VAST* (p. Virtual Reality: Archaeology and Cultural Heritage). Glyfada, Nr Athens: ACM.

Costa, S., & Zanini , E. (2008). Sharing knowledge in archaeology: looking forward the next decade(s). *Digital Heritage in the New Knowledge Environment: Shared spaces & open paths to cultural content.* Athens: Hellenic Ministry of Culture.

Doerr, M. (2008). The CIDOC Conceptual Reference Model - A new standard for Interoperability. *Digital Heritage in the New Knowledge Environment: Shared spaces & open paths to cultural content.* Athens: Hellenic Ministry of Culture.

Guarino, N. (1994). The Ontological Level. (R. Casati, & G. White, Eds.) *Philosophy and the congitive sciences*.

Jones, C. (1997). *Geographical Information Systems and Computer Cartography.* Addison Wesley Longman Ltd.

Kansa, E. C. (2008). Openin Archaeology to Mash-ups: Field Data and an Incremental Approach to Semantic. *Digital Heritage in the New Knowledge Environment: Shared spaces & open paths to cultural content.* Athens: Hellenic Ministry of Culture.

Kollias, S. (2008). Achieving Semantic Interoperability in Euroeana. *Digital Heritage in the New Knowledge Environment: Shared spaces & open paths to cultural content.* Athens: Hellenic Ministry of Culture.

Welty, C., & Ide, N. (1999). Using the right tools: enhancing retrieval from marked-up documents. *J. Computers and the Humanities, 33 (10)*, pp. 59 - 84.

Wüst, T., Nebiker, S., & Landolt, R. (2004). Applying the 3D GIS DILAS to Archaeology and Cultural Heritage Projects Requirements and First Results. In *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* (Vol. 35 Part 5, pp. 407 - 412). Natural Resources Canada.

Chapter 3

GEOGRAPHIC INFORMATION SYSTEMS

This chapter presents a general overview on the Geographic Information System technology and the different components behind this technology. It also lays out the specific areas of spatial technology in which this research focuses on. Starting with the general overview of the technology and then converging to specific components required for the research activities, this chapter provides background information on the GIS technology and its components in particular the ones which are provided for the integration work regarding the Semantic Web.

Geographic Information System or simply GIS is any system that collects, stores, manages and manipulate spatial data and apply them to perform various task in geo disciplines. Simplifying it, it could be said that a GIS is a technology which incorporates geographical features with the tabular data so the maps could be analyzed and resolve real-world problems (Dempsey, 2010). Here are few popular definitions defining GIS.

> *A GIS is a powerful set of tools for* **collecting, storing, retrieving** *at will, transforming and displaying spatial data from the real world (Burrough, 1986).*

> *A GIS is* **a database system** *in which most of the data are spatially indexed and upon which a set of procedures are operated in order to answer queries about spatial entities in the database* (Smith, Menon, Starr, & Estes, 1987).

> *GIS is any* **manual or computer based set of procedures** *used to store and manipulate geographically referenced data (Stanley, 1989).*

A geographic information system (GIS) is a computer-based tool for mapping and **analyzing things that exist and events** that happen on earth. GIS technology integrates common database operations such as queries and statistical analyses with the unique visualization and geographic analysis benefits offered by maps (ESRI).

Every definition defining a GIS emphasizes the role of storing and retrieving spatial data within the system. Spatial data also known as geospatial data are the information describing locations of objects in terms of coordinates. These coordinates belongs to different coordinate system to map the locations of the earth. Spatial data are usually stored as these coordinates along with the topologies of the objects. In general any geographical phenomenon represents the real world through two descriptors: **what** and **where**. The **what** descriptor describes what is present and is well-established objects and their entities. The **where** descriptor describes the location of objects and is thus the spatial extent of the objects. Hence, GIS takes both descriptive and spatial data into account to define an object. Figure 3.1 displays the two descriptors of a GIS with their components.

Spatial data

- Location (coordinates)
- Topology (relationships)
- Graphical representation (geometry)

Where

GIS

What

Descriptive data

- Well Structured RDBMS
- Multimedia documents
- Other documents
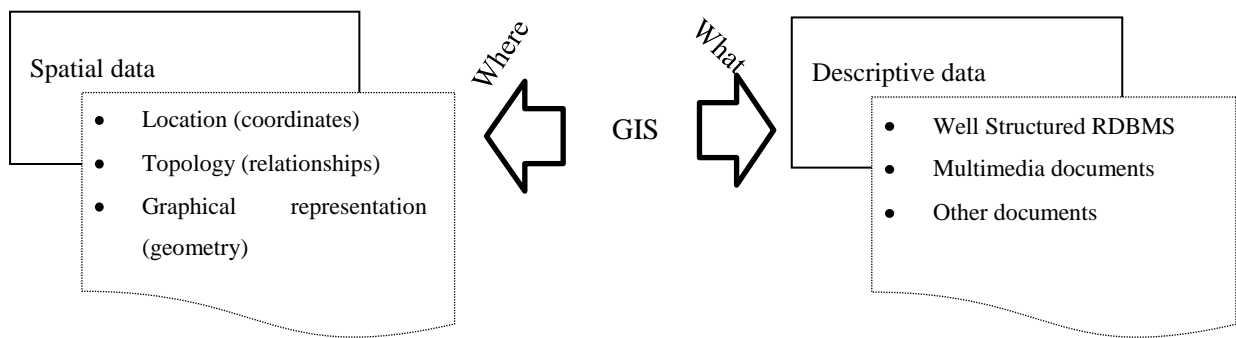
Figure 3.1: Descriptors of GIS

In general term, a location can be represented by x, y and z coordinates of longitude, latitude and elevation. Other ways can be employed to represent the coordinates. Coordinates provide spatial signature to the object. Any tangible object has its spatial signature with it. Hence, it should have some coordinate system tagged against it. The object is thus identified first with local coordinate system such as coordinate system of the container which contains the object and then with one of the global coordinate systems – the coordinate systems which apply to the identification of the object with respect to its position in the global scenario. These global coordinate systems can be at national level or the universal level. The national coordinate systems again need to be translated to the universal coordinate systems in order to identify the object universally. One of widely accepted universal coordinate system is the Universal Transverse Mercator (UTM) coordinate system. The coordinate system is based on Mercator projection. A Mercator projection is pseudo cylindrical conformal projection which means it preserves the shape of the objects. It is a cylindrical projection system and orients the 'equator' north-south (through the poles) thus providing a north-south oriented swath of little distortion (Reisterer, 2008). After that, it projects the cylindrical orientation onto the map with a successive swath through which undistorted regions are created. In the UTM coordinate system these swaths are called zones and each zone is six degrees longitudinal wide. The entire earth is covered through these six degrees zones running from west to east. This thesis research uses Gauss Krüger coordinate system which is a Transverse Mercator coordinate system and similar to UTM except it uses three degrees longitudinal wide whereas the UTM systems uses six degrees wide. The coordinates are the basic building blocks of spatial data. Likewise topology plays major role on how the spatial data are stored. A GIS topology is a set of rules and behaviors that model how different features such as points, lines and polygons share geometry (ESRI, GIS Topology, 2005). In addition, the topology defines

the geometrical relationships between two geometries. The topology has been a key to GIS for the data management and its integrity. In general, a topological data model represents any spatial objects using an underlying graph of topological primitives (ESRI, GIS Topology, 2005). The last component under spatial data in figure 3.1 is the graphical representation of the data. This is generally a digital map representing the data. Concerning the descriptive data, it is a more conventional data type so stored more conventionally. In addition, it describes the object through the attributes so they are stored and retrieved as attributive data type within a database system.

A GIS system is generally divided into three components: the data, the hardware/software and the people involved in the process (James Madison University, 2004). It has been illustrated in figure 3.2.
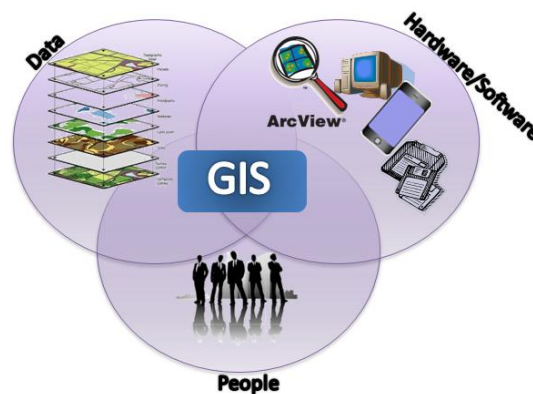
Figure 3.2: The GIS Components

- The mainly used **hardware** in the domain of GIS is the computers. Besides computer, other hardware devices are also used in a GIS which are sensors that capture spatial data such as a digitizer or Global Positioning System (GPS). Today, with the advancement in device technologies, the trend of using GIS equipped handheld devices is increasing. There is a wide range of GIS application **software** which ranges from the desktop applications to networked configured applications to handheld mobile applications. Most of them are commercial applications like ESRI ArcGIS[5] or Intergraph GeoMedia[6]. There exists numbers of open source GIS applications as well

---

[5] http://www.esri.com

[6] http://www.intergraph.com

such as GeoTools[7] or GRASS[8]. These software applications provide functions and tools needed to store and retrieve geographic information. They also provide the functionalities to query the spatial information and perform certain spatial operations for spatial analysis and then display the results. All these software packages relay on underlying database management system for the management of data. In addition, most of the existing database systems are able to store and manage the spatial data on their own without the GIS software applications.

- The GIS **data** is probably the most expensive and most important components of GIS. There are various ways to capture these data. The most common way to capture these data is through a technical process called digitizing. This process includes digitally encoding the spatial objects to acquire their spatial coordinates which is then stored in a GIS. Special consideration is taken to maintain the topological information of these digitized objects. The spatial signatures of the objects are then linked to the thematic tables in order to assign its thematic meaning.

- **People** are probably the most active component during the whole process of a GIS activity. People are involved from the start to the end of a GIS process. From the start of the problem definition to the realization of the solution it is people who are actively involved in every process. Today, a GIS is used in almost every field as this tool enables them to perform more efficiently. Hence, it is not only GIS specialists who are using GIS for analyzing GIS data now and it won't come as a surprise if this term is no longer valid in near future. All these parts are employed in the GIS processes. Figure 3.3 illustrates a typical USE CASE diagram of GIS process. Most of the GIS follow the processes provided in the diagram. The technicians are responsible for collecting the data which are either raster or vector. They feed them into a GIS. The second part is the geospatial analysis part - the analysis process is carried out by the GIS users (not necessarily GIS experts) through either raster analysis or through the vector data analysis.

---

[7] http://www.geotools.org

[8] http://www.grass.itc.it
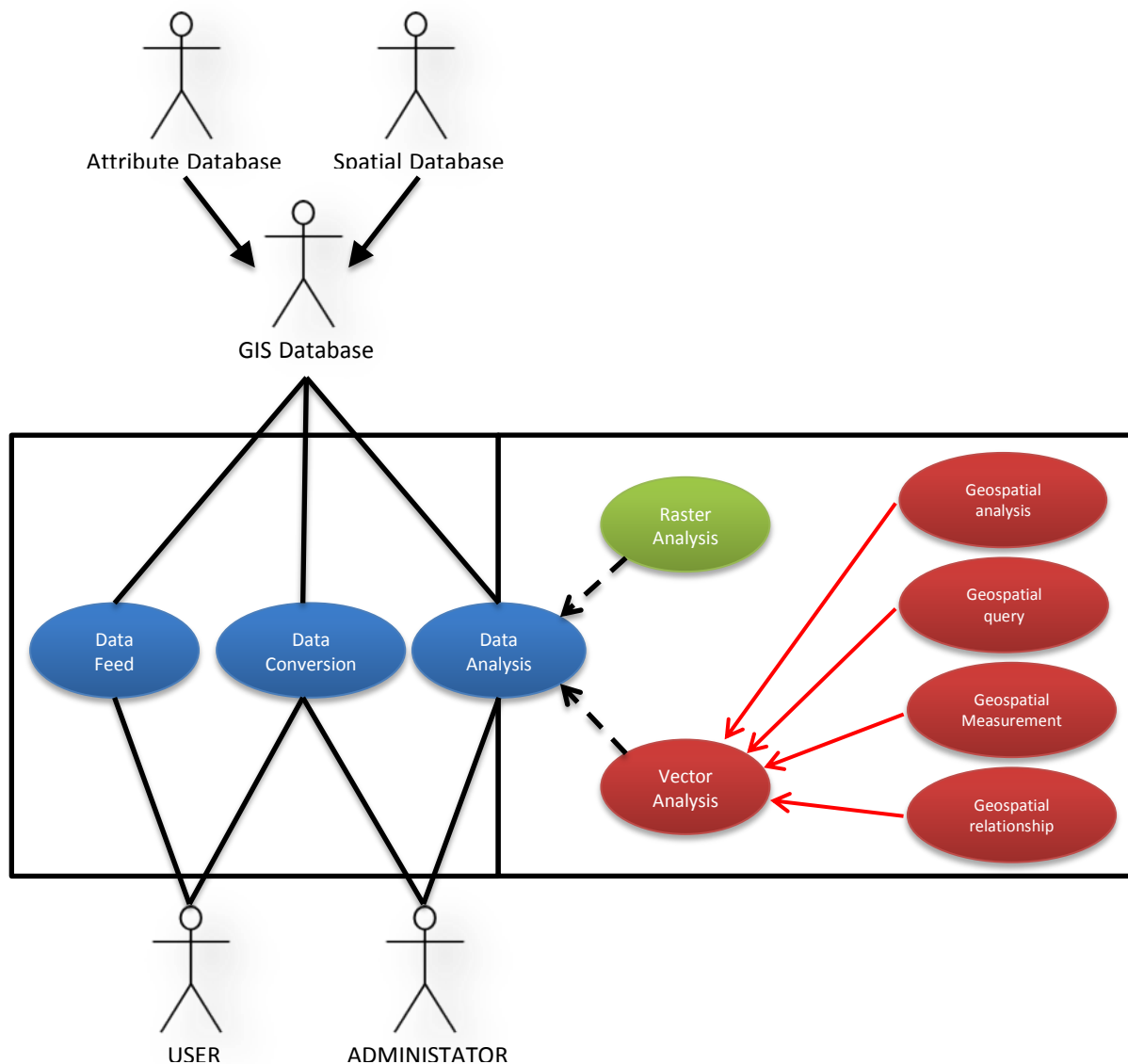
Figure 3.3: A USE CASE diagram of typical GIS Processes

(Note: Raster analysis is purposefully isolated in the figure 3.3 because this research thesis does not work on raster data.)

The geospatial data is traditionally represented as 2 dimensional maps in a GIS system. These maps represent the abstractions of the real world. In general the storage/representation of objects can be divided into two broader categories: discrete and continuous. GIS uses both these categories to store and represent the objects. These are commonly known as vector and raster representations. The next section gives more details on spatial data. Raster data are analyzed through conventional mathematical or statistical methods to generate the results and is termed Raster Analysis.

- Raster based data are increasing used within GIS community because of its potentiality of using various mathematical operations within them. Most of the operators in raster analysis are based on trigonometric, exponential and logarithmic, reclassification, selection (based on a condition), statistical, arithmetic. Raster analyses are based on the statistical and mathematical operations and are popularly used for mostly pattern analysis. Through pattern analysis one can determine the pattern of a subject over the time period. A popular example of pattern analysis is land change pattern. However, raster analysis is carried for other areas as well such as surface analysis (analyze the elevation to determine the view area).

- Vector analysis as mentioned in previous sections is the analysis process through various spatial functions and operations on vector data. It has been further classified into four sub categories which are defined in vector analysis section. The spatial functions and operations within these categories are used independently or in combination with others. They are used in conjunction with the SQL SELECT statements and the result is displayed in either maps or the resulted data. In general these queries are carried out to address requirements of certain audiences and hence based on certain themes. Such kinds of maps are generally thematic maps. Figure 3.4 illustrates examples of such kinds of maps (source: International Centre for Integrated Mountain Development (ICIMOD)).

The first map in the figure 3.4 demonstrates the elevation levels of Kathmandu Valley (a valley in Nepal comprising three cities of which Kathmandu, the capital is one) which uses the raster model. The second map displays the population density of the valley which uses vector model. Both maps have their themes classified in different levels. These maps are sometime rasterized to perform analysis on the pattern change in different periods for the pattern analysis. Details on raster analysis and how it is not directly relevant for our research work are illustrated in the Raster Analysis section.

A Spatial analysis of vector geometries are a composition of spatial functions and queries executed sequentially through well-planned steps. These functions provide suitable result to determine the ideal locations for location planning (popularly known as location analysis) or carrying out hypothetical testing (statistical analysis) within a certain area.
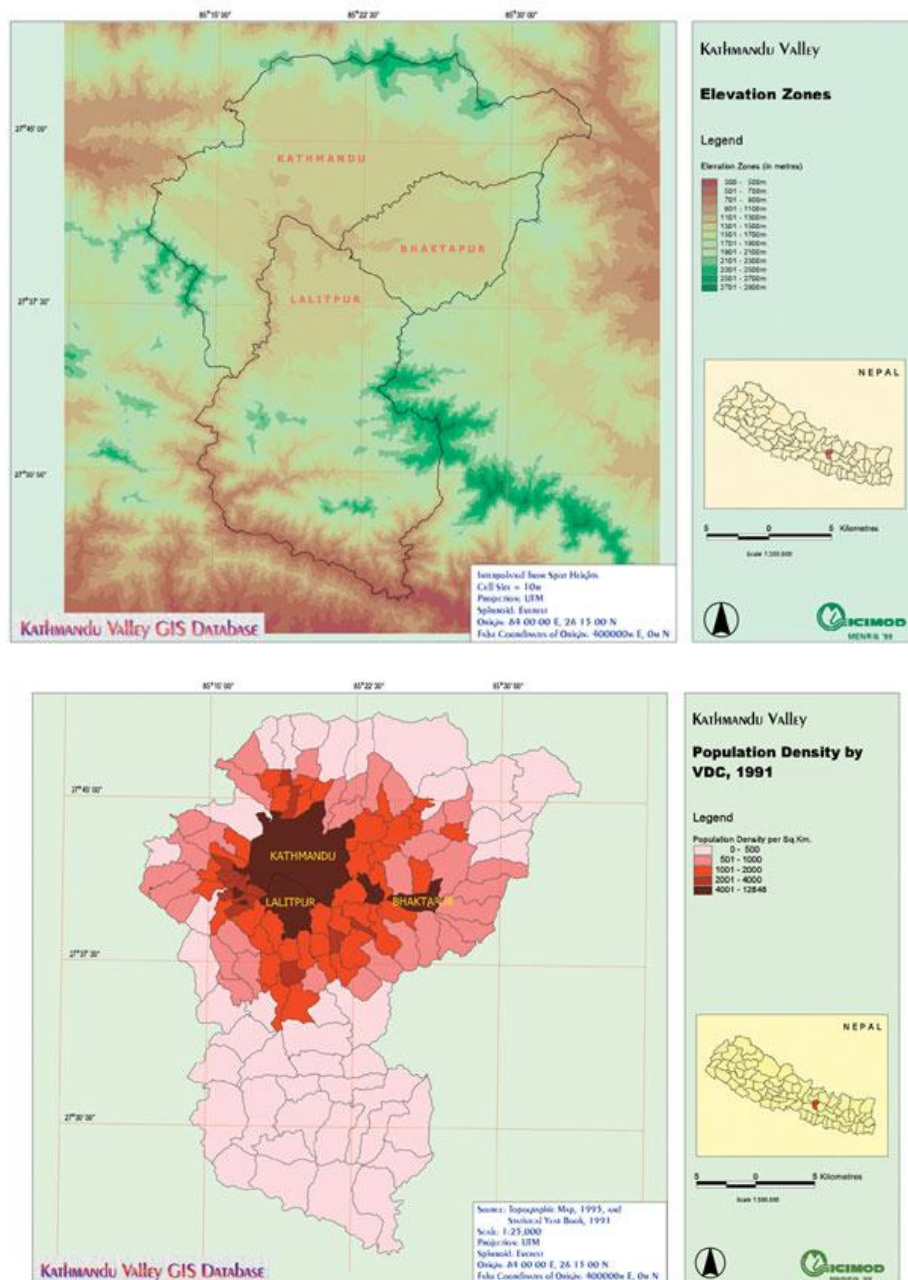
Figure 3.4: Examples of Thematic Maps

Geospatial analyses are divided into two main types of analyses which are related to the kind of data, vector or raster. The set of vector geospatial analyses are composed of the geospatial analyses, the geospatial queries, the geospatial measurement and the geospatial relationship. Every geospatial analysis is able to use the others types of analyses in order to perform its own process. More details will be given in the section 3.2. The next section presents the geospatial data with more details and the following section deals with geospatial analysis.

After what details are given about geospatial database systems. The last section concludes this chapter.

3.1. Geospatial Data

This section describes the geospatial data and focus more specifically on the vector geospatial data. One of the main reason behind focusing on vector representation is it simplifies the representation of the objects geometries which is necessary during knowledge management process of these geometries. This section is composed of a description of vector representation, a description of raster representation, and a discussion compared to the purpose of the thesis.

3.1.1 Vector Representation

In GIS domain, the term vector refers to data that are comprised of lines or arcs defined by beginning and end points which meet at nodes (Smith, Goodchild M., & Longley, 2007). The coordinates and the topological structure of these nodes are generally stored externally mostly in a database system or vendor proprietary format. The GIS handling vector data are generally termed as Vector GIS. In other words, a vector GIS is defined by the vectorial representation of its geographic data (Ellis, Vector based GIS, 2001). The vector representation of maps uses geometrical primitives such as points, lines, curves and polygons to represent features. In most cases, they are based on some mathematical equations. Since, they use geometrical primitives; operations like zooming or stretching do not affect the quality of the representation. Vector data are resolution independent and can be scaled in any size without losing details or clarity. So, there is no impact on the quality by vector operations such as zoom, pan, rotate, rescale, and translation. Hence, this representation is the most appropriate to define maps. In general, every vector GIS is thematic based so a database handling such system organizes two data types (spatial and thematic). The spatial analyses which are performed on vector data are vector analysis. There is huge collection of spatial operations which could be performed on vector data. In general, all the modern database systems have the spatial extensions with spatial operations designed for vector data. The figure 3.5 demonstrates the vector based representation in current GIS. The basic representation of objects in vector GIS is through the primitive features such as points, arcs or polygons. All these features are made up of a series of coordinates which represents the

geometries within them. The coordinates are stored as spatial data type in the proprietary formats of GIS software or in the database systems. It could be seen in figure 3.5, there are various objects labeled with different primitive features: polygons, lines and points. The polygons are constructed through joining various edges. These edges are the nodes which contains the coordinates of that edge. Thus by connecting those nodes (with set of coordinates) represents the geometry of the object. Similarly, the arcs are constructed through joining edges but the starting edge does have only one connection thus leaving the connections open. The points are nodes with no connections to other nodes.
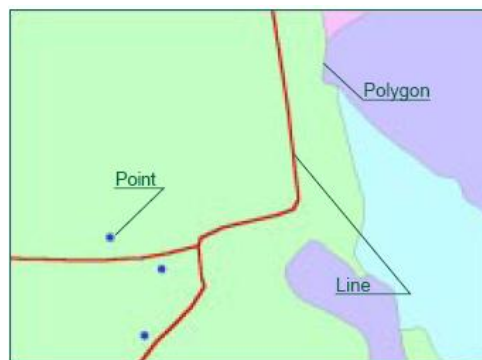
Figure 3.5: A vector representation of a real world scenario

3.1.2   Raster Representation

A Raster representation divides a geographic area into cells which could be identified through their rows and columns. The sizes of the cells determine the resolution of the representation and depends the requirements of the users. It is not necessary to explicitly code the geographic coordinates as they are implicit to the cells. In addition, the topology is also encoded implicitly within the cells. These coordinates are determine through the row and column numbers and hence through these numbers the location of an object could be determined. The cells are generally encoded with a value which represents the geographic or the nominal value like the rainfall value or elevation value of the cell representing the location. Figure 3.6 illustrates how a geographical area could be represented in grid cell.
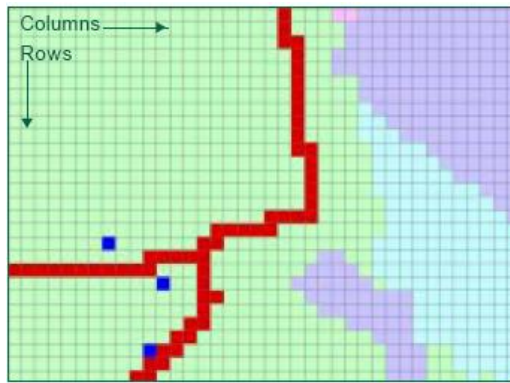
Figure 3.6: Raster representation

In raster GIS the cell can occupy only one discrete value and hence the multiple attributed maps should be broken down into a series of raster maps to represent each attribute. For example if a GIS vector map is build up with attribute data of height, population, rainfall value, then such a map is broken into raster maps representing the height, population and rainfall value individually. In simple term raster maps are one attribute maps. Since the data are stored in grid structure, raster map provides quantitative analysis techniques through map algebra. The applications of map algebra within the raster map have provided functionalities to analyse the data mathematically. It is basically useful to analyse the scenario over the time. An example would be to evaluate the land use/cover change over a certain period of time. The analysis is carried out overlaying land use/cover data of different periods. Then a simple algebraic calculation is carried out to determine where the land has changed. The figure 3.7 illustrates this with an example.

1	1	1	1
1	3	1	1
3	3	3	1
3	2	2	2
3	1	4	2
2	1	4	4
1	1	1	4

(a)

1	1	1	1
3	3	2	2
3	3	3	1
3	1	1	1
3	1	1	2
4	4	4	4
1	4	4	4

(b)

101	101	101	101
103	303	102	102
303	303	303	101
303	201	201	201
303	101	401	202
204	104	404	404
101	104	104	404

(c)

Figure 3.7: Map Calculation to determine Land use/cover change a) Land use/cover 1975 b) Land use/cover 2002 c) Land use/cover Change

The figure 3.7 contains two dataset of two periods 1975 and 2002. The land use/cover needs to be classified with the numeric code. For example 1 stands for vegetation, 2 for forest, 3 for agriculture and 4 for wasteland. A simple arithmetic operation (a * 100 + b) is applied in the datasets which returns a new dataset as projected in figure 3.7 c. This dataset provides the

information where land use/cover has been changed over the year. The cells where the first and last digits are same are the sections where the land use/cover have not changed and the cells where they have different values are the sections where the land use/cover have changed. It is also possible to determine how the land use/cover has changed. The first value of the resultant cell is the land use/cover type of 1975 and the last value is that of 2002. So the cell could be classified as the land change from the type in 1975 to that of 2002. The dataset can be thus classified accordingly. For example the cells containing 104 points out that the section of land has been changed from vegetation to wasteland. Such map algebraic operations provide impression on the nature of change and are useful tool to analysis in areas such as deforestation or human encroachment.

3.1.3   Vector vs. Raster

In the last two sections, the data representation formats that most of the current GIS use to present the data have been discussed. Both representation formats have their own benefits and limitations. This section underlines the benefits and limitations of these representation formats paving a way to the direction of the research thesis.

Representation format	Advantage	Disadvantage
Vector	- Resolution: Original - Appearance: Aesthetically pleasing - Topology: Preserved and could be used explicitly - Originality: Original as data created originally in vector format - Multiple attribute possible - Core in spatial Database technology: Spatial Data type - Spatial Operation and Functions	- Storage: Each vertex needs to be stored - Continuous data not represented effectively - Map Algebra: not possible
Raster	- No explicit storage of information - Data analysis easy to formulate due to the nature of data (algebraic calculation) - Ideal for continuous data so most often used in Digital Elevation Model (DEM) - Ideal for statistical analysis	- Storage: Coordinates represented as rows and column - Only one attribute could be used at one time - Resolution: depends on cell size - Appearance: grid structure so not pleasant

	• Topology: Implicit so not used often
	• Originality: Needs the conversion
	• Certain database system provides storage of raster data but in complex structure

Table 3.1: The general comparison between vector and raster representation

Table 3.1 outlines the general comparison between vector and raster representation. Though raster data provides the capability of map algebra through algebraic operations they are more or less independent to majority of spatial operations provided by the modern GIS. This research thesis utilizes the advancement in spatial technology in database system which concentrates more on vector data. The raster data has been included in spatial extensions of major database systems like PostGIS (Racine, 2010) or Oracle 11g (Xie & Sharma, 2007), but they are in early stages and is not matured enough to provide sophisticated operations as they do for vector dataset.

This research thesis establishes its principle based on the existing spatial technology within database systems which is primarily based on vector GIS. The spatial operations and functions provided in the research work are thus used for the vector dataset. Moreover ArchaeoKM stores its geometries as vector dataset in PostGIS extension of PostgreSQL and thus it becomes logical to implement spatial operations based on vector dataset in this case. The development in raster dataset however might provide an opportunity to extend the raster dataset integration in later future.

3.2 Geospatial Analysis

The geospatial analyses or more popularly spatial analyses are carried out on the geometries of the spatial data. The emphasis of a spatial analysis is the measurement of feature's properties and relationships. Thus, the analysis takes into account the localization of the phenomenon studied in a direct way (Camara, Monteiro, Fucks, & Carvalho). In other words:

> *Spatial analysis is a set of methods whose results change when the locations of the objects being analyzed change (Longley, Goodchild, Maguire, & Rhind, 2001).*

The spatial analysis process is one of the most important tools in a GIS database. Actually, it permits to interact with data in order to answer numerous questions materialized by spatial queries. Consequently, these answers support decisions or even reveal spatial patterns. Basically, a spatial analysis is performed through various spatial operations and functions provided by the GIS software's packages. These operations and functions are mostly applied to vector geometries.

These operations and functions can be categorized according to their nature. Four main categories have been identified for this research work. They are geospatial queries, geospatial measurement, and geospatial processing and geospatial relationships.

3.2.1 Geospatial Queries

These basic operations are carried out similarly to Structured Query Languages (SQL) queries in a relational database system. The difference is characterized by the spatial components included within these queries. The spatial operations included into the queries take geometries into account to process the geometries and the result of this process is combined with other non-spatial data to return the results. The data structure remains the same like any relational database system. For instance, *"What are countries that have area of more than 400 thousands square meters and population less than half a million?"* and the operations to answer the question could be:

```
SELECT Name
FROM Country
WHERE Area (boundary)> 400000 and Country < 500000
```
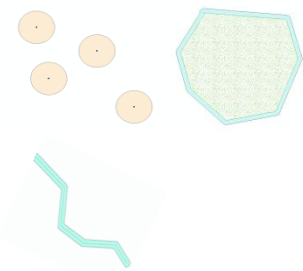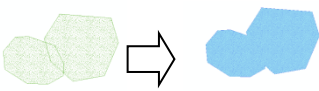
The question incorporates both spatial and attributive data. In this example geometry is stored in a separate column as spatial data type. The boundaries of every country are stored with their coordinates in the column *boundary* which is of type *geometry*. The query *Area (boundary)* function which is combined with the query first process the spatial function Area to derive the areas of the countries. Then it process the standard SQL operation first checking the countries with area more than 400000 and population less than 500000 to return the result.

3.2.2 Geospatial Measurement

These measurements are simple functions that return the spatial properties of objects through geometrical calculus. Examples of such functions are "area", "perimeter", "length", "shape", etc. Some functions within this category return the measurement relationship of two objects. For instance, the distance or the direction between two objects could be retrieved.

3.2.3 Geospatial Processing

The functions and operations under this category change the datasets either by combining the geometries of two or more objects or by processing an object to return the geometry. In either case, the functions return a geometrical shape. Although there is wide range of spatial functions lying under this category, this section covers the functions that are widely used and could be found in almost all other database systems. The table 2 highlights these functions.

Spatial functions and operations	Descriptions	Figure examples
Buffer	The buffer operation generates a new geometry that extends outward or inward from an existing geometry by a specified radius. The function could be applied to every type of geometries. In a GIS system the Buffer operation returns geometries of vector nodes of the feature whose distances are equal to the buffer distance and draws a polygon from those set of points.	
Union	The union function returns the union of two geometries. The operation is an equivalent of the logical operator OR. This function could also be applied to every feature types or the combinations of at least two distinct features but it is distinctly visible with polygons. $$x \in A \bigcup B = \{x \in A \vee x \in B\}$$ $$A \cup B \rightarrow P(A) + P(B) - P(A \cap B)$$ Here P is the probability index.	

Intersection	The intersection function returns the intersecting geometry of the geometries of two spatial objects. It could be thus said that this function is spatial equivalent of the logical operator AND. Again, this function could be applied to all the feature types or the combinations of two or more distinct features types. $$x \in A \bigcap B = \{x \in A \wedge x \in B\}$$	
Difference	The difference function is a spatial extension of logical operator AND NOT. It returns the geometry that is not the intersection of two features. $$x \in A \setminus B = \{x \in A \wedge x \notin B\}$$	

Table 3.2: Different spatial functions that belong to the category of geospatial processing

The spatial analysis with vector data involves the data to be manipulated through taking the neighborhood data into account. The spatial functions under this category could process the data by either combining or subtracting the surrounding objects to generate a new set of data itself which could be important to perform certain analysis. Besides, the buffer function provides a wide range of analysis through its capacity of creating an area within a feature. This buffer zone serves the analysis process in numerous ways. The buffer function is primarily used in proximity analysis which considers locations around a feature by measuring distance around it. It can be used to determine relationships between other features within the area. It is hence mostly a part of proximity analysis.

3.2.4   Geospatial Relationships

The functions that belong to this category are binary functions. These functions return a Boolean value i.e., true or false when executed. These functions are very important in spatial analysis as they help to determine the result of an analysis process. There are quite a few spatial relation functions that a GIS provides. These functions are shown in table 3.3.

Functions	Descriptions	Figure examples
Disjoint	The geometries have no any part in common. $A \cap B = \phi$	
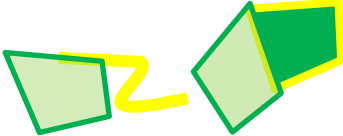
Equals	The geometries are spatially equal. That means the coordinates of the features are same.	
Touches	The geometries touch each other. That means the exteriors of the geometries can touch each other but none of the point common to the interior of both geometries intersects each other.	
Within	The first geometry is completely within the second geometry. The interior of first geometry must intersect completely with second but the intersection of exterior of first geometry with second not possible.	
Overlaps	Geometries are overlap when the intersection of the geometries results geometry which is different from the parent geometries.	

Table 3.3: Different functions that belongs to the geospatial relationships analysis

The functions that belong to this category analyse the relationship through determining the topological relationship of the geometries. They are crucial in any spatial analysis process as they could process the geometries through their topological relationships and generate the result based on these relationships.

3.3 Spatial Database Systems

Today, every mainstream database system includes spatial components within them to support the growing need of addressing spatial components within the Information Systems. Over the past decade, as RDBMS has seen a huge growth in the database technology, the spatial components within them also seen a tremendous improvement in their functionalities. In early days, spatial data were organized in dual architectures which consist of separate administrative data for data management in a RDBMS and spatial data for a GIS system. This could easily result in data inconsistency. Today all the database systems maintain the spatial component in a single RDBMS. These database systems implement the spatial information as spatial data types and spatial operators which follow the Open Geospatial Consortium (OGC 1998) Simple Features Specifications for SQL (OGC 1999)[9]. Since OGC Simple

---

[9] http://www.opengeospatial.org/standards

Feature Specifications are built within simple spatial features in 2D space, most of the spatial operations are restricted to 2D spatial data. However, storing, retrieving and visualizing 3D data is also possible. In addition, the current RDBMS system supports certain spatial queries which are possible in 3D domain as well.

3.3.1   Spatial Storage and Retrieval

Spatial information is stored as spatial data type within the current database systems. By default they store 2D spatial information of the objects however most of the mainstream DBMS support the storage of 3D objects (3Dpoint, 3Dline and 3Dpolygon). The spatial information is retrieved through the spatial queries built-ins within each database system. These queries could retrieve 3D data but the spatial functions provided by existing databases are limited to the 2D objects. This trend however is changing fast. Prior to the introduction of Oracle 11g, there were two popular concepts of storing the 3D objects which follows the similar path of representing a 3D object in the form of polyhedrons (Zlatanova, 2006): a list of polygons or as a multipolygon. In the first approach two tables are used to store the geometry – a table to store the body and the next one to store the face. This approach takes care of the topology as the bodies are defined with reference to the faces and the faces can be shared by the neighboring bodies. The second approach is straight forward – it stores the object in a single record as the multipolygon, which is supported by most of the DBMS supporting spatial features. The second approach is the most preferred one as it has one-to-one relation between the record and the object. Furthermore, the front end GIS systems find it easy to recognize as a single object than as a list of polygons in the previous case (Zlatanova, 2006). Though the mainstream databases follow the simple feature specification of OGC, each system stores the spatial data in its own format which varies from each other in certain degrees. This research thesis uses PostgreSQL and its spatial extension PostGIS, hence the thesis will be based on the spatial solutions provided by the database system.

PostGIS is the spatial extension of PostgreSQL object relational database system that allows the spatial objects to be stored in the database (PostGIS). A strong tendency has been seen in the last few years that big GIS vendors like GRASS and ESRI have shifted their support towards PostGIS. As with Oracle, PostGIS supports the storage of point, line, polygon, multipoint, multiline, multipolygon, and geometry collections. It follows the specification provided by OGC for the simple features to store these objects. Those are specified in the

Open GIS Well Know Text (WKT) or Well Known Binary (WKB) Formats[10]. From as early version as PostGIS 0.9, it supports all the objects and functions specified by OGC "Simple Features for SQL" specification. However PostGIS extends by supporting 3D and 4D objects. PostGIS has given those extensions names as EWKB or EWKT (Extended Well Known Binary and Extended Well Known Text). In contrast to the Simple Feature Specification by OGC, those extensions support the embedded SRID information. Hence it can be seen easily that PostGIS supports 2D features by following the Simple Feature Specifications of OGC and storing them in WKB or WKT and 3D features through EWKB or EWKT.

Inputs and outputs of these formats are available through the interfaces:

- 2D objects

```
bytea WKB = asBinary(geometry);
text WKT = asText(geometry);
geometry = GeomFromWKB(bytea WKB, SRID);
geometry = GeometryFromText(text WKT, SRID);
```

- 3D objects

```
bytea EWKB = asEWKB(geometry);
text EWKT = asEWKT(geometry);
geometry = GeomFromEWKB(bytea EWKB);
geometry = GeomFromEWKT(text EWKT);
```

PostGIS administers its user defined spatial tables through the system defined meta-data tables. It contains two meta-data tables: SPATIAL_REF_SYS and GEOMETRY_COLUMNS. As names imply, the SPATIAL_REF_SYS holds the numeric IDs and textual descriptions of the coordinate systems used in the spatial database and GEOMETRY_COLUMNS holds name of the tables and column with spatial data type.

---

[10]Well-known text (WKT) is a text markup language for representing vector geometry objects on a map, example: MULTIPOINT(3.5 5.6,4.8 10.5), A binary equivalent is known as well-known binary (WKB)

3.3.2 Spatial Functions and Operations

Spatial functions and operations are the queries performed on the spatial data set to generate the results which can be visually interpreted. It is a process that leads to the spatial analysis. The spatial operations take different spatial properties of an object like the topological, geometric or geographic properties for carrying out the spatial analysis. All major database systems support these functions and operations. They are basically coupled with SQL SELECT statement.

The general syntax of usage of these operations within SQL SELECT statement is similar no matter what the types of spatial operations are.

```
SELECT column_name(s), spatial_function(geometry column)
FROM   table_having_geometry_column
WHERE filter(s)
```

The syntax is an extension of SELECT statement syntax to adjust the spatial operations. There exist huge collection of spatial operations and functions in PostGIS libraries which can be used on spatial object. This thesis report lists out these functions and operations with reference to the functions presented in Vector Analysis section and are presented in table 3.4. The spatial functions are based on the spatial functions provided by PostGIS.

Category	Spatial Function/Operation	SQL Statement
Geospatial Queries	Area/Length	SELECT column_name FROM table WHERE ST_Area/ST_Length(geom) > value;
Geospatial Measurement	Area/Length/Distance	SELECT ST_Area/ST_Length(geom); SELECT ST_Distance ((_fromgeom), (_togeom));
Geospatial Processing	Buffer/Union/Intersection/ Difference	SELECT ST_Buffer(geom); SELECT ST_Union/ST_Intersection/ST_Difference(geom1),(geom2));
Geospatial Relationship	Disjoints/Equals/Touches/Within/Overlaps	SELECT ST_Disjoints/ST_Equals/ST_Touches/ST_Within/ST_Overlaps((geom1), (geom2));

Table 3.4: Spatial functions and operations in DBMS

The functions given in table 3.4 can be applied in different sequence within a SQL statement. Hence, it is possible to perform complex spatial analysis only through the database systems and do not have to rely on GIS.

3.4 Discussion

GIS technology has gained acceptance over the year analysis process where it involves objects with spatial identities. The analytic capabilities of the technology through the spatial operations and functions and through the map algebras in case of raster datasets have put it as a forth runner technology to support decision making. Hence, it is indispensable in any Decision Support Systems (DSS). The GIS systems range from tailored made GIS for executing specific task to the more general application tools. In any case the principle behind the technology remains the same. The reliability to geometry remains strong in all GISs. GIS has readily embraced technology shift. Whenever, there has been introduction of new relevant technology emerging, it has been integrated with GIS. It can be witnessed in the spatial technology in database systems. The evolvement of spatial technology in a database system has prompted the analysis process independent to vendor dependent GIS application software. This has also fuelled the development of custom based GISs. Furthermore, the data capturing and storing process has been simplified with the maturity of spatial technology in database system. However, with this maturation it has brought its own problems. With every corner of discipline being capable of acquiring and storing the spatial data in their own data structure, there is a serious question raised in the interoperability issue. The explosion of Information Technology and in particular the World Wide Web (WWW) has made the access of information global. It is unimaginable to work independent and with a specific system and structure. Information needs to be shared in order to add value to it. The emergence of Web Services in the field of spatial data has raised the concern of data interoperability for data sharing for the wider community.

Spatial data interoperability is a serious topic of research and has attracted huge interest all over the world. Traditionally, Spatial Data Infrastructure (SDI) has been used to describe the spatial data. In general, they are metadata of the spatial data and are used to exchange the data. Today most of the countries have developed their own national level SDIs. Hence, SDIs have become dependable method of data exchange. A SDI needs to follow OGC geospatial standard in order to have its effectiveness in sharing. This is defined through OGC geospatial

standards like Geography Markup Language (GML) which is basically a language to transform and exchange spatial data. GML works through its rich set of primitives which are used to define the nature of dataset which it is hosting. However, these primitives are defined to support underlying geometries and hence in every scenario semantics are ignored. Semantics define data and should be considered as one of the key issue to consider during designing a tool for data interoperability. The collaboration of semantic components in spatial data should be beneficial for understanding the data better and thus makes them easy to be utilized in different platforms. The semantically mapped spatial data are more flexible in their interpretations than the one which does not possess semantic support. This is because semantics can be mapped against their interpretations and participates in knowledge modeling.

The evolvement of the Semantic Web and its tools and techniques has provided the much needed background technology to address interoperability issues. The techniques presented in Semantic Web are fundamentally departure from the conventional one in terms of how the information is administered and processed. In addition to the interoperability issues these techniques provided new and improved analysis mechanism through knowledge interpretation. This approach is an intelligent approach and uses the knowledge modeled through the techniques to handle the complexity within the data.

Ontologies are the fundaments of the Semantic Web which represent the information in a formal machine readable format. In addition to the formal representation to the information, they provide intelligence to such representation. Ontologies are generally used to map the semantics of data hence providing the much needed semantic interpretation of the data. This property is exploited by the researchers to come out with a platform of interoperability. It has become easier to understand the meaning of data in a particular context as the data is mapped semantically with its related components. This semantic mapping to relate data to their components through relevant relationships provide semantic understanding to the data. This semantic understand then can distinguish what the components are similar to each other and what are different thus providing a base for semantic interoperability. Researches on Geospatial ontologies are among top researches that are currently undertaken within GIS community. Most of these researches are motivated by interoperability concerns (Cruz, Geospatial Data Integration, 2004), (Cruz, Sunna, & Chaudhry, Semi-Automatic Ontology Alignment for Geospatial Data Integration, 2004). The interoperability concerns shown by

the GIS community are real and should be addressed in proper way. The fact is with increasing adoption of Semantic Web technology and its techniques within geospatial community there are research works carried out to build the ontologies that express the common elements of diverse knowledge artifacts. They however focus on interoperability issues and do not support the knowledge interpretation through reasoning or inferring techniques provided by Semantic Web technology.

The direction that researches are undertaking is more one dimensional. The semantic indexing of the spatial data in consideration of interoperability alone can limit the window of research. Spatial knowledge interpretations can be achieved through the knowledge processing through proper channel. For this the geospatial ontology needs to store and manipulate the spatial data through reasoning and inference engines. Ontology engineering has not yet matured enough to consider into store spatial data or to perform complex spatial analysis. However, the expertise of database technology in spatial technology could be well collaborated with the Semantic Web technology to achieve knowledge interpretation on spatial data.

The next chapter discusses on the fundaments of the Semantic Web providing the concept on different component of the technology. The chapter provides the needed backgrounds behind for further works on spatial integration. It constitutes the information about the current state of art technology in Semantic Web technology.

References

Al-Feel, H., Koutb, M. A., & Suoror, H. (2009). Toward An Agreement on Semantic Web Architecture. *WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*(49), 806-810.

Antoniou, G., & Harmelen, F. v. (2003). Web Ontology Language: OWL. In *Handbook on Ontologies in Information Systems* (pp. 67 - 92). Springer Series.

Baader, F., & Nutt, W. (2002). Basic Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, & P. F. Patel-Schneider, *Description Logic Handbook* (pp. 47 - 100). Cambridge: Cambridge University Press.

Baader, F., & Sattler, U. (2000). An overview of tableau algorithms for description logics. *Springer-Verlag. 1847.* Scotland: LNAI.

Baader, F., & Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica*, 5 - 40.

Baader, F., Calvanese , D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2003). *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge: Cambridge University Press.

Baader, F., Horrocks, I., & Sattler, U. (2003). Description Logics as Ontology Languages for the Semantic Web. *Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence* .

Batcheller, J. K., & Reitsma, F. (2010). Implementing feature level semantics for spatial data discovery: Supporting the reuse of legacy data using open source components. *Computers, Environment and Urban Systems*, 333 -344.

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., et al. (2004, February 10). *OWL Web Ontology Language.* Retrieved November 27, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-ref/

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific AmericaN*, pp. 34-43.

Berry, J. K. (1999). GIS TECHNOLOGY IN ENVIRONMENTAL MANAGEMENT: a Brief History, Trends and Probable Future. In D. L. Soden, & B. S. Steel (Eds.), *Handbook of Global Environmental Policy and Administration* (pp. 49 - 81). Decker, Marcel Inc.

Bobillo, F., & Straccia, U. (2009). Supporting Fuzzy Rough Sets in Fuzzy Description Logics. *10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. 5590/2009*, pp. 676-687. Verona: LNCS.

Boley, H. (2009, August 11). *RuleML: The Rule Markup Initiative*. Retrieved November 27, 2009, from RuleMLL: http://ruleml.org/

Boley, H., & kifer, M. (2010, June 22). *W3C Recommendation* . Retrieved August 24, 2010, from RIF Basic Logic Dialect: http://www.w3.org/TR/rif-bld/

Brachman, R., Eugene, C., Norton, G., & Martin, Y. (1978). *KLONE Reference Manual.* Cambridge: Bolt Beranek And Newman Inc.

Burrough, P. A. (1986). *Principles of geographic information systems for land resources assessment* (Vol. 12). Clarendon Press (Oxford Oxfordshire and New York).

Calvanese, D., Giacomo, G. D., Lenzerini, M., & Nardi, D. (2001). Reasoning in expressive description logics. In A. Robinson, & A. Voronkov (Eds.), *Handbook of Automated Reasoning* (Vol. II, pp. 1581-1634). Elsevier Science Publishers.

Camara, G., Monteiro, A. M., Fucks, S. D., & Carvalho, M. S. (n.d.). *Spatial Analysis and GIS: A Primer.*

Cormode, G., & Krishnamurthy, B. (2008). *Key Differences between Web1.0 and Web2.* Florham Park: AT&T Labs-Research.

Cruz, I. F. (2004). *Geospatial Data Integration.* Chicago: ADVIS Lab, Dept. of Computer Science, University of Illinois.

Cruz, I. F., Sunna, W., & Chaudhry, A. (2004). Semi-Automatic Ontology Alignment for Geospatial Data Integration. *GIScience 2004* (pp. 51 - 66). MaryLand: LNCS.

da Silva, P. P., McGuinness, D. L., & Fikes, R. E. (2004). *A Proof Markup Language for Semantic Web Services.* Stanford: Stanford University.

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is Knowledge Representation. *AI Magazine, 14*(1), 17 - 33.

Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Erdmann, M., et al. (2000). The Semantic Web: the roles of XML and RDF. *IEEE Internet Computing* , 63-74.

Dempsey, C. (2010, January 03). *What is GIS?* Retrieved August 2010, 2010, from GIS Lounge: http://gislounge.com/what-is-gis/

Ellis, F. (2001, March 12). *Vector based GIS.* Retrieved November 27, 2009, from GIS self learning tool: http://www.sli.unimelb.edu.au/gisWeb/GISModule/GIST_Vector.htm

ESRI. (2005, July). GIS Topology. *An ESRI White Paper*. New York: ESRI.

ESRI. (n.d.). *GIS.com: the Guide to Geographic Information Systems* . Retrieved August 4, 2010, from What is GIS: http://www.gis.com/content/what-gis

Fensel, D., Horrocks, I., Harmelen, F. v., McGuinnes, D. L., & Patel-Schneider, P. (2001). OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 38 - 45.

Gerber, A., Merwe, A., & Barnard, A. (2008). A functional Semantic Web architecture. *Proceedings of the 5th European Semantic Web conference on The Semantic Web: research and applications* (pp. 273 -287). Tenerife: Springer-Verlag.

Goebel, R., Zilles, S., Ringlstetter, C., Dengel, A., & Grimnes, G. (2008). What is the Role of Semantic Layer Cake for Guiding the Use of Knowledge Representation and Machine Learning in the Development of the Semantic Web? *Fourth Artificial Intelligence for Interactive Digital Entertainment Conference.* Chicago: AAAI 2008.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acqusition, 5*(2), 199 - 220.

Harlmelen, F. v., Patel-Schneider, P., & Horrocks, I. (2001, March 27). *A model-theoretic semantic for DAML+OIL.* Retrieved August 30, 2010, from http://www.daml.org/2001/03/model-theoretic-semantics.html

Harris, S., & Seaborne, A. (2010, January 26). *SPARQL Query Language 1.1.* Retrieved September 2, 2010, from W3C Working Draft: http://www.w3.org/TR/2010/WD-sparql11-query-20100126/

Hayes, P., & McBride, B. (2004, February 10). *RDF Semantics.* Retrieved August 26, 2010, from W3C Recommendation: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

Hellerstein, J. M., Naughton, J. F., & Preffer, A. (1995). Generalized Search Trees for Database Systems. *Proceedings of the 21th International Conference on Very Large Data Bases* (pp. 562 - 573). San Francisco: Morgan Kaufmann Publishers Inc.

Hendler, J., & McGuinness, D. L. (2000). The darpa agent markup language. *IEEE Intelligent System, 15*(6), 67 -73.

Herman, I., Halpin, H., Hawke, S., Prud'hommeaux, E., Raggett, D., & Swick, R. (2010, February 06). *W3C Semantic Web .* Retrieved August 15, 2010, from W3C Semantic Web Activity: http://www.w3.org/2001/sw/

Horrocks, I., Parsia, B., Patel-Schneider, P., & Hendler, J. (2005). Semantic Web Architecture: Stack or Two Towers. *3rd Internation Workshop on Principles and Practice of Semantic Web Reasoning* (pp. 37 - 41). Dagstuhl: LNCS .

Horrocks, I., Pater-Schneider, P. F., McGuinness, D. L., & Welty, C. A. (n.d.). *OWL: a Description Logic Based Ontology Language for the Semantic Web.*

Horrocks, I., Schneider, P. f., Boley, H., Tabelt, S., Grosof, B., & Dean, M. (2004, December 21). *SWRL - A Semantic Web Rule Language - Combining OWL and RuleML.* Retrieved May 22, 2009, from SWRL: A Semantic Web Rule Language - Combining OWL and RuleML: http://www.w3.org/Submission/SWRL/

James Madison University. (2004, March 31). *Spatial Information Clearinghouse.* Retrieved October 18, 2010, from Components of a GIS: http://maic.jmu.edu/sic/gis/components.htm

Jurisica, I., Mylopoulos, J., & Yu, E. (1999). Using Ontologies for Knowledge Management: An Information Systems Perspective. *Annual Conference of the American Society for Information Science.* Washington: American Society for Information Science.

Klyne, G., Caroll, J. J., & McBride, B. (2004, February 10). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* Retrieved August 26, 2010, from W3C Recommendation: http://www.w3.org/TR/rdf-concepts/

Lassila, O., & Swick, R. (1999, February 22). *Resource Description Framework (RDF) Model and Syntax Specification.* Retrieved August 29, 2010, from W3C Recommendation: p://www.w3.org/TR/REC-rdf-syntax/

Lieberman, J. (2007, June 23). *Geospatial Incubator Group.* Retrieved July 2010, 14, from W3C Incubator Activity: http://www.w3.org/2005/Incubator/geo/charter

Lieberman, J., Singh, R., & Goad, C. (2007). *W3C Geospatial Ontologies - W3C Incubator Report.* W3C Incubator Group.

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2001). *Geographic Information Systems and Science.* London: Wiley.

Mallery, J. (1994). A Common LISP hypermedia server. *Proc. First International World-Wide Web Conference*, (pp. 239 - 247). Geneva.

Mazzocchi, S. (2000). Toward the Semantic Web: a view of XML from outer space. *Proceeding: ApacheCON 2000.* London: Apache.

McGuinness, D. L. (2001). *Description Logics Emerge from Ivory Towers.* Standford: Standford University.

McGuinness, D. L., & Harmelen, F. v. (2004, February 10). *OWL Web Ontologgy Language: Overview.* Retrieved December 2, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-features/

McGuinness, D. L., & Harmelen, F. v. (2004, February 10). *OWL Web Ontology Language: Overview.* Retrieved July 2010, 12, from W3C Recommendation: http://www.w3.org/TR/2004/REC-owl-features-20040210/

Patel-Schneider, P. F., Hayes, P., & Horrocks, I. (2004, February 10). *OWL Web Ontology Language.* Retrieved July 2010, 12, from W3C Recommendation: http://www.w3.org/TR/owl-semantics/

PostGIS. (n.d.). *PostGIS documentation.* Retrieved August 11, 2010, from http://www.postgis.org/documentation/

Prud'hommeaux, E., & Seaborne, A. (2008, January 2008). *SPARQL Query Language for RDF.* Retrieved May 22, 2010, from http://www.w3.org/TR/rdf-sparql-query/

Racine, P. (2010). Introducing PostGIS WKT Raster: Seamless Raster/Vector Operations in a Spatial Database. *FOSS4G 2010.* Barcelona: FOSS4G 2010.

Reisterer, J. (2008, April 7). *UTM - Universal Transverse Mercator Geographic Coordinate System.* (S. Hughes, D. Narsavage, & D. Boyack, Editors) Retrieved October 7, 2010, from Geospatial Training and Analysis Cooperative.

Reitsma, F., & Hiramatsu, K. (2006). Exploring GeoMarkup on the Semantic Web. *Proceedings:9th AGILE Conference on Geographic Information Science* (pp. 110 - 118). Visegrád: 9th AGILE Conference on Geographic Information Science.

Schmidt-Schauß, M., & Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence, 48*, 1 - 26.

Shannon, V. (2006, May 23). *The New York Times - Technology.* Retrieved August 16, 2010, from A 'more revolutionary' Web: http://www.nytimes.com/2006/05/23/technology/23iht-Web.html

Simpson, S. G. (1998). *Math 563: Model Theory.* Pennsylvania: Pennsylvania State University.

Smith, M. J., Goodchild M., F., & Longley, P. A. (2007). *Geospatial Analysis: A Comprehensive guide to Principles. Techniques and Software Tools.* Metador.

Smith, M. K., Welty, C., & McGuinness, D. L. (2004, February 10). *OWL Web Ontology Language.* Retrieved November 17, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-guide/

Smith, T. R., Menon, S., Starr, J. L., & Estes, J. E. (13 - 31). Requirements and principles for the implementation and construction of large-scale geographic information systems. *International Journal of Geographical Information Systems*, 1987.

Spaccapietra, S., Cullot, N., Parent, C., & Vangenot, C. (2004). On Spatial Ontologies. *GEOINFO* . Campos do Jordao.

Stanley, A. (1989). *Geographic information systems: a management perspective.* Ottawa: WDL Publications.

Sterling, L., & Shapiro, S. (1994). *The Art of Prolog.* MIT Press.

Stojanovi , N., & Handschuh, S. (2002). A Framework for Knowledge Management on the Semantic Web. *Poster - the 11th International World Wide Web Conference.* Hawaii.

Tanasescu, V., Gugliotta, A., Domingue, J., Gutiérrez-villarías, L., Davis, R., Rowlatt, M., et al. (2006). A Semantic Web Services GIS based Emergency Management Application . *The 5th International Semantic Web Conference (ISWC06).* Athens, Georgia: ISWC06.

Welty, C. (1996). Intelligent assistance for navigating the Web. *Proc of the 1996 Florida AI Research Symposium* . Florida: AAAI Press/MIT Press.

Welty, C., & Jenkins, J. (2000). Untangle: a new ontology for card catalog systems. *Proc of 17th National Conference on Artificial Intelligence (AAAI 2000)* (pp. 1137 - 1138). Austin: The MIT Press.

Xie, Q. (., & Sharma, J. (2007). *Oracle Spatial 11g: GeoRaster.* Redwood Shores: Oracle.

Zlatanova, S. (2006). 3D geometries in spatial DBMS. *Proceedings of the International Workshop of 3D Geoinformation.*

Chapter 4

THE SEMANTIC WEB

This chapter presents a general overview on the Semantic Web technologies and the different components behind. It also lays out the specific areas of semantic definitions and languages with the general overview on the technologies of the Semantic Web. The description logics are the core of the semantic definition and a large part of this chapter is devoted to the formalization of the Description Logics and its Semantic Web counterpart OWL. In addition, SPARQL and SWRL, which are respectively a query and a rule language, are the key technologies which will allow queries and inferences on spatial data once these languages are updated. This chapter focuses on principles required to update these languages.

The World Wide Web (WWW or the Web) is the single largest repository of information. The growth of Web has been tremendous since its evolvement both in terms of the content and the technology. The first generation Webs were mainly presentation based. They provided information through the Web pages but did not allow users to interact with them. In short they contained read only information. Moreover, the early pages were text only pages and do not contain multimedia data. These Web sites have higher dependency on the presentation languages as Hypertext Markup Languages (HTML) (Horrocks, Pater-Schneider, McGuinness, & Welty, 2004). With the introduction of e**X**tensible **M**arkup **L**anguage (XML), the information within the pages became more structured. Those XML based pages could hold up the contents in more structured method but still lack the proper definition of semantics within the contents (Berners-Lee T. , 1998). Needs of intelligent systems which could exploit the wide range of information available within the Web are widely felt. Semantic Web is envisaged to address the need. The term "Semantic Web" is coined by Tim Berners-Lee in his work (Berners-Lee, Hendler, & Lassila, 2001) to propose the inclusion of semantic for better enabling machine-people cooperation for handling the huge information that exists in the Web.

The term "Semantic Web" has been defined numerous time. Though there is no formal definition of Semantic Web, some of its most used definitions are

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is a source to retrieve information from the Web (using the Web spiders from RDF files) and access the data through Semantic Web Agents or Semantic Web Services. Simply Semantic Web is data about data or metadata* (Berners-Lee, Hendler, & Lassila, 2001).

> *A Semantic Web is a Web where the focus is placed on the **meaning** of words, rather than on the words themselves: information becomes knowledge after semantic analysis is performed. For this reason, a Semantic Web is a network of **knowledge** compared with what we have today that can be defined as a network of information* (Mazzocchi, 2000).

*The **Semantic Web** provides a common framework that allows data to be shared and reused across application, enterprise and community boundaries* (Herman, Halpin, Hawke, Prud'hommeaux, Raggett, & Swick, 2010).

This section covers different features of Semantic Web. This includes the tool and techniques involved within a Semantic Web activity.

Any information systems which have to interoperate with various other information systems have to face the problem of interoperability. The archaeological community has seen the tremendous change in the manner the data are collected and manipulated. In one hand the technology growth provides the added functionalities to handle information which archaeologists cherish but at the same time they provide heterogeneity in the information pattern. The differences in manners and methods of individual community with the archaeological domain have led to development of independent systems and this has contributed in data incompatibility. A platform providing interoperability between different systems and in particular different sets of information has been widely felt within the community. Actually, the data heterogeneity is the main issue when the time comes to exchange and to manage information that describe the real world.

The issue of interoperability has always been there in the field of Information Technology ever since the computer systems started to communicate with each other through various modes. Factors like data authority, system autonomy and data heterogeneity are involved in the concerns of achieving efficient interoperability among different information systems. During the initial stages of the technology when a system was restricted to a department or at most a company, the issue of interoperability was limited within departments of a company. Hence the concern of data authority was not a big issue. However, the involvement of different departments and with them different players raised the issue of data heterogeneity. The evolvement of database management system (DBMS) fuelled up the necessity for data interoperability. Different underlying issues needed to be considered for achieving data interoperability in database systems like the structural differences, constraints differences or the difference in query languages. These information systems are based on DBMSs and hence the efficiency of system interoperability depended on tackling the question of heterogeneity of underlying data models of these DBMSs. As data models are represented through their schemas, the most common approach was to compare the schemas of the DBMSs and convert

a schema of a DBMS to the next DBMS. Other approaches like building up a common model which acts as a broker to interchange the data between different DBMSs were also preferred to achieve the interoperability. In short, the first generation problem of data interoperability was mainly due to the fact of the differences in technical issues such as structures, constraints and different techniques. These problems are short term problems as they could be sorted out with a broker technology mediating between different technical approaches. The main problem of interoperability arises when there is a difference in understanding. The semantic differences between information fuel up the interoperability issues as the information gets more accessible and easy to use.

The next generation of systems saw gradual acceleration in the data types which are not necessarily structured. Those kinds of data could be semi-structured data or digital data like multimedia data. During this period data like geospatial data or temporal data got more acceptances within structured data community expanding the horizon of structured data. The influx of tailored made software applications for these kinds of data has raised the arguments of interoperability in much stronger manner. To add this there is the rapid growth of Internet technology and rapid growth in tendency to depend on internet for information. The information is thus distributed through various systems with their independent methods of developments and presentations. The issue of interoperability revolved around factors like technology for dealing heterogeneous systems with different data structures and patterns or handling the semantic interoperability through handling the difference in terminologies (Sheth, 1999). The necessity to have a common understanding of the information led to the concept of inclusion of some form of semantics to represent them. Metadata provided the semantic representations to the information. Metadata is data of data which provides information about the data in terms of their creation, storage, management, authority and in certain term their intended purpose. Metadata became essential part of any reliable information source and a medium to maintain interoperability. Likewise the trend to have standardization or adoption of ad hoc standards made significant progress towards achieving system, syntactic and structural interoperability (Sheth, 1999).

The current generation has followed the previous trend of heterogeneity in the data source and has carried it even further. The users have become more sophisticated in using this information. They expect the system to help them not at the data level but at the information and increasingly knowledge level (Sheth, 1999), thus expecting to have interoperability at the

semantic level. Though metadata provides certain level of semantics for the data, they are generally not enough for managing the ever exploding information. The contexts of information needs to be taken into account to understand the information and these contexts are managed through the ontologies as traditionally they are built for specifying the vocabularies and their relationships. The underlying semantics in ontology provides foundation to interpret the knowledge within. This has provided a huge boosting achieving interoperability between systems. The use of knowledge to understand information between systems and find a common linkage between them provides a framework for the interoperation. The issue of interoperability which started with technical difference has come to difference in understanding. The technical difference in dealing with interoperability is long been exercised but the semantic difference has come in a big way. It became even bigger issue with the amount of information that is available today. The problem could be tackled with resolving the difference in understanding of information. So a form of semantic mapping can address such issues of understanding.

The Web followed this evolution. Traditionally, the resources within the Web are built to represent information and not to access the semantics and hence do not consider the contents behind them. The early day of Web as popularly termed Web 1.0 basically represent information for visualizations. The information could not be processed in any way as they were presented through technologies like Hypertext Mark-up Languages (HTML). They were Read-only Web. With emergence of Web 2.0, the Web technologies got more matured and this allowed limited interaction with the Web. Technologies such as Asynchronous JavaScript and XML (AJAX)have allowed users to share their contents (Cormode & Krishnamurthy, 2008). They are in a way more Read-Write technology. The social networks like Facebook[11] or Twitter[12] or the Webcasting sites like YouTube[13] are some of the examples developed with Web 2.0 technologies. The evolvement of Web is shown in figure 4.1. This thesis report does not intend to go beyond discussing Web 3.0 as they are mostly hypothetical technologies.

---

[11]http://www.facebook.com
[12]http://www.twitter.com
[13]http://www.youtube.com
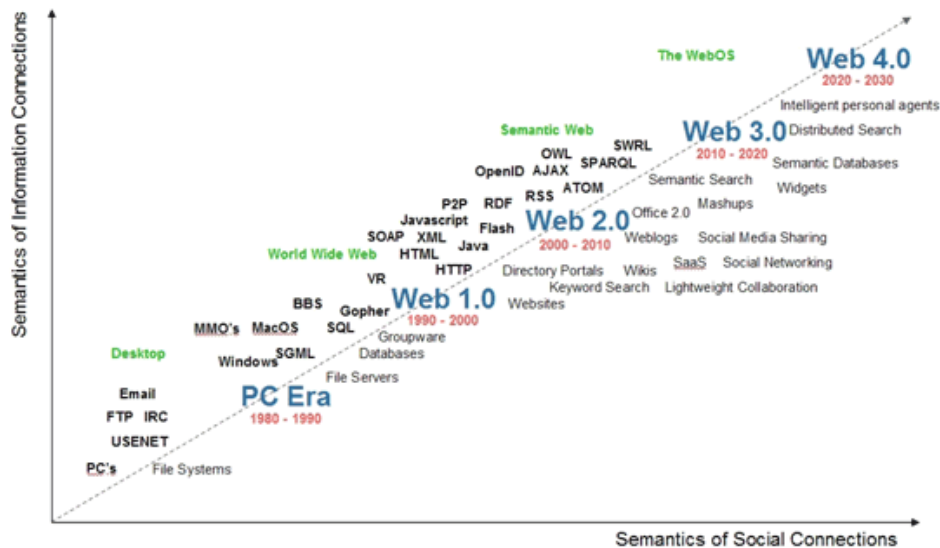
Figure 4.1: The evolution of WWW (source: Radar Networks & Nova Spivak 2007 - www.radarnetwork.com)

Both Web 1.0 and Web 2.0 do not answer the problem of interoperability stated in previous paragraphs as both of them rely heavily on the presentational technologies which describes syntax and not the semantics. Though Web 2.0 uses content based technology through XML in AJAX based systems, they are not enough to process the vast amount of information in the Web and they are not designed primarily to hold semantic. However, the XML technology has laid foundation for sophisticated technologies that can express semantics from the information.

Web 3.0 aims to make computers understand semantics behind information. This would make them intelligent to process information and deliver the required knowledge. It could be argued that the information when encapsulated by semantics would provide knowledge. The relationship between Web 3.0 and Semantic Web is a topic of argument. There are suggestions that they are the same whereas some argue that Semantic Web is a sub-set of Web 3. Sir Tim Berners-Lee has described Semantic Web as a component of Web 3.

> *"People keep asking what Web 3.0 is. I think maybe when you've got an overlay of scalable vector graphics - everything rippling and folding and looking misty - on Web 2.0 and access to a Semantic Web integrated across a huge space of data, you'll have access to an unbelievable data resource"*

> - Tim Berners-Lee, 2006, (Shannon, 2006)

This chapter covers different features of the Semantic Web. The first part consists of discussions on semantic definition languages and focuses on the Description Logics (DLs). As DLs are always considered as dominant components in any knowledge technologies, it becomes essential to have an overview on them. In the second chapter, the tools and techniques behind the Semantic Web are discussed through the foundational technology. Lastly, the chapter concludes through a discussion on relevancy of this research thesis and how does it fit in the overall scenario of the technology.

4.1. Semantic definitions

This section discusses different issues related to the semantic definition. It begins with the semantic graphs, then presents the Description Logic and finally presents Description Logic languages.

4.1.1. The Semantic Graphs

A semantic graph or a semantic network is a graphical representation of the real world. The real world scenarios are represented through the nodes and links in the graphs. Objects are represented through nodes and their relations are represented as links. Links are directed and labeled and hence semantic graphs are directed networks. In general nodes are represented through circles or boxes and links are arrows connecting those circles and boxes. Besides the objects and their relationships semantic graphs represent the instances of the objects and logics behind the representations. The links are generally used to associate either the relationships between the nodes or the types of the nodes. When they are used as relationships, they assert the nature of the relationships between two nodes and when they are used as a type of the node, they define the information of the node itself and not its relationship.

Every semantic graph is a declarative graphic representation that can be used either to represent knowledge or to support automated systems for reasoning the knowledge (Sowa, 2006). Semantic graph has been traditionally used to represent knowledge in the discipline of Artificial Intelligence (AI). Traditionally semantic graphs are used in the discipline of linguistic to model the grammatical structure a language. Semantic Web has benefited from the concept of semantic graph and hence outlined its technologies based on the graphical structure. Hence it becomes relevant to have understanding of how these graphs are

constructed. An example of semantic graph to represent a section of house is illustrated in the figure 4.2. Through this graph one gets the general knowledge about the house.

There are few components one needs to take into account. As already mentioned the links represent various purposes in a semantic graph. In the figure 4.2, there are three kinds of links: instanceOf is-a and hasSomething. The first kind of link represents the instantiation of the node while the second kind defines the type of object the node represents. This link defines the association of nodes to the directed node. The last one is more common link which represents the binary relationship between nodes. In general it could be considered that the links in a semantic graph defines its purpose and thus defining the knowledge.
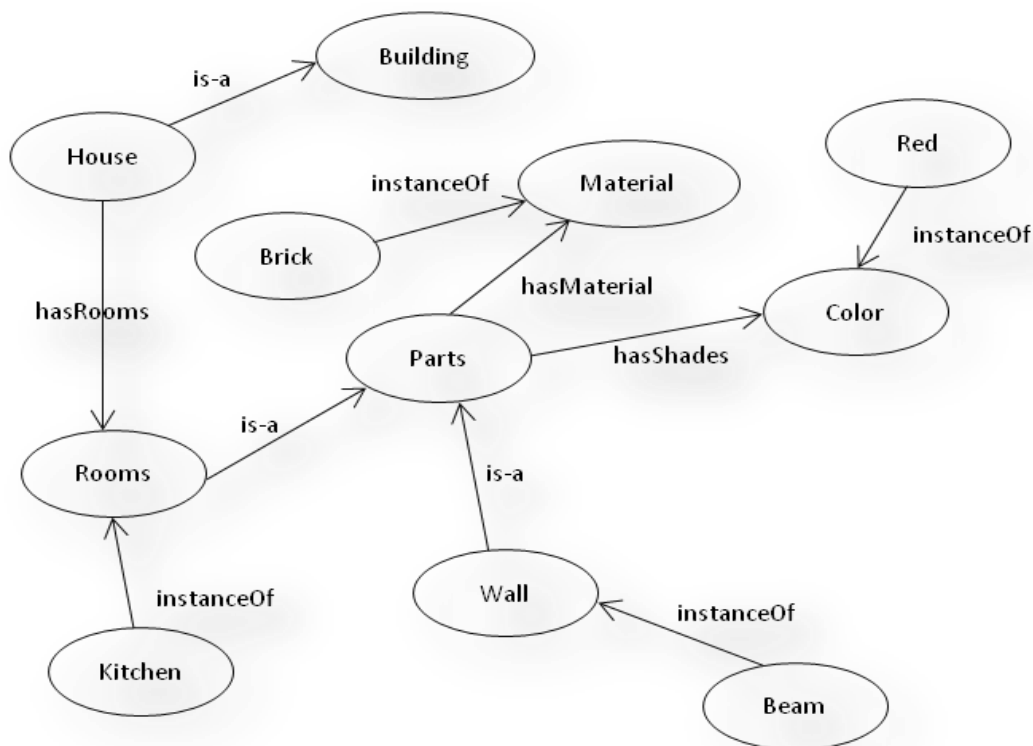
Figure 4.2: An Example of Semantic Graph

4.1.2. The Description logics

The convergence of formal foundations for extensible, semantically understood structure within description logic, the overall usability targets of the predecessor of DL and the Web languages for broader usability of Web has led to the effort such as Ontology Interface Language (OIL) (Fensel, Horrocks, Harmelen, McGuinnes, & Patel-Schneider, 2001). It is the first major effort to develop a language which has its base in Description Logic. It was a

part of broader project called On-To-Knowledge funded by European Union. This is the first time that the concept within ontology is explicitly used within a Web based environment. However, it did not completely left out the primitives of frame base languages with the formal semantics and reasoning capabilities by including them within the language. The syntax of OIL is based on RDF and XML with their limitations to provide complete semantic foundations at that time. However, it has started a trend of mapping description logic within the Web based language for Semantic Web. It maps description logic through $\mathcal{SHIQ}$. The derivation of $\mathcal{SHIQ}$ with respect to naming convention of the Description Logic (figure 4.4) is given as:

$\mathcal{S}$: Used for all $\mathcal{ALC}$ with transitive roles R_+

$\mathcal{H}$: Role inclusion axioms $R1 \sqsubseteq R2$ (is_component_of $\sqsubseteq$ is_part_of)

$\mathcal{I}$: Inverse Role R^- (isPartOf $=$ hasPart$^-$)

$\mathcal{Q}$: Qualified number restrictions ($\geq 1.hasCar.Human$)

During the same period, DARPA Agent Markup Language (DAML) was initiated in the United States. DAML was designed for the then next generation Web which would be increasingly using automated applications within Web environment and the dependency to human interpretation will decrease (Hendler & McGuinness, 2000). DAML-ONT was the extension of DAML integrating ontology. It worked to extend W3C RDF to add expressiveness to it. As both these systems were so similar, they were combined together for their effectiveness and service. Thus DAML+OIL is the language resulted in combination of DAML-ONT and OIL which provides machine-human understandability (Harlmelen, Patel-Schneider, & Horrocks, 2001). DAML+OIL has tighter integration with RDFS which provides the specification of the language and its only serialization (Horrocks, Pater-Schneider, McGuinness, & Welty, 2004). Even though DAML+OIL through its tighter integration with RDFS provides the essence of to become tool for the Semantic Web, it was still considered as result of *ad-hoc* researchers and was not made official part of the Semantic Web. The W3C (World Wide Web Consortium) later charted two groups to come out with a standard language for ontology within Web: first group was responsible to improve the RDF specification and second one to design the language based on these improve specifications. Result was the emergence of Web Ontology Language (OWL). OWL can be considered as the latest edition of DAML+OIL and it is hugely influenced by its predecessors. This section

talks the technology of RDFS and OWL in detail as they are the main technology to process semantics for knowledge manipulation.

4.1.3. The Description Logics Formalization

Description logics (DLs) are a family of logics which represents the structured knowledge. The Description Logic languages are knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way (Baader, Calvanese , McGuinness, Nardi, & Patel-Schneider, 2003); (Baader & Sattler, 2001); (Calvanese, Giacomo, Lenzerini, & Nardi, 2001). Description logics contain the formal, logic-based semantics, the major reason for its choice for Semantic Web languages over its predecessors. The reasoning capabilities within the DLs add a new dimension. Having these capabilities as central theme, one can infer implicitly represented knowledge. One of the earliest systems which utilized Description Logics within it was UNTANGLE system (Welty & Jenkins, 2000), a system representing bibliographic information. The system was being developed during the period of revolutionary development of Web through the Internet and thus the first Description Logic based Web system was born (Baader, Calvanese , McGuinness, Nardi, & Patel-Schneider, 2003). The project combined LISP-CLASSIC and the COMMONLIST Hypermedia Server (CL-HTTP) (Mallery, 1994) to implement hypertext based on the knowledge base comprising ABox and TBox statements through their semantic networks to list them in sub/super concept taxonomies. The initial aim of the project was to apply DLs to classify and retrieve the card catalog information. Later as the Internet became prominent information source, the aim shifted from card cataloging to page classification system within Web (Welty C. , 1996). FINDUR is another system which used description logic before DL languages were introduced. The concept behind the system was to use synonyms or hyponyms of a term to expand the query. The extensive use of thesaurus information built in a DL through the basic notion of WorldNet has set up the current trend of using controlled vocabulary for the data interoperability. The movement of Description Logic into its applicability can be viewed in terms of its progression in Web environment (McGuinness D. L., 2001). Web languages such as XML or RDF(S) could benefit from the approach DL takes to formalize the structured knowledge representation (Lassila & Swick, 1999). This has laid background behind the emergence of Description Logic languages in Web.

4.1.3.1. The Knowledge Base

Description Logics supports serialization through the human readable forms of the real world scenario with the classification of concepts and individuals. Moreover, they support the hierarchical structure of concepts in forms of subconcepts/superconcepts relationships of a concept between the concepts of a given terminology. This hierarchical structure provides efficient inference through the proper relations between different concepts. The individual-concept relationship could be compared to instantiation of an object to its class in object-oriented concept. In this manner, the approach DL takes can be related to classification of objects in a real world scenario.

Description logics provide formalization to knowledge representation of real world situations. This means it should provide the logical replies to the queries of real world situations. This is currently most researched topic in this domain. The results are highly sophisticated reasoning engines which utilize the capabilities of expressiveness of DLs manipulate the knowledge. A Knowledge Representation system is a formal representation of a knowledge described through different technologies. When it is describe through DLs, they set up a Knowledge Base (KB), the contents of which could be reasoned or infer to manipulate them. A knowledge base could be considered as a complete package of knowledge content. It is however only a subset of a KR system that contains additional components.
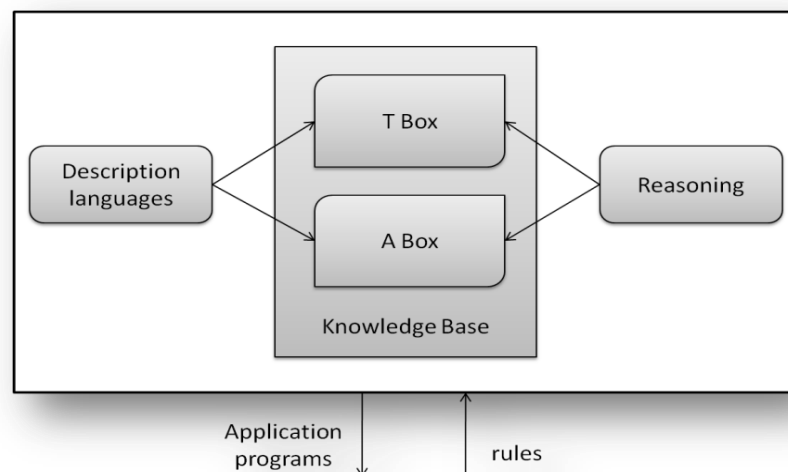
Figure 4.3: The Architecture of a knowledge representation system based on DLs

Figure 4.3 (Baader & Nutt, Basic Description Logics, 2002) sketches the architecture of any KR system based on DLs. It could be seen the central theme of such a system is a Knowledge Base (KB). The KB constitutes of two components: the TBox and the ABox.

TBox statements are the *terms* or the *terminologies* that are used within the system domain. In general they are statements describing the domain through the controlled vocabularies. For example in terms of a social domain the TBox statements are the set of concepts as *People*, *Male, Female*, *Father*, *Daughter* etc. or the set of roles as *marriedTo*, *siblingOf*, *sonOf*, *hasDaughter* etc. ABox in other hand contains *assertions* to the TBox statements. For example *Ashish is a Male* is an ABox statement. In object oriented concept **ABox** statements compliant TBox statements through instantiating what is equivalent to classes in TBox and relating the roles (equivalent to methods or properties in OO concept) to those instances.

The DLs are expressed through the *concepts* and *roles* of a particular domain. This complements well with the fact how knowledge is expressed in the general term. Concepts are sets of classes of individual objects. Classes provide an abstraction mechanism for grouping resources with similar characteristics (Bechhofer, et al., 2004).The concepts can be organized into superclass-subclass hierarchy which is also known as taxonomy. It shares the object-oriented concepts in managing the hierarchy of superconcept-subconcept. The subconcepts are specialized concepts of their superconcepts and the superconcepts are generalized concepts of their subconcepts. The *subsumption* algorithm determines the superclass-subclass relationships. For an example all individuals of a class must be individuals of its superclass. In general all concepts are subsumed by their superclass. In any graphical representation of knowledge concepts are represented through the nodes. Similarly the roles are binary relationship between concepts and eventually the relationships of the individuals of those concepts. They are represented by links in the graphical representation of knowledge. The description language has a model-theoretic semantics as the language for building the descriptions is independent to each DL system. Thus, statements in the TBox and in the ABox can be identified as first-order logic or, in some cases, a slight extension of it (Baader & Nutt, Basic Description Logics, 2002).

4.1.3.2. The Base Languages

Complex descriptions can be built up through these elementary descriptions of concepts and roles. These descriptions are given different notations over the time. The Attributive Language (*AL*) has been introduced in 1991 as minimal language that is of practical interest (Schmidt-Schauß & Smolka, 1991). It is further complemented through *Attributive Concept Language* with Complements (*ALC*) to allow any concepts or roles (and not just atomic concepts and atomic roles which were the previous elements of descriptions) be included. *ALC* is the important notation format to express Description Logics. The table 3.4 illustrates the syntax rules on describing the concept descriptions. We introduce terminological axioms, which make statements about how concepts or roles are related to each other. Then we single out definitions as specific axioms and identify terminologies as sets of definitions by which we can introduce atomic concepts as abbreviations or names for complex concepts. In the most general case, terminological axioms have the form

$$C \sqsubseteq D \quad R \sqsubseteq S \quad or \ C \equiv D \quad R \equiv S \ .$$

Here C, D are concepts (and R, S are roles). Axioms of the first kind are called inclusions, while axioms of the second kind are called equalities. An equality whose left-hand side is an atomic concept is a definition. Definitions are used to introduce symbolic names for complex descriptions. For instance, by the axiom

$$Mother \equiv Women \sqcap \exists hasChild.Person$$

Notations	Syntax	Semantics	Read-as
⊤	C, D→⊤	⊤(x)	Universal Concept
⊥	⊥	⊥(x)	Bottom Concept
⊓	$C \sqcap D$	$C(x) \sqcap D(x)$	Intersection
⊔	$C \sqcup D$	$C \ x \ \sqcup D(x)$	Union
¬	¬C	¬C(x)	Negation
∃	∃R.C	∃R(x,y) ∩ C(y)	Existential Quantification
∀	∀R.C	∀y.R(x,y)→C(y)	Value Restriction

Here C and D are concept description and R is role

Table 4.1: The syntax and semantics based on *ALC*

It could be clearly seen that these concept descriptions are built with the concept constructors. The first four constructors are not dependent on the roles whereas the last two utilizes the roles in the constructors. This dependency is called role restrictions. Formally, a role restriction is an unnamed class containing all individuals that satisfy the restriction. DLs expressed through $\mathcal{ALC}$ provide two such restrictions in Quantifier restriction and value restrictions.

The Quantifier restriction is again classified as

- The ***existential quantifier*** (∃) is read as *at least one*, or *some*

  This constraint links a restriction concept to a concept description or a data range. This restriction describes the unnamed concept for which there should be at least one instance of the concept description or value of the data value. Simplifying, the property restriction *P* relates to a concept of individuals *x* having at least one y which is either an instance of concept description or a value of data range so that *P(x,y)* is an instance of *P*.

  An example of *Person hasFaith Religion* is a good example for this qualifier. It says any person should have faith in at least one religion. So this built-in could be used for this example.

  So the Restriction: $\exists\ hasFaith.\ Religion\ \sqsubseteq\ Person$

- The ***universal quantifier*** (∀) is read as *every*

  This value constraint links a restriction concept to a concept description or a data range. This restriction describes the unnamed concept for which there should all instance of the concept description or value of the data value. Simplifying, the property restriction *P* relates to a concept of individuals *x* having all y which is either an instance of concept description or a value of data range so that *P(x,y)* is an instance of *P*.

  Examples like individuals of *Human hasParent Human* is a good example for this quantifier. It says that every human has parent which are instances of human. So this built-in could be used for this example.

So the Restriction ∀hasParent.Human⊑Humandescribes

- The **Value restriction** links a restriction concept directly to a value which could be either an individual or data value. An example is specific individual of the concept *Person hasFaith* which is an instance of concept *Religion*.

This example could be shown like *hasFaith*(Ashish, Hinduism)

4.1.3.3. The Normative Convention

The names of the operators in Description Logics have not yet been formalized. However, there is an agreed method to encode these operators using an alphabetic letter to denote expressivity of DLs. These letters in combinations are used to define the capabilities of DLs in terms of their performances. This implies to the DL languages as well. As could be seen in figure 4.4 (Bobillo & Straccia, 2009), $\mathcal{ALC}$ has been extended to transitive role and given abbreviation $\mathcal{S}$ in the convention. This $\mathcal{S}$ is used in every DL systems and languages as it plays significant role in shaping the behavioral nature of every DL systems.

$$\mathcal{AL}: \quad C, D \quad \longrightarrow \quad \top \mid \bot \mid A \mid C \sqcap D \mid \neg A \mid \exists R.\top \mid \forall R.C$$

$\mathcal{C}$: Concept negation, $\neg C$. Thus, $\mathcal{ALC} = \mathcal{AL} + \mathcal{C}$
$\mathcal{S}$: Used for $\mathcal{ALC}$ with transitive roles $\mathcal{R}_+$
$\mathcal{U}$: Concept disjunction, $C_1 \sqcup C_2$
$\mathcal{E}$: Existential quantification, $\exists R.C$
$\mathcal{H}$: Role inclusion axioms, $R_1 \sqsubseteq R_2$, e.g.,
  $is\_component\_of \sqsubseteq is\_part\_of$
$\mathcal{N}$: Number restrictions, $(\geqslant n\ R)$ and $(\leqslant n\ R)$, e.g., $(\geqslant 3\ has\_Child)$
  (has at least 3 children)
$\mathcal{Q}$: Qualified number restrictions, $(\geqslant n\ R.C)$ and $(\leqslant n\ R.C)$, e.g.,
  $(\leqslant 2\ has\_Child.Adult)$ (has at most 2 adult children)
$\mathcal{O}$: Nominals (singleton class), $\{a\}$, e.g., $\exists has\_child.\{mary\}$.
  Note: $a{:}C$ equiv to $\{a\} \sqsubseteq C$ and $(a, b){:}R$ equiv to $\{a\} \sqsubseteq \exists R.\{b\}$
$\mathcal{I}$: Inverse role, R^-, e.g., $isPartOf = hasPart^-$
$\mathcal{F}$: Functional role, f, e.g., $functional(hasAge)$
$\mathcal{R}_+$: transitive role, e.g., $transitive(isPartOf)$
$\mathcal{R}$: role inclusions with composition, $R_1 \circ R_2 \sqsubseteq S$, e.g.,
  $isPartOf \circ isPartOf \sqsubseteq isPartOf$

Figure 4.4: Naming convention of Description Logic

4.2. The Semantic Web Technologies

The Semantic Web is generally used to define the technologies involved and is proposed by the W3C. This section discusses about these technologies. This discussion starts by an

overview given by the Semantic Web Stack which underlines the fact that every language is based on another. In this stack two groups of languages are identified. Consequently, the two following sections focus on these groups.

4.2.1. The Semantic Web Stack

The Semantic Web stack also called the Semantic Web cake is basically a hierarchy of the technologies composed of different layers. Each layer takes advantages of the capabilities concerning all the sub-layers. The following figure 4.5. illustrates the Semantic Web cake.
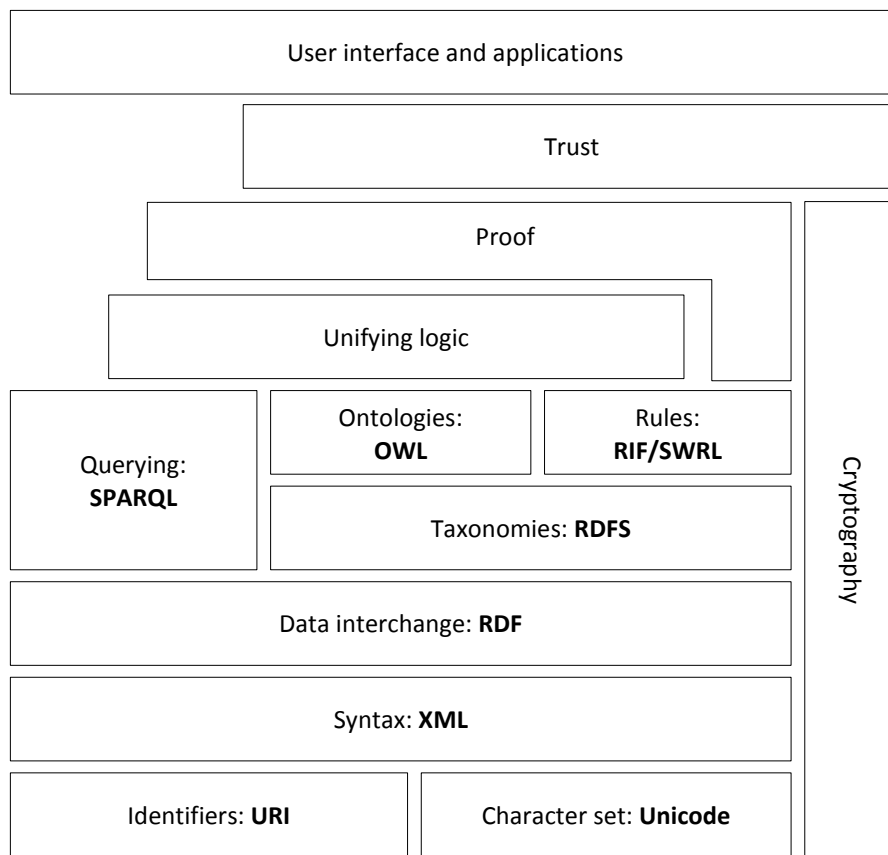
Figure 4.5: The Semantic Web Stack

There is a degrees of uncertainty in which the Semantic Web cake is defined. There are four versions of this cake till date, and none of them have been published in the literatures. All the four versions are presented by Berners-Lee in his presentations (Gerber, Merwe, & Barnard, 2008). The components and their relationships are hence been defined profoundly. It is thus necessary to isolate each component and discuss their role in terms of the Web. The definitions of some layers within the semantic cake are illusive and could be interpreted in many ways. However, some layers especially the lower layers have clear definitions. Here,

these hierarchical layers are discussed in terms of the knowledge representation approach. The layers in the Semantic Web stack can generally be divided into three categories: syntactic layers, knowledge layers and certifying layers composing of different technologies to support the technology.

The bottom layers are information holding layers and are either presented in uniform language or the through XML based information. The components within this layer hold the technologies that are direct descendent technologies from the hypertext Web. Though they are carryovers from basic technologies, they provide strong base to the Semantic Web. The technologies within these layers present syntactical representation of the information and thus be grouped into one common category of syntactic layers. They are capable to hold huge amount of information in each of the individual technologies within the level. These technologies include basic technologies as URI or content based technologies as XML and RDF. Despite rich with contents they lack interpretations as they do not possess semantics within.

The middle section contains layers which represents knowledge. These layers generally represent the technologies standardize by W3C for processing knowledge and can be grouped together as knowledge layers. The technologies here utilize the syntactically rich technologies in layers beneath. The knowledge is generated through attaching semantics to the information. RDFS provides vocabulary to RDF thus providing semantics to the structured statements representing the information as triplets. Through RDFS technology it is possible to derive hierarchical representations of objects and relate the objects to each other. The technology breaches the gaps between syntactically rich contents and tools to interpret knowledge from these contents. RDFS can define ontologies. Ontologies play important roles in order to provide semantics to the information or to the contents by providing suitable vocabulary to the contents and uplifts contents to resources which could be related to real world objects. As a result of the work of the W3C Web Ontology Working Group, the "Ontology" layer has now been instantiated with the Web Ontology Language (OWL (Smith, Welty, & McGuinness, 2004)) (Horrocks, Parsia, Patel-Schneider, & Hendler, 2005) due to its extended constructs to describe the semantics of the RDF statement. The semantic within the ontologies expressed through OWL can be used within the ontologies and the knowledge bases themselves for the inferences. However, in order to express the rules independent to the languages two standards are emerging in the form of RIF (Boley & kifer, 2010) and Semantic

Web Rule Language (SWRL) (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004). The rules are supported through inference engines. Simple Protocol and RDF Query Language (SPARQL) (Prud'hommeaux & Seaborne, 2008) is SQL equivalent language for querying data stored as RDF resources. As OWL is basically written in RDF pattern so the query could be applied to it as well. The topmost layer within knowledge layer is the unifying logic layer. This layer provides the logic behind knowledge manipulation through the reasoning capabilities of reasoning engines. This layer has not been formally defined so subjected to certain degrees of manipulation.

The top two layers in the stack are not yet fully conceptualized in terms of their applicability. These layers contain technologies which are not standardized yet but still they point toward maintaining the authenticity in the knowledge generated. The layer describing proof is therefore presumed responsible for providing evidence for the accuracy. At present there is no technology recommended to support this layer but there is an attempt for developing a proof language called Proof Mark-up Language (PML (da Silva, McGuinness, & Fikes, 2004)) (Al-Feel, Koutb, & Suoror, 2009) by knowledge systems laboratory at Stanford University. The top most layer *Trust* is to certify the knowledge reliability and there is a degrees of confidence in the knowledge generated within the layers under it. Again, at present there is no technology to support the layer.

The figure 4.5 can hence be updated with the three categories defined in this section and is illustrated in figure 4.6.
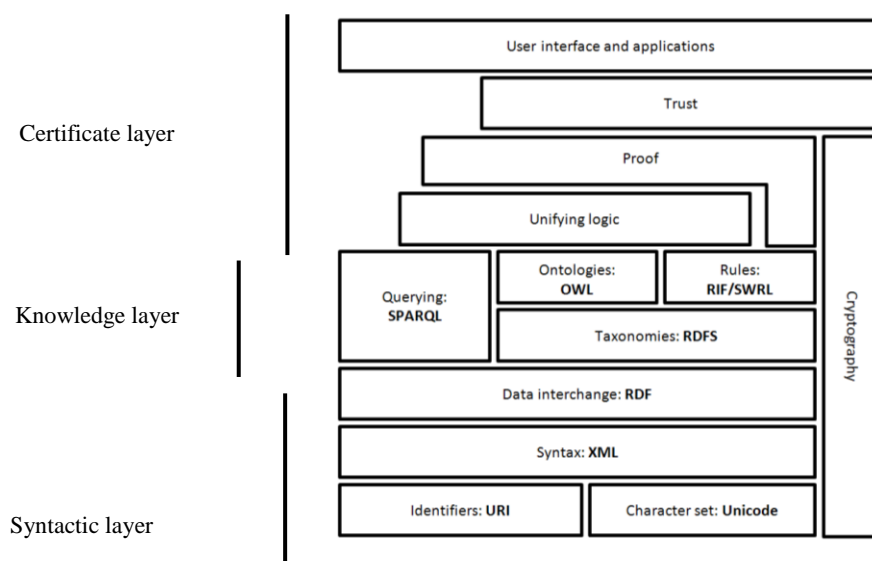
Figure 4.6: The layers of the Semantic Web stack

4.2.2. The Syntactic Layer

Semantic Web technologies are built up through the Web technologies that could hold up contents. The emergence of the **eXtensible Markup Language** (**XML**) marked the beginning of content based information in the Web environment. The language can encode information in machine readable format. The XML syntax is recommended in various data models and this syntactical approach laid a foundation for data models for defining metadata as Resource Description Framework (RDF). Resources are conventionally described through their metadata. The W3C recommended RDF as a standard to define the resources on Web. W3C has defined five major reasons for developing the standard (Klyne, Caroll, & McBride, 2004). They focus on automatization of the information processing through serialization. That means the contents inside the documents are machine processable. In order for the documents to be machine processable they need to be machine readable and since the syntax of RDF is based on XML, it provides a mechanism to represent the information in machine readable manner.

The **RDF** (Resource Description Framework) is a graph data model. It is basically a framework to represent information on the Web. It has also been assigned as the standard model for data interchange on the Web by W3C because it can merge different sets of data irrespective to the underlying schema. RDF is conceptualized through graph data model which demonstrates the underlying structure of its expression. The nodes in the graph model are resources which can represent Uniform Resource Identifiers (URI reference or simply URIRef) or literals or even blank. The link the graph representing properties are generally URI references. The literals within RDF expressions are generally assigned values of certain data types. RDF syntax is primarily based on its predecessor XML and is defined by RDF abstract syntax. This abstract syntax is the syntax over which the formal semantic are defined. It is a set of triples known as RDF graph (Klyne, Caroll, & McBride, 2004). It consists three parts which are normally called RDF triplet and represent a statement of relationship between the objects. Figure 4.7 sketches a general graph data model representing a RDF statement.
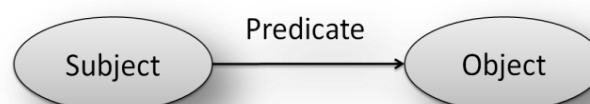
Figure 4.7: The RDF Graph Model

As could be seen the RDF graph is a set of RDF triplets. Each triplet consists of three parts: *Subject*, *Predicate* and *Object*. The node of a graph is either a subject or an object. The subject and the object represent the resources and the predicate represents their relations. RDF itself does not include semantic information. It is still a general purpose language for representing information in the Web. However, it is an assertion language and could provide a base for future technologies to express propositions using precise formal vocabularies (Hayes & McBride, 2004). RDF Schema or RDFS introduces controlled vocabularies to RDF syntax thus linking the semantic into it. Details on RDFS technology will be discussed later in this section.

4.2.3. The Knowledge Layer

Knowledge representation has been described in five distinct roles it plays in (Davis, Shrobe, & Szolovits, 1993). Those roles are

- A surrogate for the *thing* itself used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than acting it.
- A set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- A fragmentary theory of intelligent, reasoning, expressed in terms of three components
  - The representation's fundamental conception of intelligent reasoning
  - The set of inferences the representation sanctions; and
  - The set of inferences it recommends
- A medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished.
- A medium for human to express, i.e., a language human expresses things about the world.

With these roles in view, different languages that represent the knowledge have been conceived over the time. They vary in each other in terms of their characteristics, expressive power and computational complexity. The effectiveness of any representation language can be measured in:

- The **expressiveness** of the language is measured in terms of the range through which the language can use its constructs to describe the components in knowledge model.

- The **strictness** in the language is measured through the consistency and satisfiability within the knowledge model. The consistency and satisfiabilty issue is important in any knowledge model because they decide the reliability of the model. If any model contains statements which contradict with each other, the model cannot be considered reliable. For an example A cannot be a father and son of B at the same time. Such statements should be rigorously audited for the model to be reliable enough.
- The **semantic** within the model should not be ambiguous. The meaning of each statement within the model should be clear and unambiguous.

4.2.3.1. RDFS

RDFS or the RDF Schema is the semantic extension of RDF. The applications using RDF uses it to describe its resources and those descriptions can be modeled as relationships among Web resources. These models constitute of interrelationships among the resources. They are carried out through the named properties and values. It however lacks the mechanism of defining the relationships between properties and other resources. Furthermore RDF data models do not declare these properties. They are hence information without any semantics. RDFS is designed to address these shortcomings. RDFS provides mechanisms for describing groups of related resources and the relationships between these resources.

4.2.3.1.1 Syntax

RDFS is an extension of RDF so it does not provide actual classes and properties for its own application specific but uses RDF resources to define a framework to define application specific classes and properties. The syntax it follows is similar to the syntax of RDF. It means that it uses XML based syntax to attach vocabularies to RDF resources.

Resources are divided into groups called **classes** which are equivalent to concepts in description logics. These classes can be instantiated to have their members which are called **instances**. Those classes are described through **properties**. Till this point, RDFS has been following the culture of its predecessor RDF. Differences start to arise when the structures of their presentation are analyzed. In contrary to RDF where RDFS distinguishes between a class and set of its instances, RDFS may have a member of a class as its own class extension and also an instance of itself. It uses the components defined in RDF and carries them forward to supply vocabularies to them.
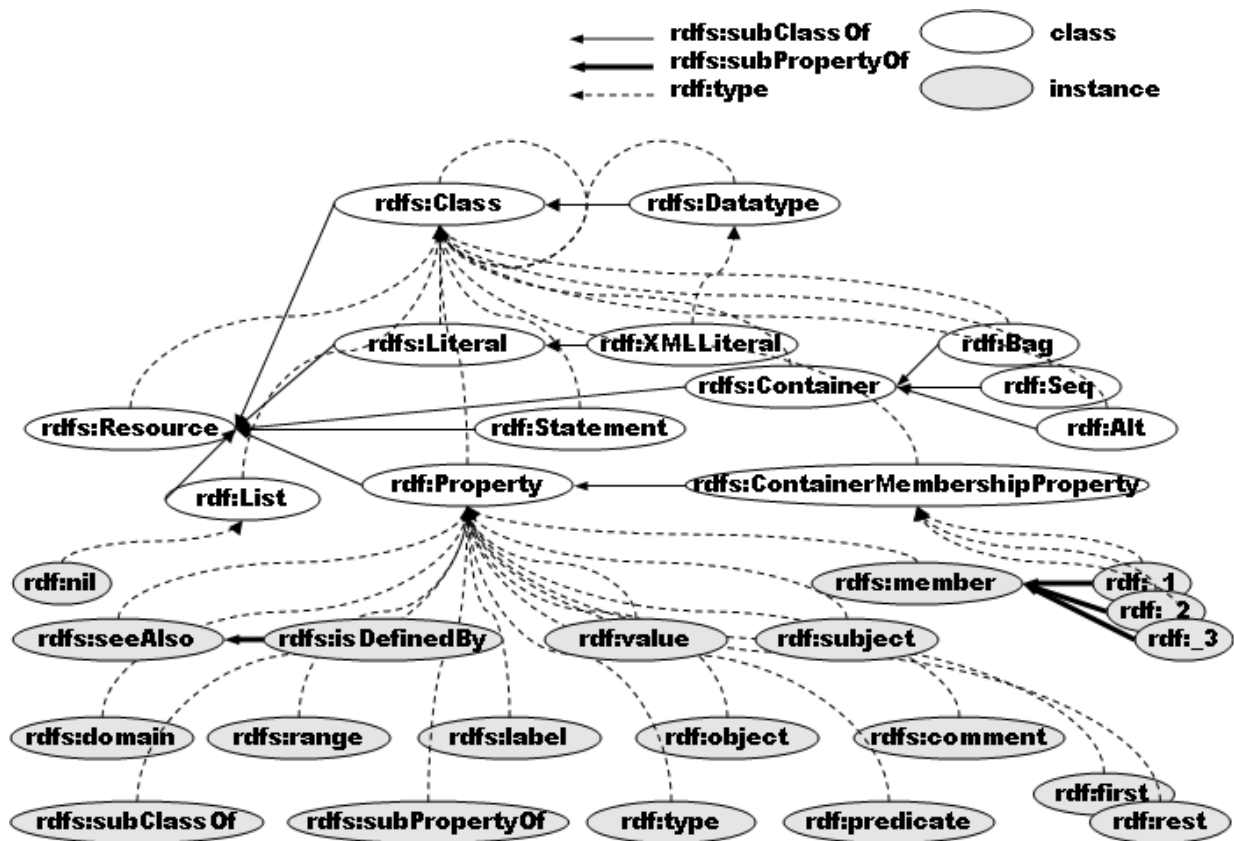
Figure 4.8: The Hierarchical structure of RDFS

4.2.3.1.2 Class

Every component described by RDF is a resource here and is grouped together with other similar resources. This group of resources classes itself as `rdfs:Class`. The class `rdfs:Class` instantiates itself as instance `rdfs:Class`. Every resource in RDFS is an instance of class `rdfs:Resource` whereas `rdfs:Resource` is again an instance of `rdfs:Class`.

The detail hierarchal diagram of RDFS class is given in figure 4.8. The diagram can be viewed as complex of classes and their instances. The pattern of a possibility of a class being member of itself or its extension is embraced by `rdfs:Literal` and `rdfs:Datatype`. As could be seen `rdfs:Literal` is a class which is both a subclass and an instance of `rdfs:Class` and constitutes of literal values. In same fashion `rdfs:Datatype` is a class which represents the datatypes and is both a subclass and an instance of class `rdfs:Class`.

4.2.3.1.3 Property

Another important component in RDFS is `rdf:Property` which is instance of `rdfs:Class`. It is itself a class representing the property specification of RDF properties. The class `rdf:Property` is instantiated through `rdfs:subPropertyOf` which represents the sub property of a property. The concept of Property – subProperty provides the functionality of grouping properties together with common elements. Instances of `rdf:Property` like `rdfs:domain` and `rdfs:range` provide methods to define a property linking two classes defined in independent namespaces. It is thus flexible enough to be adopted in different processing mechanism as it just describes the information through some mechanism but does not restrict to how they should be used.

4.2.3.2 *The Web Ontology Language (OWL)*

OWL or the Web Ontology Language is a family of knowledge representation language to create and manage ontologies. It is in general term an extension of RDFS with addition to richer expressiveness that RDFS lacks through its missing features (Antoniou & Harmelen, 2003). The OWL Working Group has approved two versions of OWL: OWL 1 and OWL 2. This thesis research uses OWL 1 for the applications of ontology as this version was the most used version at the time of research. The later version of OWL 1 was just evolving during the period. This research thesis discusses its activities in terms of OWL 1.

The expressiveness of OWL depends upon the level of serialization. It should be considered that the expressiveness of OWL comes at the cost of computational efficiency and reasoning effectiveness. This tradeoff between expressiveness and reasoning support was addressed through classifying OWL into three sub languages by the W3C Web Ontology working group.

OWL Full contains the maximum expressiveness but may lack in computational processing capability. It may also have restricted reasoning efficiency. *OWL Full* is completely compatible with RDF/RDFS both syntactically and semantically. *OWLDL* is compatible to the components of description logics and provides the functionalities of *DLs*. It provides the complete computational efficiency and reasoning capabilities. It is sub language of *OWL Full* with all OWL language constructs which could be used only through certain restrictions (McGuinness & Harmelen, 2004). This restriction is even more in *OWL Lite* – the third

sublanguage of OWL. The advantage of this language is its easiness to understand and implement but the drawback is it is just a simple and fast migration from thesauri and other taxonomies.

4.2.3.2.1 Syntax

OWL uses RDF's XML as its primary syntax. As OWL builds on RDF and RDF Schema, it uses the features in a similar approach as its counterpart. Figure 4.9 shows the comparison of different elements between OWL and RDF/RDFS (Antoniou & Harmelen, 2003).
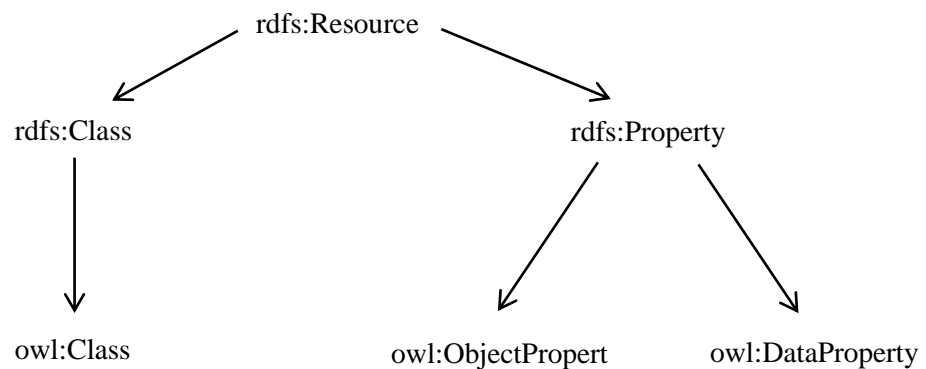
Figure 4.9: Elements comparison between OWL and RDF/RDFS

Other forms of syntax are also used to define OWL like the XML – based syntax which does not follow RDF convention or the abstract syntax which is compact and more readable. However to represent an ontology, the general trend is to represent it in graphical structure of graph model as representing RDF triplets. A graphical approach based on the convention of Universal Modeling Language (UML) is also widely used to represent ontology graphically.

4.2.3.2.2 Header

The header of an ontology document generally describes about itself. As it is basically a RDF document so the root is generally described through an `rdf:RDF` tag. This also specifies a number of namespace for defining the prefix within the ontology.

```
<rdf:RDF
xmlns:sa="http://www.archaeokm.com/sa#"
xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
xmlns:shape="http://www.archeokm.com/shape#"
xmlns="http://www.archeokm.com/archeokm.owl#"
xmlns:feat="http://www.archeokm.com/feature#"
```

```
                    xmlns:site="http://www.archeokm.com/site#"
                    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
                    xmlns:doc="http://www.archeokm.com/document#"
                    xmlns:ann="http://www.archeokm.com/annotation#"
                    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

Once these namespaces are defined, the OWL document defines itself through the tag

```
                    <owl:Ontologyrdf:about="">
```

4.2.3.2.3 Class

Classes are used through `owl:Class` tag within the OWL document. The classes can be organized into superclass-subclass hierarchy which is also known as taxonomy. It shares the object-oriented concepts in managing the hierarchy of superclass-subclass. The subclasses are specialized classes of their superclasses and superclasses are generalized classes of their subclasses. The subsumption algorithm determines the superclass-subclass relationships. For an example all individuals of a class must be individuals of its superclass. In general all classes are subsumed by their superclass. In any graphical representation of knowledge classes are represented through the nodes.

OWL classes are described through "class descriptions" through the class name or by through the property restrictions of the class. There are six different types of class description (Bechhofer, et al., 2004); however this thesis discusses the class description of Property Restriction due to its relevancy on the thesis activities. They are listed in the table 4.2

Restriction Type	Sub Type	Tag in the Document	DL Syntax	Description

Value Constraints	AllValuesFrom	`<owl:Restriction>`
		`  <owl:onPropertyrdf:resource="#hasParent"`
		`  />`
		`  <owl:allValuesFromrdf:resource="#Human"`
		`  />`
		`</owl:Restriction>`
			∀ hasPar	Equivalent to Universal quantifier

	SomeValuesFrom	`<owl:Restriction>`
		`  <owl:onPropertyrdf:resource="#hasFaith"`
		`  />`
		`  <owl:someValuesFromrdf:resource="#Religio`
		`n" />`
		`</owl:Restriction>`
			∃ hasFai	Equivalent to Existential quantifier

hasValue		`<owl:Restriction>` `  <owl:onPropertyrdf:resource="#hasFaith"`		Links directly to an individual or data

		`/>`
		`<owl:hasValuerdf:resource="#Hinduism" />`
		`</owl:Restriction>`
			value

Cardinality Constraints	MaxCardinality	`<owl:Restriction>`
		`  <owl:onPropertyrdf:resource="#hasParent"`
		`  />`
		`  <owl:maxCardinalityrdf:datatype="&xsd;no`
		`  nNegativeInteger">2</owl:maxCardinality>`
		`</owl:Restriction>`
			The maximum value a property can have

	MinCardinality	`<owl:Restriction>`
		`  <owl:onPropertyrdf:resource="#hasFaith"`
		`  />`
		`  <owl:minCardinalityrdf:datatype="&xsd;non`
		`  NegativeInteger">0</owl:minCardinality>`
		`</owl:Restriction>`
			The minimum value a property can have

	Cardinality	`<owl:Restriction>`
		`  <owl:onPropertyrdf:resource="#hasPlayerCr`
		`  " />`
		`  <owl:cardinalityrdf:datatype="&xsd;nonNeg`
		`  ativeInteger">11</owl:cardinality>`
		`</owl:Restriction>`
			Exact number a property can have

Table 4.2: The Property Restrictions in OWL

4.2.3.2.4 Properties

Properties are binary relationships between two objects. In general they are the relationships between two classes which apply to the individual of those classes. They are known as *roles* in description logic and are represented through links in the graphical representation. In OWL there are two main categories of properties:

- Object properties – relationships between concepts and consequently instances of the concepts
- Data properties – relation of an instance to the data value

Properties can be categorized as super/sub property. Any properties have a domain and a range. In general a domain is a class description through which the property is related and range is a class description with which the property is related to.

Any property can be characterized through the list of *property characteristics*. Almost all ontology language supports these property characteristics but as already mentioned we discuss them in terms of OWL.

- **Functional Properties:** The property is *functional* if it relates an individual of a class to at most one individual of another class. In short there cannot be two distinct instances y1 and y2 for an individual x with a functional property P. So the property P(x,y1) and P(x,y2) cannot exists. The *hasBirthCountry* property below is functional property as an individual can have only one birth country.

  hasBirthCountry(Ashish, Nepal)

- **Inverse Functional Properties:** The property is *inversefunctional* when it inverses to the *functional* property. That means there cannot be two distinct individuals x1 and x2 related to individual y with this property P. So the property P(x1,y) and P(x2,y) cannot exists. The example *isBirthCountryOf* which is just inverse of the property *hasBirthCountry* is inverse functional

  isBirthCountryOf(Nepal, Ashish)

- **Transitive Properties:** A property is transitive if any individual of a class is related to another individual through this property and the later individual is related to third individual through the same property then the first individual is related to third individual with the same property. In short if a is related to b and b is related to c then a is related to c.

$$a \rightarrow b, b \rightarrow c \equiv a \rightarrow c$$

  The property *hasAncestor* in the example below is a transitive property because

  hasAncestor(Ashish, Bishnu), hasAncestor(Bishnu, Krishna)→hasAncestor(Ashish, Krishna)

- **Symmetric Properties:** A property is symmetric if any individual is related to another with this property then the latter is related with the formal with the same property. In short if a is related to b then b is related to a.

$$a \rightarrow b \equiv b \rightarrow a$$

  The property *hasBrother* in the example below is symmetric because

hasBrother(Ashish, Avish) →hasBrother(Avish, Ashish)

4.2.3.2.5 Individuals

In object oriented terms, individuals are instances of classes but in general terms they are real world objects. In general individuals are building blocks of a knowledge base. Until we populate any ontology schema with these individuals, they are just schema. As we populate them through the individuals, they become knowledge base (Spaccapietra, Cullot, Parent, & Vangenot, 2004). In any knowledge base, the relationships among individuals define the knowledge that it possesses. The OWL does not use Unique Name Assumption (UNA) which means the same individual can have different names. This is dealt through the Identity of individuals through (Bechhofer, et al., 2004):

- `owl:sameAs` is used to state that two URI references refer to the same individual.
- `owl:differentFrom` is used to state that two URI references refer to different individuals
- `owl:AllDifferent` provides an idiom for stating that a list of individuals are all different.

4.2.3.2.6 OWL 2

OWL 2 or OWL 2 Web Ontology Language is the extension and revision of OWL 1. The basic principle and aim of OWL 2 is similar to OWL 1 as both are designed to facilitate ontology development and sharing via the Web, with the ultimate goal of making Web content more accessible to machines (W3C OWL Working Group, 2009). The overview of OWL 2 as proposed by W3C is shown in figure 4.10.
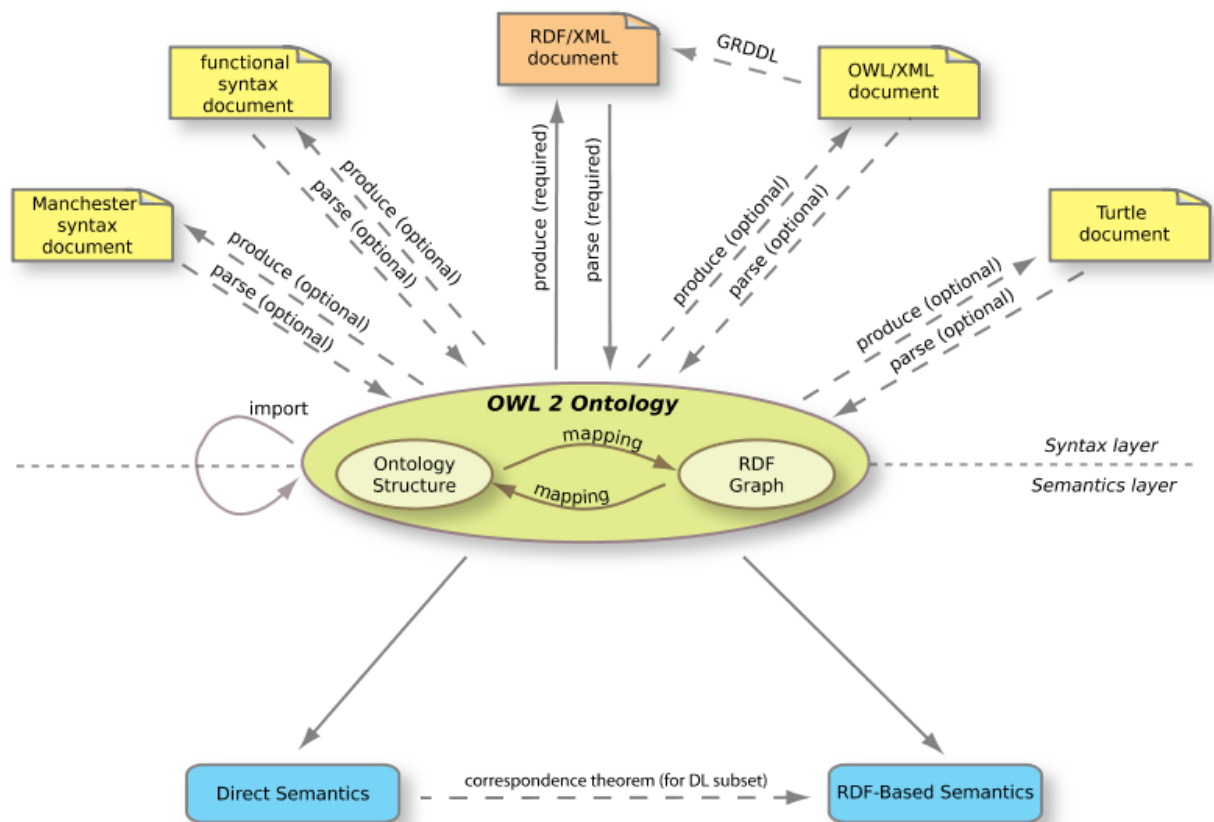
Figure 4.10: The OWL 2 Structure

As could be seen at the center of the structure an elliptical figure represents the notion of an abstract ontology or a notion of RDF graph. The structures at revolving on the top of the ellipse are the concrete syntaxes that can be used to serialize the information and exchange the ontologies. The primary syntax for OWL 2 is RDF/XML which needs to be supported by all OWL 2 tools. Besides, other concrete syntaxes can as RDF serialization turtle or XML serialization can also be used. The bottom in the structure represents the semantic specifications defining the meaning of OWL 2 ontologies. It either assigns meaning through Direct Semantics defined directly by OWL 2 or RDF-Based Semantics assigning meanings to RDF graph and hence indirectly to the ontology. The two methods of assigning semantics are mapped through a correspondence theorem.

OWL 2 have three sub languages known popularly as OWL 2 Profiles (Motik, Grau, Horrocks, Wu, Fokou, & Lutz, 2009). These profiles are OWL2 EL, OWL 2 QL, and OWL 2 RL. **OWL 2 EL** is a polynomial time algorithm which provides fast execution of the tasks. Hence such profile could be used for all standard reasoning tasks by trading off the expressiveness of the ontologies. OWL 2 EL is suitable for large scales ontologies containing

huge collections of classes and properties. **OWL 2 QL** is based on DL-Lite family of description logics. It performs reasoning in LOGSPACE with respect to the size of data. It is aimed at the ontologies consisting huge collection of instances. Such systems focus more on querying so could comply with conventional relational database management systems through query answering implementation of them. **OWL 2 RL** ontologies do not trade off too much in its expressiveness and still aims to have scalable reasoning. It is designed to adjust tradeoff in expressivity as in OWL 2 ontologies and still add up some expressiveness of RDFS. The reasoning system can be implemented using rule based reasoning engines.

The overall structure of OWL 2 is very similar to that of OWL 1. Every OWL 1 DL ontology is OWL 2 ontology (Motik, Patel-Schneider, & Grau, 2009). Most of the structures of OWL 1 have remained the same in OWL 2 and so is the central role of RDF/XML in the concrete syntax. OWL 2 though adds few new features. These features are expressed below (Bai, Kendall, McGuinness, & Patel-Schneider, 2009):

Name	Features
Class Expressions	• local reflexivity (self restriction) • object and data qualified exact/maximum/minimal cardinality restriction • universal and existential restriction on n-ary data range
Class Axioms	• pairwise disjoint classes • class disjoint union
Property Expressions	• universal and empty object property • universal and empty data property • inverse object property expression
Property Axioms	• property chain inclusion • disjoint object properties • disjoint data properties • reflexive, irreflexive, and asymmetric object property.
Data Ranges	• datatype definition • data range complement, intersection and union • datatype restriction and facets • hook for n-arydatatype
Assertions	• negative object property assertion • negative data property assertion
Annotation	• annotation assertion • annotation of an axiom or an annotation • annotation subproperties • annotation property domain and range

	• owl:deprecated annotation property
Extra Built-in Datatypes	• owl:rational, owl:real, xsd:dateTimeStamp, rdf:PlainLiteral
Others	• key • declaration • metamodeling capabilities (Punning) • anonymous individual

Table 4.3: The List of new features in the OWL2 language

4.2.3.3 SPARQL

It has been stated before that RDF statements store data in the form of informative contents. In this manner, it could be easily argued RDF documents are datasets complimenting the data storage capability of its conventional counterparts which are database systems. As database systems provide efficient retrieval of the data through its query language in form of Structured Query Language (SQL), the dataset within a RDF document can be retrieve through the query language called SPARQL. As with its counterpart SPARQL is also used to manage the RDF document. It is a key component of Semantic Web technology. As a query language, SPARQL is "data-oriented" in that it only queries the information held in the models; there is no inference in the query language itself. SPARQL does not do anything other than taking the description of what the application wants, in the form of a query, and returns that information in the form of a set of bindings or an RDF graph. In addition, the SPARQL is able to query OWL ontologies which use RDF graphs to structure it. However, no inferences are possible on that structure. SWRL is used for that purpose.

The query language has been standardized by W3C and has been recommended as official query language to retrieve RDF data (Prud'hommeaux & Seaborne, 2008).

SPARQL queries the RDF data in four distinct forms.

- SELECT returns the resulted dataset from this form. The results could be used accessed by the APIs as well could be serialized into XML or RDF graph.
- CONSTRUCT form constructs a RDF graph through running the query to derive the solution in solution sequence and then combines these triplets.

- ASK form is used to ask the authenticity of the query pattern. That means whether certain query pattern returns a solution or not.

- DESCRIBE forms describe the RDF data about its resources.

This research thesis concentrates on the SELECT form to execute the SPARQL query. This form can be easily compared to the SELECT statement is a SQL query as both have similar sequence. A SPARQL query comprises, in order:

- Prefix declarations, for abbreviating URIs
- A result clause, identifying what information to return from the query
- The query pattern, specifying what to query for in the underlying dataset
- Query modifiers, slicing, ordering, and otherwise rearranging query results

```
# prefix declarations
PREFIX foo: <http://example.com/resources/>
...

# result clause
SELECT ( 'DISTINCT' | 'REDUCED' )? ( Var+ | '*' )

# file selection
FROM <http://rdf.example.org/personA.rdf>

# query pattern
WHERE {
    ...
}

# query modifiers
ORDER BY ...
```

The DISTINCT or REDUCED modifier in the query removes or allows being removed from the result set (Harris & Seaborne, 2010). An example of the SELECT form is given below. The example is given in Turtle which is basically a textual syntax of RDF that allows RDF graph written in a natural form (Beckett & Berners-Lee, 2008). It can be understood better than its XML format and hence is used to express the statements in the example.

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .
_:a foaf:name           "Ashish".
_:a foaf:hasFriend      _:b.
_:a foaf:hasFamily      _:c.

_:b foaf:name           "Bimal".

_:c foaf:name           "Aryan".
```

```
        _:c foaf:nick          "RN".
```

The Query:

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?me ?frnd ?frndNm ?fml ?fmlNm ?fmlNick
WHERE
{
      ?me   foaf:hasFriend    ?frnd.
      ?frnd foaf:name         ?frndNm.
      ?me   foaf:hasFamily    ?fml.
      ?fml  foaf:name         ?fmlNm.
      OPTIONAL { ?fml foaf:nick ?fmlNick }
}
```

And the result is:

frndNm	fmlNick
"Bimal"	
"Aryan"	"RN"

SPARQL uses FILTERS to limit the solutions to only those which are returned true with the expression. The section presents the syntax of the FILTERS with an example. Generally FILTER comes at the end of any SPARQL expressions. There are two major restrictions that are possible through the FILTER at the moment.

4.2.3.3.1 String

FILTER can be used to compare the string and derive results. The functions like regex which matches plain literal with no language tag can be used to match the lexical forms of other literals by using string comparison function. Sticking with turtle to present the example

```
@prefix ls: <http://example.de/list/>
@prefix add: <http://example.de/address/>

:ppl1 ls:Name     "Ashish"
:ppl1 add:Address "Wiesbaden Moritzstr 37"
:ppl2 ls:Name     "Anand"
:ppl2 add:Address "Kathmandu 42 Kakani"
```

Applying SPARQL query with FILTER

```
@prefix ls: <http://example.de/list/>
@prefix add: <http://example.de/address/>

SELECT ?name ?addr
```

```
WHERE {
     ?x    ls:Name    ?name
     ?x    add:Address ?addr
           FILTER regex(?addr, "^Wiesbaden")
}
```

The result is:

Name	Address
Ashish	Wiesbaden Moritzstr 37

4.2.3.3.2 Numbers

SPARQL FILTER uses the relational operators as = or > or < for the comparison and restrict the result. Continuing the turtle example

```
@prefix ls: <http://example.de/list/>
@prefix add: <http://example.de/address/>
@prefix age: <http://example.de/age/>

:ppl1 ls:Name    "Ashish"
:ppl1 add:Address "Wiesbaden Moritzstr 37"
:ppl1 age:Age    35
:ppl2 ls:Name    "Anand"
:ppl2 add:Address "Kathmandu 42 Kakani"
:ppl2 age:Age    30
```

Applying SPARQL query with FILTER

```
@prefix ls: <http://example.de/list/>
@prefix add: <http://example.de/address/>
@prefix age: <http://example.de/age/>

SELECT ?name ?addr ?age
WHERE {
        ?x ls:Name ?name
        ?x add:Address ?addr
        ?x age:Age ?age
        FILTER (?age< 35)
}
```

The result is:

Name	Address	Age
Anand	Kathmandu 42 Kakani	30

4.2.3.2. SWRL

An inference process consists of applying logic in order to derive a conclusion based on the observations and hypothesis. In computer science interferences are applied through inference engines. These inference engines are basically computer applications which derive answers from a knowledge base. These engines depend on the logics through logic programming.

The horn logic more commonly known Horn clause is a clause with at most one positive literal. It has been used as the base of logic programming and Prolog languages (Sterling & Shapiro, 1994) for years. These languages allow the description of knowledge with predicates. Extensional knowledge is expressed as facts, while intentional knowledge is defined through rules (Spaccapietra, Cullot, Parent, & Vangenot, 2004). These rules are used through different Rule Languages to enhance the knowledge possess in ontology. The Horn logic has given a platform to define Horn-like rules through sub languages of RuleML (Boley H. , 2009). There have been different rule languages that have emerged in last few years. Some of these languages that have been evolving rapidly are Semantic Web Rule Language (SWRL) and JenaRule. Both have their own built-ins to support the rules. This thesis uses SWRL to demonstrate the concepts but it could be applied to others rule language based on Horn clauses.

Semantic Web Rule Language (SWRL (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004)) is a rule language based on the combination of the OWL-DL (($\mathcal{SHOIN(D)}$) illustrated in figure 4.4) with Unary/Binary Datalog RuleML which is a sublanguage of the Rule Markup Language. One restriction on SWRL called DL-safe rules was designed in order to keep the decidability of deduction algorithms. This restriction is not about the component of the language but on its interaction. SWRL includes a high-level abstract syntax for Horn-like rules. The SWRL as the form, *antecedent→consequent*, where both antecedent and consequent are conjunctions of atoms written $a_1 \wedge ... \wedge a_n$. Atoms in rules can be of the form $C(x)$, $P(x,y)$, $Q(x,z)$, *sameAs*(x,y), *differentFrom*(x,y), or *builtIn*(*pred*, z1, …, zn), where C is an OWL description, P is an OWL individual-valued property, Q is an OWL data-valued property, *pred* is a datatype predicate URIref, x and y are either individual-valued variables or OWL individuals, and z, z_1, … z_n are either data-valued variables or OWL data literals. An OWL data literal is either a typed literal or a plain literal. Variables are indicated by using the

standard convention of prefixing them with a question mark (e.g., ?x). URI references (URIrefs) are used to identify ontology elements such as classes, individual-valued properties and data-valued properties. For instance, the following rule asserts that one's parents' brothers are one's uncles where parent, brother and uncle are all individual-valued properties.

$$parent(?x, ?p) \wedge brother(?p, ?u) \rightarrow uncle(?x, ?u)$$

The set of built-ins for SWRL is motivated by a modular approach that will allow further extensions in future releases within a (hierarchical) taxonomy. SWRL's built-ins approach is also based on the reuse of existing built-ins in XQuery and XPath, which are themselves based on XML Schema by using the Datatypes. This system of built-ins should also help in the interoperation of SWRL with other Web formalisms by providing an extensible, modular built-ins infrastructure for Semantic Web Languages, Web Services, and Web applications. Many built-ins are defined and some of most common built-ins can be found below (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004). These built-ins are keys for any external integration. The thesis work uses spatial built-in for the integration of spatial data structure.

- *Comparisons Built-Ins*

The built-ins within this category are used to compare two values. They are generally relational operators. W3C (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004) has listed the built-ins lying in this category as:

Built-ins	Description
swrlb:equal	Satisfied if and only if the first argument and the second argument are the same.
swrlb:notEqual	The negation of swrlb:equal.
swrlb:lessThan	The lessThan operation is Satisfied if and only if the first argument and the second argument are both in some implemented type and the first argument is less than the second argument according to a type-specific ordering (partial or total), if there is one defined for the type. The ordering function for the type of untyped literals is the partial order defined as string ordering when the language tags are the same (or both missing) and incomparable otherwise.

swrlb:lessThanOrEqual	Either less than, as above, or equal, as above.
swrlb:greaterThan	Similarly to swrlb:lessThan.
swrlb:greaterThanOrEqual	Similarly to swrlb:lessThanOrEqual.

Table 4.4: The comparisons Built-Ins

- *Math Built-Ins*

These built-ins are defined for numeric data types. They are generally mathematical operations which could be implemented within SWRL. W3C (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004) has listed these built-ins as:

Built-ins	Description
swrlb:add	Satisfied iff the first argument is equal to the arithmetic sum of the second argument through the last argument.
swrlb:subtract	Satisfied iff the first argument is equal to the arithmetic difference of the second argument minus the third argument.
swrlb:multiply	Satisfied iff the first argument is equal to the arithmetic product of the second argument through the last argument.
swrlb:divide	Satisfied iff the first argument is equal to the arithmetic quotient of the second argument divided by the third argument.
swrlb:integerDivide	Satisfied if the first argument is the arithmetic quotient of the second argument idiv the third argument. If the numerator is not evenly divided by the divisor, then the quotient is the xsd:integer value obtained, ignoring any remainder that results from the division (that is, no rounding is performed).
swrlb:mod	Satisfied if the first argument represents the remainder resulting from dividing the second argument, the dividend, by the third argument, the divisor. The operation a mod b for operands that are xsd:integer or xsd:decimal, or types derived from them, produces a result such that (a idiv b)*b+(a mod b) is equal to a and the magnitude of the result is always less than the magnitude of b. This identity holds even in the special case that the dividend is the negative integer of largest possible magnitude for its type and the divisor is -1 (the remainder is 0). It follows from this rule that the sign of the result is the sign of the dividend
swrlb:pow	Satisfied iff the first argument is equal to the result of the second argument raised to the third argument power.

swrlb:unaryPlus	Satisfied iff the first argument is equal to the second argument with its sign unchanged.
swrlb:unaryMinus	Satisfied iff the first argument is equal to the second argument with its sign reversed.
swrlb:abs	Satisfied iff the first argument is the absolute value of the second argument.
swrlb:ceiling	Satisfied iff the first argument is the smallest number with no fractional part that is greater than or equal to the second argument.
swrlb:floor	Satisfied iff the first argument is the largest number with no fractional part that is less than or equal to the second argument.
swrlb:round	Satisfied iff the first argument is equal to the nearest number to the second argument with no fractional part.
swrlb:roundHalfToEven	Satisfied iff the first argument is equal to the second argument rounded to the given precision. If the fractional part is exactly half, the result is the number whose least significant digit is even.
swrlb:sin	Satisfied iff the first argument is equal to the sine of the radian value the second argument.
swrlb:cos	Satisfied iff the first argument is equal to the cosine of the radian value the second argument.
swrlb:tan	Satisfied iff the first argument is equal to the tangent of the radian value the second argument.

Table 4.5: The Math Built-Ins

- *Built-Ins for Strings*

These built-ins are specifically designed for the string manipulation. They cannot be used with untyped literals with language tags. W3C (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004) has listed following built-ins for string manipulation.

Built-ins	Description
swrlb:stringEqualIgnoreCase	Satisfied iff the first argument is the same as the second argument (upper/lower case ignored)
swrlb:stringConcat	Satisfied iff the first argument is equal to the string resulting from the concatenation of the strings the second argument through the last argument.
swrlb:substring	Satisfied iff the first argument is equal to the substring of optional length the fourth argument starting at character offset the third argument in the string the second argument.

swrlb:stringLength	Satisfied iff the first argument is equal to the length of the second argument.
swrlb:normalizeSpace	Satisfied iff the first argument is equal to the whitespace-normalized value of the second argument.
swrlb:upperCase	Satisfied iff the first argument is equal to the upper-cased value of the second argument.
swrlb:lowerCase	Satisfied iff the first argument is equal to the lower-cased value of the second argument.
swrlb:translate	Satisfied iff the first argument is equal to the second argument with occurrences of characters contained in the third argument replaced by the character at the corresponding position in the string the fourth argument.
swrlb:contains	Satisfied iff the first argument contains the second argument (case sensitive).
swrlb:containsIgnoreCase	Satisfied iff the first argument contains the second argument (case ignored).
swrlb:startsWith	Satisfied iff the first argument starts with the second argument.
swrlb:endsWith	Satisfied iff the first argument ends with the second argument.
swrlb:substringBefore	Satisfied iff the first argument is the characters of the second argument that precede the characters of the third argument.
swrlb:substringAfter	Satisfied iff the first argument is the characters of the second argument that follow the characters of the third argument.
swrlb:matches	Satisfied iff the first argument matches the regular expression the second argument.
swrlb:replace	Satisfied iff the first argument is equal to the value of the second argument with every substring matched by the regular expression the third argument replaced by the replacement string the fourth argument.
swrlb:tokenize	Satisfied iff the first argument is a sequence of one or more strings whose values are substrings of the second argument separated by substrings that match the regular expression the third argument.

Table 4.6: The Built-Ins for Strings

Built-ins in the SWRL can be used with the standard SWRL expressions. The built-ins process the rule expressions to deduce the result and couple with the standard expression to return the results. For example

```
Person(?x) ^ hasHeight(?x, ?h) ^ swrlb:greaterThan(?h, 6.5) → Tall(?x)
```

Here the expression `swrlb:greaterThan(?h, 6.5)` compares the value of the heights of each individual of the class person with the threshold value of 6.5 to determine him as tall. In the same manner other built-ins could be used along with the regular SWRL expression with the reserved word *swrlb* – standing for SWRL Built-ins.

4.3. Discussion

The Semantic Web, a set of technologies complementing the conventional Web tools proposed by Sir Tim Berners-Lee is seen as the most probabilistic approach to reach the goal of semantic interoperability. The Semantic Web is envisaged as an extension to the existing Web from a linked document repository into the platform where information is provided with the semantic allowing better cooperation between people and their machines. This is to be achieved by augmenting the existing layout information with semantic annotations that add descriptive terms to Web content, with meaning of such terms being defined in ontologies (Horrocks, Pater-Schneider, McGuinness, & Welty, 2004). Ontologies play crucial role in conceptualizing a domain and thus play an important role in enabling Web-based knowledge processing, sharing and reuse between applications.

This research takes advantages of the tools of Semantic Web technology to make a case of information management through knowledge. The case study of Industrial Archaeology fits perfectly to put forward the concept of information handling through knowledge as the domain generates huge and heterogeneous dataset. In addition the sites are not preserved for continuing excavation as in case of the conventional archaeology, making it ideal for utilizing knowledge techniques to manage the information because of the flexibility in knowledge techniques to handle information long after they are collected. The definition of a domain ontology representing the site is sketched out by the archaeologists. It is again their task to fill in knowledge in the domain ontology to make it a knowledge base where one can reason to derive new knowledge. Archaeologists use collaborative Web platform based on Semantic Web technology to identify the objects and define them in the ontology. These objects once defined, performs as common schemas between data sources to achieve a sense of data interoperability. The definitions of objects add semantics to the objects and thus adding knowledge about the objects. Knowledge techniques based on Description Logics (DLs) exploit these semantics manipulate implicit knowledge within the knowledge base. Inference engines utilize the definition of DLs to infer the knowledge base through Horn based rules.

The knowledge based stored in OWL syntactic structure is inferred through SWRL to infer the rules. This inference is complimented through querying with SPARQL.

Carrying the discussion from last chapter, this research attempts to use the Semantic Web techniques to perform spatial analysis in form of spatial SPARQL and spatial SWRL. The spatial analysis through Semantic Web can only be possible through providing spatial signatures to the defined objects in the ontology. This will allow the knowledge techniques to process spatial solutions. The spatial integration is carried out through OWL/RDF again and the spatial management is carried out again through tools as SWRL and SPARQL. This simplistic yet but effective approach of spatial integration into Semantic Web technologies provides the possibility to include different modes of data into its framework.

The Semantic Web stack shown in figure 4.5 and 4.6 can adjust a layer of spatial information into it. The research proposes such an arrangement in the stack. A layer of spatial data mixing seamlessly with the semantic proposition in the layer Ontology through its OWL/RDF based syntax can be envisaged. This layer since uses the standard syntax of OWL/RDF can perform spatial queries through SPARQL or infer rules through standards as SWRL. The next chapter discusses this integration process of spatial technology and Semantic Web technology which is undertaken by defining spatial FILTERs for SPARQL queries and spatial Built-ins for SWRL rules. Ideally the layer should be the top most layer of knowledge level but spatial layer does not yet possess any standards that are standardized by W3C so could not be placed there. It is hence placed as the bottom layer in the certificate level. The next chapter discusses this adjustment in stack in detail and how to apply spatial queries and rules on any existing ontology.

References

Al-Feel, H., Koutb, M. A., & Suoror, H. (2009). Toward An Agreement on Semantic Web Architecture. *WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY* (49), 806-810.

Antoniou, G., & Harmelen, F. v. (2003). Web Ontology Language: OWL. In *Handbook on Ontologies in Information Systems* (pp. 67 - 92). Springer Series.

Baader, F., & Nutt, W. (2002). Basic Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, & P. F. Patel-Schneider, *Description Logic Handbook* (pp. 47 - 100). Cambridge: Cambridge University Press.

Baader, F., & Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica* , 5 - 40.

Baader, F., Calvanese , D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2003). *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge: Cambridge University Press.

Bai, J., Kendall, E. F., McGuinness, D. L., & Patel-Schneider, P. (2009, October 27). *OWL 2 Web Ontology Language: Quick Reference Guide*. Retrieved October 28, 2010, from W3C Recommendation: http://www.w3.org/TR/2009/REC-owl2-quick-reference-20091027/

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., et al. (2004, February 10). *OWL Web Ontology Language.* Retrieved November 27, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-ref/

Beckett, D., & Berners-Lee, T. (2008, January 14). *Turtle - Terse RDF Triple Language*. Retrieved October 28, 2010, from W3C Team Submission: http://www.w3.org/TeamSubmission/turtle/

Berners-Lee, T. (1998, October 14). *Why RDF model is different from the XML model*. Retrieved October 2010, 22, from W3C: A roadmap to the Semantic Web: http://www.w3.org/DesignIssues/Overview.html

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific AmericaN* , pp. 34-43.

Bobillo, F., & Straccia, U. (2009). Supporting Fuzzy Rough Sets in Fuzzy Description Logics. *10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. 5590/2009*, pp. 676-687. Verona: LNCS.

Boley, H. (2009, August 11). *RuleML: The Rule Markup Initiative*. Retrieved November 27, 2009, from RuleMLL: http://ruleml.org/

Boley, H., & kifer, M. (2010, June 22). *W3C Recommendation* . Retrieved August 24, 2010, from RIF Basic Logic Dialect: http://www.w3.org/TR/rif-bld/

Calvanese, D., Giacomo, G. D., Lenzerini, M., & Nardi, D. (2001). Reasoning in expressive description logics. In A. Robinson, & A. Voronkov (Eds.), *Handbook of Automated Reasoning* (Vol. II, pp. 1581-1634). Elsevier Science Publishers.

Cormode, G., & Krishnamurthy, B. (2008). *Key Differences between Web1.0 and Web2.* Florham Park: AT&T Labs-Research.

da Silva, P. P., McGuinness, D. L., & Fikes, R. E. (2004). *A Proof Markup Language for Semantic Web Services.* Stanford: Stanford University.

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is Knowledge Representation. *AI Magazine , 14* (1), 17 - 33.

Fensel, D., Horrocks, I., Harmelen, F. v., McGuinnes, D. L., & Patel-Schneider, P. (2001). OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems* , 38 - 45.

Gerber, A., Merwe, A., & Barnard, A. (2008). A functional Semantic Web architecture. *Proceedings of the 5th European Semantic Web conference on The Semantic Web: research and applications* (pp. 273 -287). Tenerife: Springer-Verlag.

Goebel, R., Zilles, S., Ringlstetter, C., Dengel, A., & Grimnes, G. (2008). What is the Role of Semantic Layer Cake for Guiding the Use of Knowledge Representation and Machine Learning in the Development of the Semantic Web? *Fourth Artificial Intelligence for Interactive Digital Entertainment Conference.* Chicago: AAAI 2008.

Harlmelen, F. v., Patel-Schneider, P., & Horrocks, I. (2001, March 27). *A model-theoretic semantic for DAML+OIL.* Retrieved August 30, 2010, from http://www.daml.org/2001/03/model-theoretic-semantics.html

Harris, S., & Seaborne, A. (2010, January 26). *SPARQL Query Language 1.1*. Retrieved September 2, 2010, from W3C Working Draft: http://www.w3.org/TR/2010/WD-sparql11-query-20100126/

Hayes, P., & McBride, B. (2004, February 10). *RDF Semantics*. Retrieved August 26, 2010, from W3C Recommendation: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

Hendler, J., & McGuinness, D. L. (2000). The darpa agent markup language. *IEEE Intelligent System , 15* (6), 67 -73.

Herman, I., Halpin, H., Hawke, S., Prud'hommeaux, E., Raggett, D., & Swick, R. (2010, February 06). *W3C Semantic Web* . Retrieved August 15, 2010, from W3C Semantic Web Activity: http://www.w3.org/2001/sw/

Horrocks, I., Parsia, B., Patel-Schneider, P., & Hendler, J. (2005). Semantic Web Architecture: Stack or Two Towers. *3rd Internation Workshop on Principles and Practice of Semantic Web Reasoning* (pp. 37 - 41). Dagstuhl: LNCS .

Horrocks, I., Pater-Schneider, P. F., McGuinness, D. L., & Welty, C. A. *OWL: a Description Logic Based Ontology Language for the Semantic Web.*

Horrocks, I., Schneider, P. f., Boley, H., Tabelt, S., Grosof, B., & Dean, M. (2004, December 21). *SWRL - A Semantic Web Rule Language - Combining OWL and RuleML*. Retrieved May 22, 2009, from SWRL: A Semantic Web Rule Language - Combining OWL and RuleML: http://www.w3.org/Submission/SWRL/

Klyne, G., Caroll, J. J., & McBride, B. (2004, February 10). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. Retrieved August 26, 2010, from W3C Recommendation: http://www.w3.org/TR/rdf-concepts/

Lassila, O., & Swick, R. (1999, February 22). *Resource Description Framework (RDF) Model and Syntax Specification*. Retrieved August 29, 2010, from W3C Recommendation: p://www.w3.org/TR/REC-rdf-syntax/

Mallery, J. (1994). A Common LISP hypermedia server. *Proc. First International World-Wide Web Conference*, (pp. 239 - 247). Geneva.

Mazzocchi, S. (2000). Toward the Semantic Web: a view of XML from outer space. *Proceeding: ApacheCON 2000.* London: Apache.

McGuinness, D. L. (2001). *Description Logics Emerge from Ivory Towers.* Standford: Standford University.

McGuinness, D. L., & Harmelen, F. v. (2004, February 10). *OWL Web Ontologgy Language: Overview*. Retrieved December 2, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-features/

Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokou, A., & Lutz, C. (2009, October 27). *OWL 2 Web Ontology Language Profiles*. Retrieved October 28, 2010, from W3C Recommendation: http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/

Motik, B., Patel-Schneider, P. F., & Grau, B. C. (2009, October 27). *OWL 2 Web Ontology Language: Direct Semantics* . Retrieved October 28, 2010, from W3C Recommendation: http://www.w3.org/TR/2009/REC-owl2-direct-semantics-20091027/

Prud'hommeaux, E., & Seaborne, A. (2008, January 2008). *SPARQL Query Language for RDF*. Retrieved May 22, 2010, from http://www.w3.org/TR/rdf-sparql-query/

Schmidt-Schauß, M., & Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence , 48*, 1 - 26.

Shannon, V. (2006, May 23). *The New York Times - Technology.* Retrieved August 16, 2010, from A 'more revolutionary' Web: http://www.nytimes.com/2006/05/23/technology/23iht-Web.html

Sheth, A. P. (1999). Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In *Interoperating Geographic Information Systems* (pp. 5 - 30). MA: Kulewar Academic Publishers.

Simpson, S. G. (1998). *Math 563: Model Theory.* Pennsylvania: Pennsylvania State University.

Smith, M. K., Welty, C., & McGuinness, D. L. (2004, February 10). *OWL Web Ontology Language.* Retrieved November 17, 2009, from W3C Recommendation: http://www.w3.org/TR/owl-guide/

Sowa, J. F. (2006, February 02). *Semantic Networks*. Retrieved October 27, 2010, from http://www.jfsowa.com/pubs/semnet.htm

Spaccapietra, S., Cullot, N., Parent, C., & Vangenot, C. (2004). On Spatial Ontologies. *GEOINFO* . Campos do Jordao.

Sterling, L., & Shapiro, S. (1994). *The Art of Prolog.* MIT Press.

W3C OWL Working Group. (2009, October 27). *OWL 2 Web Ontology Language: Document Overview*. Retrieved October 28, 2010, from W3C Recommendation : http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

Welty, C. (1996). Intelligent assistance for navigating the Web. *Proc of the 1996 Florida AI Research Symposium* . Florida: AAAI Press/MIT Press.

Welty, C., & Jenkins, J. (2000). Untangle: a new ontology for card catalog systems. *Proc of 17th National Conference on Artificial Intelligence (AAAI 2000)* (pp. 1137 - 1138). Austin: The MIT Press.

Chapter 5

THE SPATIAL LAYER AND ITS TOOLS: The Spatial Facilitator

This chapter presents the integration process of spatial technologies and the Semantic Web technologies at the backdrop of Industrial Archaeology, and its associated tool called the spatial facilitator which is a query and rule engine. The technologies discussed in previous chapters are used and adjusted for processing the spatial knowledge through knowledge technologies within the Semantic Web framework in the research works. This chapter attempts to outline the methods and the processes of these adjustments and how they return the results through knowledge tools as SWRL and SPARQL.

The discussions of the last two chapters aim at laying a background on the concepts of integration process. The discussions on Semantic Web and its underlying technologies and the spatial technology in GIS in the last two chapters have clearly pointed out that the technical advancements toward semantic technologies are integrating every data structures so it will integrate spatial data structure in future. However, for now it is still a topic of research. It could be conceived from earlier discussions that the integration process requires adjustments of the spatial components within the ontological framework from the discussions. This chapter is dedicated to discuss the steps and process of this adjustment. The spatial signature of objects plays an important role in determining them. The identification of objects is the process of signing these spatial signatures on them. These signatures should be integrated within the semantics of the objects seamlessly in order to process the spatial knowledge through the knowledge technology. It should be noted however that the Semantic Web technologies are in the maturation process and hence there exists certain processing problems within especially for the non-conventional data type as that of spatial data. Thus, it needs to be sorted out through the existing tested techniques. The research in GIS systems uses the capabilities of existing RDBMS to process the spatial data through spatial operations and functions and used the results of these processes.

This chapter begins with the presentation of the integration of the semantic spatial technologies with respect to the Semantic Web Stack discussed in previous section. In section 2, the top ontology defines the upper level concepts for the integration process and the spatial facilitator tool is presented. These high level concepts and relationships are specialized in order to define the features and finding of the case study in the domain of Industrial archaeology. The top ontology is composed of four major top level concepts `feat:Feature`, `shp:shape`, `doc:document`, `tag:tag`, and two top level properties `feat:objRel` and `att:attribute`. These concepts and properties are involved to process the indexing of documents and spatial shapes with an ontology adjusted. Section 3 presents the principles of the ontology adjustment processed on an existing ontology in order to integrate spatial properties for spatial queries and inference rules which are carried out by the translation engine. Section 4 presents this translation engine which uses the SPARQL engine and the SWRL engine for spatial retrievals and deductions. The last section concludes this chapter.

5.1. Revisiting the Semantic Web Stack

The Semantic Web stack discussed in the previous chapter can be updated to address the inclusion of a spatial component. Every tangible object has its spatial signature and thus it becomes indispensable to address the spatial component within its semantic framework. The Semantic Web technologies and its architecture are mostly influenced by the nature of information available on the Internet. Hence, these levels deals mostly with managing the semantic based information through knowledge technologies. However, in recent years there has been huge surge of other forms of information on Web platform and they need to be managed as well. With the advancement in spatial technologies, the trend of disseminating spatial information through Web based environment is rapidly growing. This has raised the issue of the integration of spatial component into the Semantic Web framework.
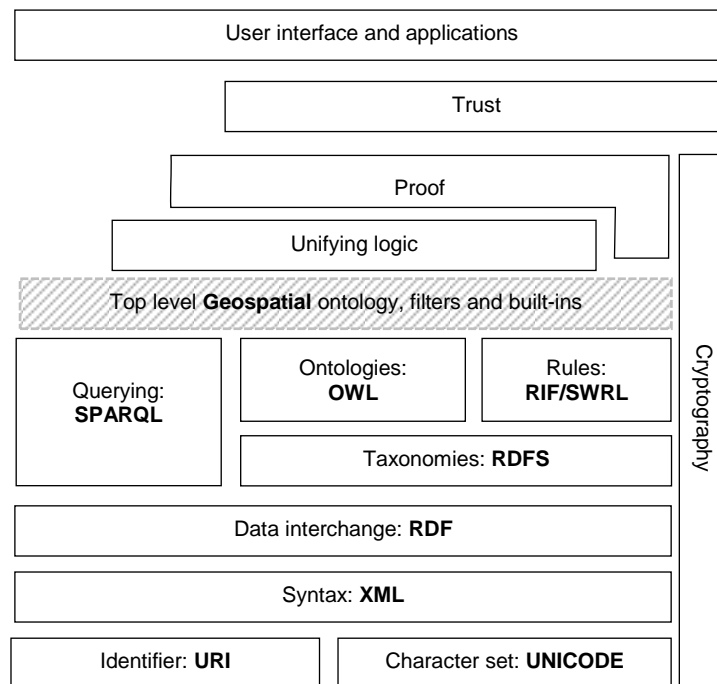
Figure 5.1: The inclusion of a Spatial layer in the Semantic Web Stack

A layer representing geospatial data in the Semantic Web stack can be placed just above the knowledge layers as could be seen in figure 5.1. As the technologies within knowledge level are standardized by W3C, the geospatial layer needs to be above the level. However, the technologies within knowledge level needs to blend spatial components seamlessly both syntactically and semantically to maintain the satisfiability required for the consistency of the ontology. This integration procedure should be adjusted within the knowledge tool within the

knowledge level of the stack. This approach thus uses the knowledge techniques through adding the spatial structures within them and implementing the spatial knowledge processing along with semantic knowledge processing. The first Semantic Web tool that comes direct in contact with the integration procedure is the structural schema of the knowledge base which is termed as top level ontology in general sense. The top level ontology is the structural schema that represents the nature of knowledge the ontology possesses. It should include the components to adjust the behavior of the knowledge base. Hence the initial task that needs to be adjusted within any top level ontology to perform spatial knowledge processing is to include spatial components within it.

The top level ontology is syntactically presented through OWL/RDF and contains the top level concepts of the domain. Among these top level concepts, the concepts presenting the spatial components store, retrieve and process the spatial knowledge. In each of these processing steps the knowledge technologies are used in combination. It starts with the spatial components in the top level ontology that is encoded through OWL/RDF syntax. Moving down to the enrichment process, the spatial signatures are mapped to the objects within the knowledge base is again encoded with OWL/RDF syntax. The methodology of this integration is discussed in later sections within this chapter.

Similarly, the filters and built-ins defined in this layer facilitate the spatial querying and the spatial rule definition through spatial filters and spatial built-ins. The layers of Rules: SWRL/RIF and Querying: SPARQL provide a base to knowledge management through processing the spatial information semantically within the knowledge base. The only adjustment that is needed is to execute the built-ins and filters in conjunction to the processing capabilities of spatial extensions within current database systems.

Lastly, defending the authenticity of the layer with respect to other layers, the geospatial layer exploits the capabilities of the layers below maintaining the trend of the stack. At the time of integration, the spatial components are included within the top level ontology which stores, retrieves and processes spatial knowledge and utilizes the capabilities of the technologies in the stack. The spatial components on the top level ontology and the mapped spatial signatures are encoded through the OWL/RDF syntactical structure thus justifying the involvement of ontologies in the stack. Then after, the capability of the SWRL language is exploited through spatial built-ins for spatial SWRL rules. Similarly, the querying capability of the SPARQL

language is exploited through spatial filters for the query language. These filters can be used with conjunction to already standardized built-ins with both the technologies thus forwarding the arguments of the process in standardizing these built-ins too.

5.2. The Top Level Ontology

The top level ontology or more popularly upper ontology describes the general concept behind the knowledge domain. This ontology varies with the domain it addresses. There are efforts to come out with a universal upper ontology which addresses the requirements of every knowledge domains but they still are in the phase of researches. Every domain uses its own standard upper ontology for its purpose. This thesis research attempts to propose an upper level ontology for the domain of industrial archaeology. This top level ontology is the main driving force behind the ArchaeoKM framework. It represents the knowledge possessed by archaeologists in form of descriptions, observations and rules represented through different axioms within the ontology. This ontology serves as a foundation ontology to which objects can be instantiated during identification process. The axioms are the building blocks of the ontology and hence these axioms in the context of the top level ontology of the application should be discussed to provide an overview of the system. This section discusses the top level ontology first in terms of its axioms and later continues to describe these axioms in accordance to the functionalities of the ArchaeoKM framework.

It should be noted that this top level ontology fulfills the necessity of the archaeologists to document their knowledge. It is just an example to demonstrate the functionalities of knowledge processing within Semantic Web framework. The integration of spatial components within the framework holds major importance and is a part of discussion. This section emphasizes on the example of using the top level ontology to document the knowledge possessed by the archaeologists and then in the next section demonstrate how it could be extended to include the spatial components within. The example could be carried to other upper ontology for the integration process as will be shown in wine ontology later in this chapter.

5.2.1. Components

The top level ontology constitutes mainly the classes and their relationships. Hence this section discusses these components in terms of axioms representing them.

5.2.1.1. *Class Axioms*

Every tangible and non-tangible object which needs to be instantiated to store information in some level is collectively represented through Class Axioms. They are basically building blocks of the domain ontology storing the objects excavated from the sites to the information needed for proper functionalities of the features within the application tool. The class axioms contain their own prefixes which are used to define their names. One of the big advantages of using prefix is that the same class could be used by applying different prefix for the class. Other advantages include the simplification in defining the resource and to solve the ambiguity for different context. The hierarchical structure of the top level class axioms of the ontology is given in figure 5.2.
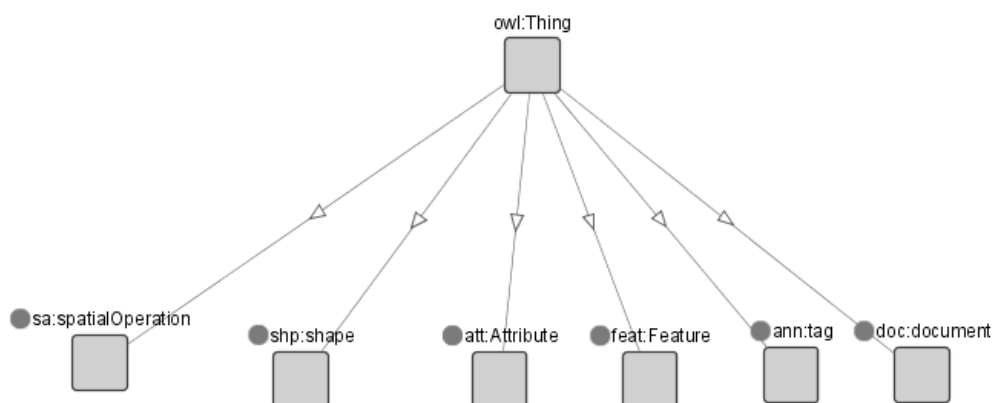
Figure 5.2: The class axioms of top level ontology

At the heart of the ontology is the class axiom `feat:Feature` which represents the objects. In this project these features represent the objects found in the excavation site. This class axiom is the generalized class of any objects found in the site. This class is further specialized into classes representing the different objects discovered. The figure 5.3 illustrates the `feat:Feature` into its specialized classes. Archaeologists were involved to use their archaeological knowledge for listing out the subclasses. This class is the only interactive class as they are allowed to create individuals for the identified objects within the subclasses of this

class. For instance, a feature wall `feat:Wall` with its own properties will be defined as a subclass of the class `feat:Feature.`

The next important class axiom is `shp:shape` which stores the local coordinates of the objects identified in the excavation site. This generalized class is specialized into `shp:_3D` and `shp:_2D` sub classes to represent the dimensions of the coordinates. Currently, an *orthophoto* is used to identify objects on a map and hence the 2D coordinates are returned of the objects.

The part `doc:document` is related to the management of the corpus of multimedia documents. Every kind of digital document can be linked to a specific individual of the ontology in the document part. The properties of the individual are used to identify the documents. The class `doc:image` is a subclass of `doc:document` that allows the identification of an image in a file system. This image has also an image type which is constrained for the moment by only one of the following value "jpeg", "tiff", "gif". This can be easily extended. The name is the name of the image file. In similar fashion class axiom `doc:archive` stores the archive documents which are digitized and stored. The principles in managing such documents are similar to management of image documents. The other specialized class `doc:spatialDoc` stores the spatial information of the object. They will be discussed in details in next section when the class axioms are discussed in terms of their functionalities.

The class axiom `ann:tag` stores annotation information through the individuals of its subclasses. Again, this axiom has three specialized classes each for annotating three different data types: image files, archive documents and spatial data. This class axiom is closely related to the management of documents discussed earlier.

The class axiom `att:Attribute` presents different types of attribute properties possible. Those properties are classified as subclasses of the axiom and the characteristics of the properties as individuals. Simplifying, if the color is an attribute property possible in the objects that are excavated, they are represented through `att:Color` within the ontology and different types of color like Red, Blue, Green etc. are represented through individuals `att:Red` and so on.

Lastly class axiom `sa:spatialOperation` stores the results of the spatial operation performed during the spatial knowledge management process. This will again be dealt with details in the sections discussion spatial integration in details.

5.2.1.2. *Property Axioms*

These axioms define relationships between different classes in the ontology. The majority of properties are defined through the interactive involvement of the archaeologists. These properties are based upon their experiences through observations in the field and the archaeological notes they have drawn during the excavation process. However, certain properties are defined to support the proper functionalities with the application tool. There are five major object properties axioms in the top level: `feat:objRel`, `shp:hasShape`, `ann:hasAnnotation`, `att:hasAttribute` and `sa:hasSpatialRelations`.

At the heart of these relationships is the object property `feat:objRel` belongs to the group of properties that are defined through the involvement of the archaeologists. They are best informed about the relationships between objects and they decide which objects could be related through which object properties. However, in general the specialized properties within `feat:objRel` are related to the individuals under specialized classes of `feat:Feature` to themselves.

Another object property which has been influenced through the observations of archaeologists is `att:hasAttribute`. This property relates the objects identified to their attributive properties. An example could be an instance of `feat:Wall` subclass of `feat:Feature att:hasMaterial att:brick` which is an instance of `att:Material` subclass of `att:Attribute`.

The other object properties require lesser involvements of the archaeologists. The object properties under `shp:hasShape` link the identified objects to their location coordinates stored as instances under class `shp:_2D` or `shp:_3D` which are subclasses of `shp:Shape`. Likewise object properties under `ann:hasAnnotation` relate objects to their relevant data and documents through mapping the data and documents under `doc:Document` to `feat:Feature` through the intermediate class `ann:tag` and object property `doc:hasDocument`.

Lastly, a major object property `sa:hasSpatialRelations` relates the objects in `feat:Feature` against each other through some spatial relationships. It also relates the `feat:Feature` to the results of operations in `sa:SpatialOperation` class. This will be covered in detail again in spatial integration section of this chapter.

5.2.2. Functionalities

The application of the ArchaeoKM framework is primarily based on three general foundations dedicated to their genre of functionalities. The top level axioms are designed to facilitate these functionalities. Theses functionalities are the semantic definition of features, the semantic annotation of documents and the semantic indexation of spatial signatures.

5.2.2.1. Semantic Definitions of Features

The first and far most important task in any knowledge acquisition process is to identify the objects to which knowledge is related to. These objects can be both tangible and non-tangible. In case they are tangible, they need to have spatial signatures attached to them. It is hence possible to localize them through tagging in their respective domain. The objects once identified and tagged will enriches the knowledge base. Then is the time to attach semantics to them. This process is carried out through the collaborative knowledge of archaeologists. They attach these semantic through proper arrangement of the axioms in the top level ontology. This part presents the semantic definitions of the features identified. This part also introduces spatial signatures of the objects; however the details on spatial linkage in semantic definition are covered in part the Semantic Indexation of Spatial Signatures.

5.2.2.1.1 Classes

Two major classes are involved managing semantic information of the object. The subclasses of class `feat:Feature` stores the objects identified as their instances. The spatial signatures of these objects are stored in their relevant type under class `shp:_2D` which is a subclass of class `shp:Shape`. These two classes are classified to adjust the natures of the objects identified and their natures of the geometries. The figure 5.3 illustrates the hierarchical structures of class axioms of `feat:Feature`.
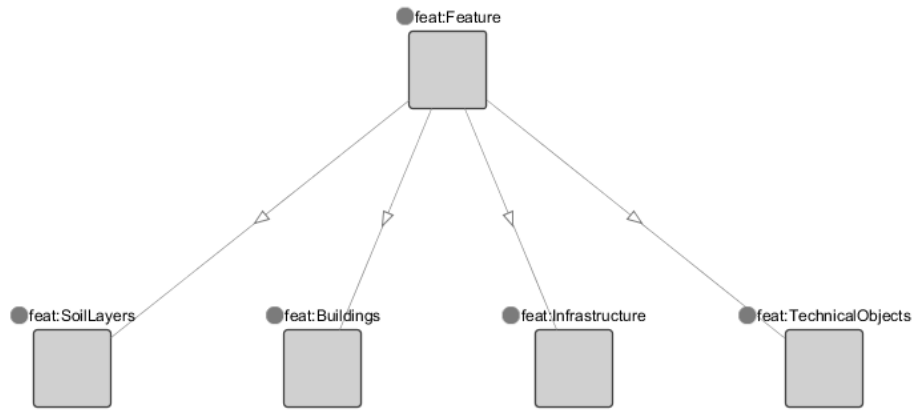
Figure 5.3: The hierarchical Structure of `feat:Feature`

As could be seen there are four subclasses and each of these subclasses can be broken down further to store the objects in their most appropriate class. Likewise the hierarchical structure of class `shp:Shape` is illustrated in figure 5.4.
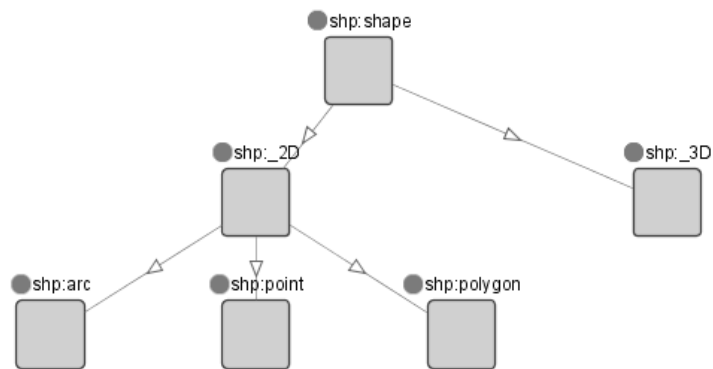
Figure 5.4: The hierarchical Structure of `shp:Shape`

It could be seen that the `shp:_3D` does not have the sub classes assign to it and `shp:_2D` consists of three subclasses as `shp:arc`, `shp:point` and `shp:polygon`. The main reason behind the arrangement is to keep in line with the standards of simple geometry specification of OGC (Herring, 2010). They are yet to standardize 3D geometries so the top level ontology provides the provision of adding the shapes of 3D geometry in future but has not provided one.

5.2.2.1.2 Relationships

Semantics of objects in the knowledge base are defined through object property `feat:objRel`. But before that, they need to relate to their spatial signature that is to their

coordinates. This is managed through the specialized object property of `shp:hasShape`. As mentioned the coordinates of the object is derived through the digitization are stored as an individual of `shp:_2D`. This instance stores the coordinate of object. Once both the object and its coordinates are populated, `shp:hasShape` provides a relationship between them. Figure 5.5 summarizes such relationship.
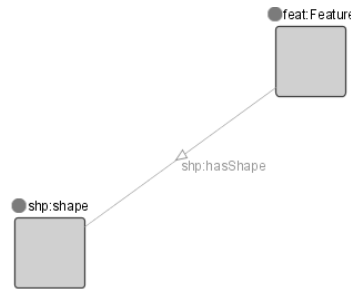

Figure 5.5: The Object Property `shp:hasShape` relating objects to their coordinates

The object property `feat:objRel` is specialized object property of all the possible semantic relationships that an object can have with others. Hence, here the objects in the specialized classes in `feat:Feature` can possess such relationships with the objects in specialized class of the same class. For instance, if `feat:dining` is an instance of `feat:room` (which is a specialized class of `feat:Feature`) `feat:hasTable` (which is a specialized object property of `feat:objRel`) as `feat:diningTable` (an instance of `feat:Table` and the specialized class of `feat:Feature`), then it shows whatever the conditions or situations may be, the `feat:objRel` has relations from and to `feat:Feature`. Figure 6.6 presents such relationship.
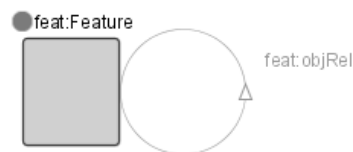
Figure 5.6: The Object Property `feat:objRel` relating objects to other objects

5.2.2.1.3 Attributes

Attributes are managed in two ways in the ArchaeoKM framework. The first types of attributes are simple attributes in a sense that are directly represented through certain data types. These attributes have only one node and the next points to the data types. Attributes as `feat:height, feat:width, feat:length` store the dimensions of the objects. In the same

way the `feat:localPlacement` stores the coordinate values is also a data property. Similarly, the description of the object is stored within `feat:description`. These attributive data are managed through data property in OWL.

The next set of attributes describes the nature of the object through their attributive behaviors. These behaviors could present the nature of the object either individually or through the collective behaviors of other attributes. For example a case of "the library building has color red" is considered. In this case, the attributive behavior of color is not dependent to any other attributes as the color is red. It thus can directly map the individual `feat:libraryBuilding` (individual of class `feat:Building`) to individual `att:Red` (individual of class `feat:Color`) through object property (`att:hasColor`). The next case is dependent on the behavior of other attribute. A case of "library building has construction of steel Framework". This means the construction is a framework which is made up of steel. So a possibility is suggested to map library to steel through first object property `att:hasConstruction` and then `att:hasMaterial`. This could be presented through the statements like `feat:libraryBuilding` (individual of `feat:Building`) `att:hasConstruction` `att:steelConstruction` (individual of `att:Construction`) and then `att:steelConstruction` (individual of `att:Construction`) `att:hasMaterial` `att:Steel` (individual of `att:Material`). The top level ontology provides an open window to incorporate such behaviors of attributes through the schema which maps the general concepts of `feat:Feature` to `att:Attribute` through generalized object property `att:hasAttribute` to address the first scenario. Likewise the `att:Attribute` is mapped to itself through object property `att:hasAttribute` to address second scenario. This is illustrated in figure 5.7. The top level ontology has left this open as this possibility could provide detailed knowledge processing in future.
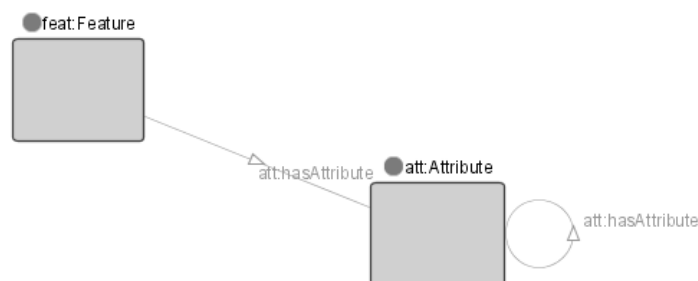

Figure 5.7: The relationship with attributes through object property `att:hasAttribute`

5.2.2.2. *Semantic Annotation of Documents*

In general semantic annotation is a process of attaching meaning to a subject. This can be carried out through various ways. Parts of semantic annotation activities are already carried out during describing the objects through `feat:description` during the enrichment process. This part covers the semantic annotation process of data and documents in accordance to their relationships to the objects populated.

It has already been discussed in chapter 2 that huge amount of data and documents are collected during the excavation process.The data and documents vary in their structures and in their formats with each other. In most cases, it is difficult to find a common background to link these data and documents together. These data and documents also require experience eyes to connect them to the objects excavated. It is thus necessary to use the knowledge of archaeologists to link them to the appropriate objects. In process they also define the data and documents in accordance to their own natures and natures related to the objects with which they have been related to. The ArchaeoKM framework provides the functionalities to facilitate archaeologists to annotate these data and documents semantically through the underlying ontology. The top level axioms provide the needed framework for the annotating process. It functions as a global schema for the data and documents to provide a common platform to link them together. This thus provides a sense of data interoperability.

5.2.2.2.1 Classes

Two major classes have crucial roles to provide the annotation to the documents: `feat:Feature, doc:document`. A third class `ann:tag` provides a bridge between these two classes. The data and document collected can broadly be divided into file systems and database systems. The subclasses of `doc:document` complements this through subclasses `doc:fileSystem` and `doc:RDBMS`. The complete hierarchy of the class `doc:document` can be seen in figure 5.8.
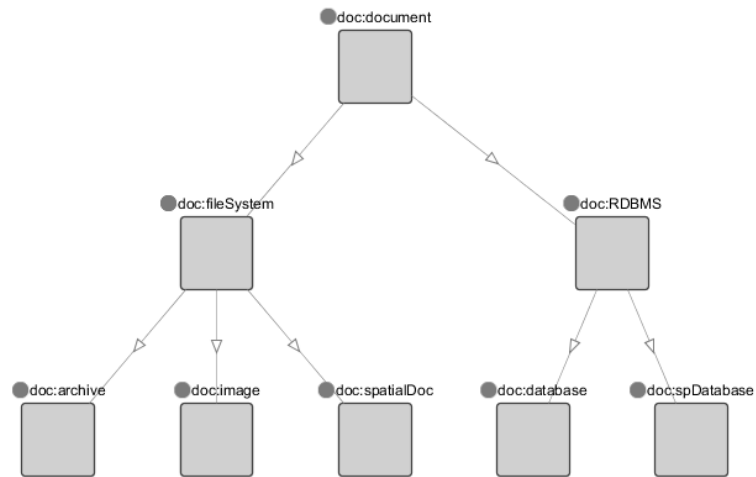
Figure 5.8: The Class Hierarchy of the class `doc:document`

The leaf nodes of hierarchy illustrated in figure 5.8 stores the documents and data. First the data and documents are instantiated to their relevant classes within the subclasses of `doc:document`. This instantiation provides a framework for data and documents to be annotated semantically. The actual semantic annotation process is carried out through linking them to the relevant objects and assigning these data and documents to their semantic descriptions. They are undertaken through assigning proper relationships to the data and documents.

The next class `ann:tag` is a bridge class linking the instances storing data and documents to their respective objects in the knowledge base. Hence, its subclasses should reflect the leaf classes in the class hierarchy of class `doc:document`. Its class hierarchy is illustrated in the figure 5.9 and an example is given in the figure 5.10.
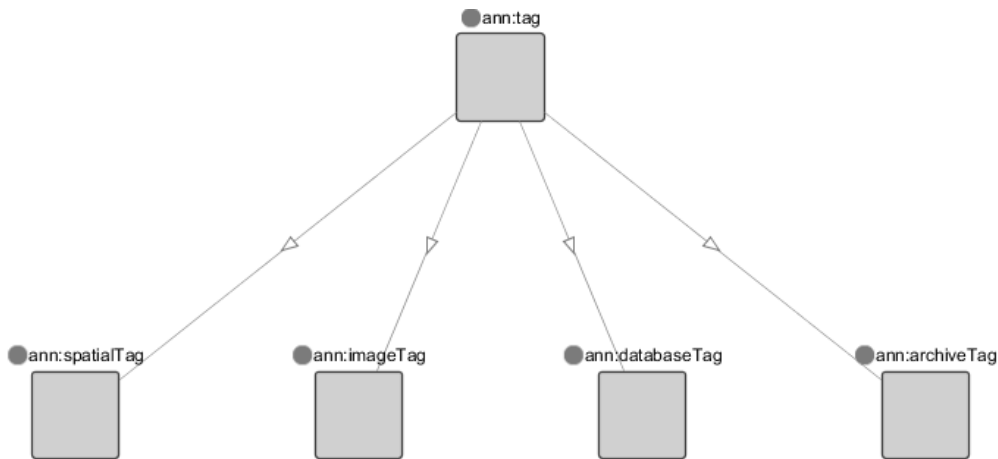
Figure 5.9: The Class Hierarchy of the class `ann:tag`

5.2.2.2.2 Relationship

The ontology in the ArchaeoKM framework has dedicated two object properties to associate data and documents to the objects in the knowledge base semantically. These object properties are `ann:has Annotation` and its specialized properties in `ann:hasArchiveAnno`, `ann:hasImageAnno` and `ann:hasSpatialAnno` and `doc:hasDocument` and its specialized properties in `doc:hasArchive, doc:hasDatabase, doc:hasImage` and `doc:hasSpatial`.

The first process of annotating the data and documents is to link them to the proper objects in the knowledge base. It is carried out through the combination of three top level class axioms (`feat:Feature, ann:tag` and `doc:Document`) and two top level object property axioms (`ann:hasAnnotation` and `doc:hasDocument`). As soon as an object is set to be annotated, an instance within the relevant subclasses of class `ann:tag` is created. This instance is connected to the object through `ann:tag`. Similarly, it maps to relevant instances of `doc:Document` through `doc:hasDocument`. It thus annotates the documents semantically by mapping them to their respective objects. The process is illustrated in figure 5.10.

Figure 5.10: The semantic mapping of documents to features

5.2.2.2.3 Attributes

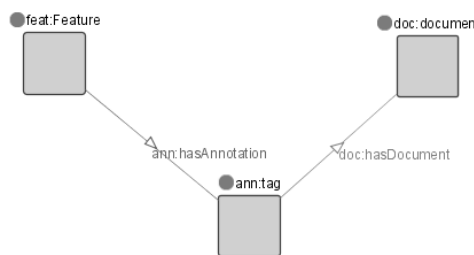The semantic annotations of the data and documents which have been instantiated under `doc:Document` are carried out through assigning properties. They are mostly done through assigning the attributes to them. The data properties within the ontology are designed in such a way that the documents are assigned properties which reflect the nature of themselves and also those properties which reflect the nature of their association to the objects. For example an image file have properties as the file format, the photographer, camera type as the properties of the image itself and the side of the object, the distance of the camera from the object when the picture was taken, etc. are the properties in accordance to the relation to the object.

The data property reflecting the semantics of the data and document can thus be divided into two broader categories. The first is the semantics of the documents and next is semantics of the mapping. The ArchaeoKM framework applies this through the data property `doc:hasDocumentDetails` and `ann:hasAnnoDetails` where the first data property has domain as `doc:Document` specifying the documents properties and second one has domain ann:tag thus specifying the properties of mapping.

5.2.2.3. Semantic Indexation of Spatial Signatures

The previous two parts provide outline of the underlying ontology of the ArchaeoKM framework. Figure 5.11 illustrates the basic structure of the ontology which summarizes the structural behavior of the ontology that has been discussed in those parts.
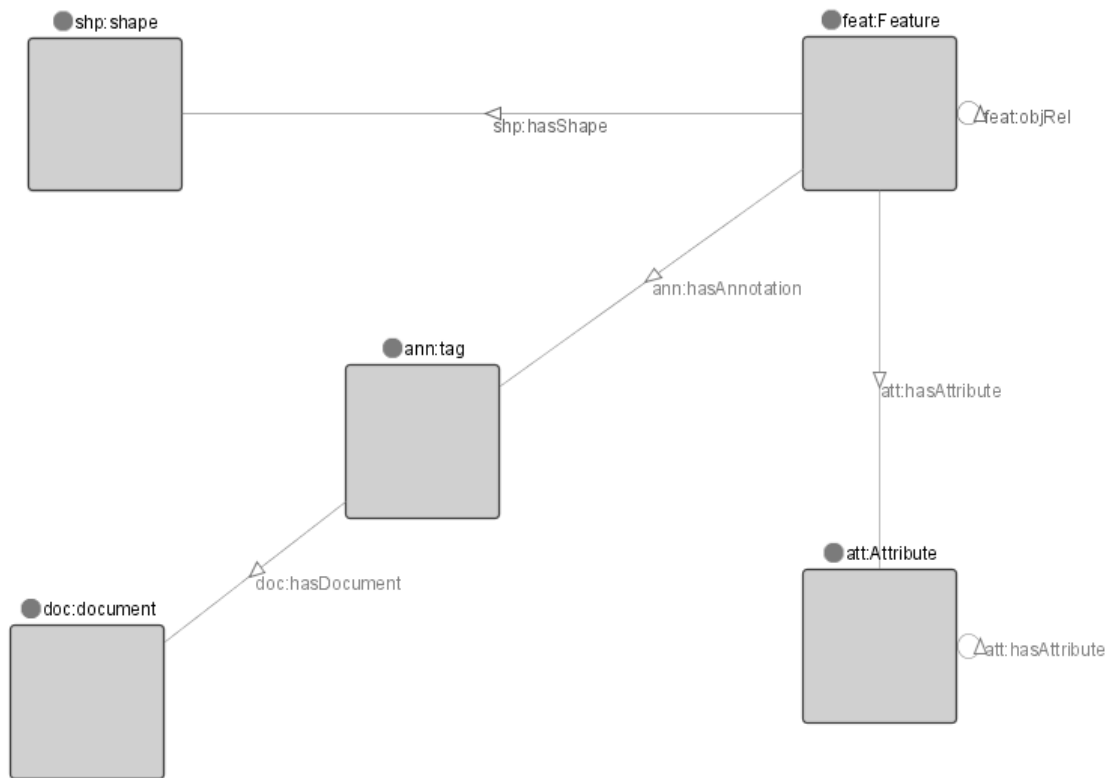
Figure 5.11: The basic ontological structure of the ArchaeoKM framework

This part presents the process of managing spatial signatures of the objects. The spatial knowledge through spatial operations and functions is presented in this section. This section mostly complements the previous two parts by discussing the spatial definition of features and spatial annotation process. As discussed in part 1, the semantic indexation begins with identifying spatial signature of the object. This part carries forward the discussion on how the spatial signatures are stored within the ArchaeoKM framework.

5.2.2.3.1 Classes

As seen in figure 5.4, three subclasses of `shp:_2D` present the natures of tagging the objects. They are first spatial signatures of the objects which will be utilized later to process spatial information. The process to semantically link these spatial signatures has already covered in the first part of this section. Summarizing it, the subclasses store the instances of the shapes in which they are tagged and link them to the objects in the class `feat:Feature`. This part discusses spatial annotation and how spatial signatures are processed to make them accessible for spatial knowledge management in terms of top level class axioms.

As with the other annotation processes, `ann:Tag` and `doc:document` class axioms present the specialized class for annotating the spatial data and documents. The class hierarchy of `doc:document` illustrated in figure 5.8 shows specialized class `doc:spatialDoc` and `doc:spDatabase` to provide instances representing the spatial components in the ontology.

The spatial signatures through tagging are processed to generate the Minimum Bounding Rectangles (MBR) which are then stored in database system. An instance is created to represent the details of the database. Those instances are the objects within `doc:spDatabase`. Likewise, the documents storing the geometries of the spatial signatures are represented through the instances present in `doc:spatialDoc`. These geometries can be set of point cloud or a GIS document. However, currently the ArchaeoKM framework uses point cloud to represent the geometries. In the same manner with other annotation process, the class `ann:tag` and its subclass `ann:spatialTag` provide the bridge between the objects stored in `feat:Feature` and the instances of these two classes.

5.2.2.3.2 Relationship

Abiding by the process of semantic annotation, the spatial annotation process also carries its activities through specialized object properties of `doc:hasDocument` and `ann:hasAnnotation`. Both object properties have specialized object properties designed to map the spatial data and documents to their semantics. The object property `doc:hasSpatial`, a specialized property of `doc:hasDocument` provides these mappings. The first set of semantics mapping to them is to attach the geometries to their relevant objects. It has already been mentioned that the geometries are stored in their proprietary files represented through the objects in class `doc:spatialDoc` and the MBRs are stored as spatial data within database system and represented through the objects in class `doc:spDatabase`. Mappings of these data and documents are carried out through the spatial annotation process. The complete spatial annotation activities are presented in next chapter. However, this part covers the principle behind the spatial annotation and why the process provides spatial basement for the later spatial activities.

In most cases a scan (point cloud scan obtained through laser scanning) constitutes the geometry of the complete site. This often contains very large set of point cloud. In order to identify and extract the features within the site, one has to identify the extent of that object in the point cloud and extract through some manual or semi-automatic process. The ArchaeoKM

framework provides a framework to carry out these processes which will be discussed in next chapter. The underlying principle uses the tagging geometries and extracts the MBRs and stores them in the database. It extracts the sets of point cloud from the large scan file, creates separate files for each set and stores them as ASCII files. Now instances are created in the ontology which has already been discussed to represent these MBRs and ASCII files. With the object properties `doc:hasSpatial` and `ann:hasSpatialAnno` these instances are mapped to the objects in `feat:Feature`. The process is almost the same with other annotation processes. However, the MBRs and its mapping to objects provide spatial linkages to the objects which can be exploited to carry out spatial operations and functions to enrich the knowledge base spatially.

5.2.2.3.3 Attribute

It could be realized from the last part that the spatial annotation process possesses two annotation interrelated sides. First the annotation to the MBRs present in the database and the next annotations to the file systems. They are related together through their connections to the objects on which they are annotated. Attributes through the data properties again play important role in providing the semantics to the annotations. The annotations to the database are carried out through assigning semantics to the annotations as assigning the relevant database and its relevant table in which the data is stored. It also provides the connection to the spatial column in which geometries of the objects are stored. An object property `doc:hasDBDetails` under general class `doc:hasDocumentDetails` provides these attributive connections. The three data properties to address the semantics of spatial annotation part of connecting to the MBRs are `doc:dbName`, `doc:spColumn` and `doc:tableName`.

5.3.   The Spatial Top Level Ontology

The realization of spatial signatures of the identified objects in the knowledge base has been discussed in the part of spatial annotation in the last chapter. The attachments of these spatial signatures provide a framework that could exploit the developments in spatial technology to provide the objects their spatial identity in respect to their surrounding objects. However, it is important to adjust the components within spatial technology in the top level ontology. This section covers the spatial top level ontology of the ArchaeoKM framework.

Although the impact of spatial integration is realized in the semantic level when the spatial components are integrated in the ontology, the usage of spatial features begins earlier than that. The spatial functionalities provided by database system form foundations to how they should be adjusted. A parallel structure facilitating the spatial components in different levels of the system architecture has already been presented in chapter 2 through figure 2.4. At the syntactic level where most of knowledge generation activities are carried out, spatial components are handled through spatially annotating the identified objects. This spatial annotation process draws a Minimum Bounding Rectangles (MBRs) around the objects and stores them as spatial data type in PostgreSQL database system. These MBRs would be used to carry out spatial rules while managing knowledge. It should be noted that the MBRs are not the optimal way of representing the objects and would constitutes some degrees of error during the analysis process. The ideal approach would be to use the boundaries of the objects for representation and analysis purpose. The algorithm to extract point cloud from the boundary is still in the domain of research and not completely matured and hence this thesis research uses MBRs to put forward the ideas.

It is the semantic level where the most of the integration work is carried out. The domain ontology is modified to represent the spatial functions and operations within it. It has been mentioned that the thesis work revolves around two categories of spatial operations and the integration process takes the functions and operations within these two categories which are the geospatial relationship functions and the geospatial processing functions. These functions are defined by the OGC consortium. The Open Geospatial Consortium, Inc (OGC) is an international industry consortium of 404 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OpenGIS® Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

The top level ontology should model spatial technology in terms of its spatial functions and operations. This modeling process should accommodate the spatial functions and operations and maintain their true identity. The following parts discuss the accommodation process with respect to two categories of functions discussed in chapter 3 that contains the maximum numbers of these functions.

5.3.1. Geospatial Processing Functions

These functions return geometries on their executions. It is hence important to have provision to store these returned geometries in the top level ontology. A generalized class `sa:spatialOperation` is introduced in the top level ontology. Every spatial operation under geospatial processing functions is then adjusted as its subclass. The class hierarchy of `sa:spatialOperation` reveals that the subclasses within it are the classes which need to represent returned geometries in some form and is illustrated in figure 5.12.
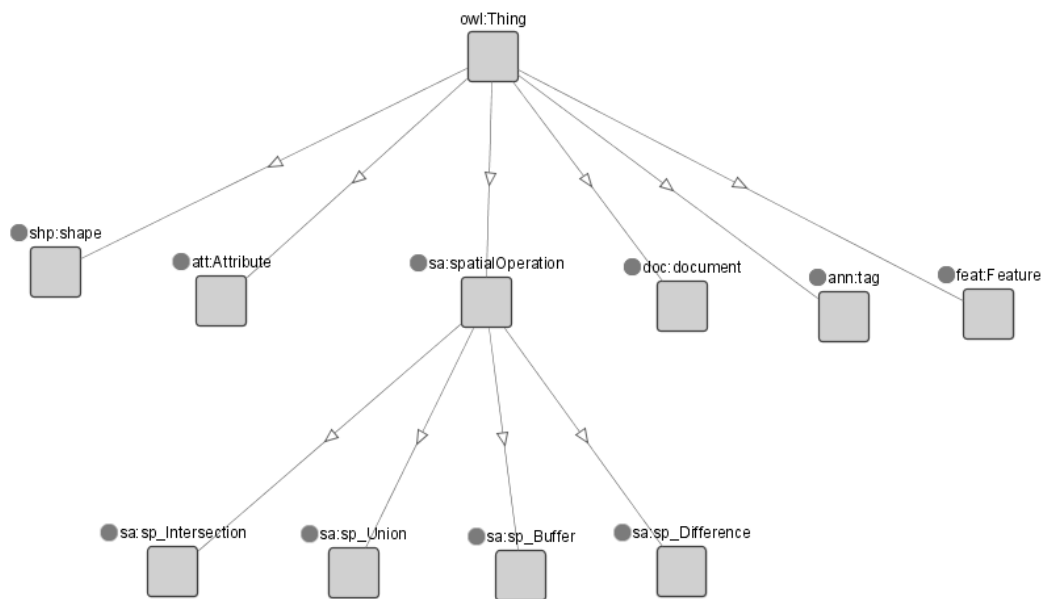
Figure 5.12: The class hierarchy of `sa:spatialOperation`

The four spatial processing functions which are discussed here are *Buffer*, *Union*, *Intersection* and *Difference*. These spatial functions compute new spatial geometries. These new geometries are also stored in the spatial database in order to be computed by future spatial functions. As a solution, we defined four new classes called `sa:sp_Buffer`, `sa:sp_Union`, `sa:sp_Intersection` and `sa:sp_Difference` which are specialized classes of `sa:spatialOperation`. The classes here are instantiated when the spatial operation of this category is executed. This instantiation process creates an individual for the class representing the kind of operation which just got executed. The result geometries of executions are stored within the data properties `feat:localPlacement` of these instantiated individuals.

The functions under this category need to take a feature to execute them. The feature are objects within class `feat:Feature`. In order to maintain a relationship between the spatial operations representing classes under `sa:spatialOperation` and features under `feat:Feature` in the ontology an object property `sa:hasSpatialRelations` is added in the top level ontology. The specialized property relates the individuals under `sa:spatialOperation` and `feat:Feature`. For example for every instance in class `sa:sp_Buffer` (sub class of `sa:spatialOperation`) there is a property `sa:hasBuffer` (specialized object property of `sa:hasSpatialRelations`) which relates the `sa:sp_Buffer` class to the classes specializing `feat:Feature`. There are also four `sa:hasSpatialRelations` defined corresponding to each geospatial processing functions (`sa:hasBuffer, sa:hasUnion, sa:hasIntersection, sa:hasDifference`). Besides these object properties, data properties which correspond to attributive nature of the relationships are also adjusted in the top level ontology. The importance of `feat:localPlacement` has already been covered. A generalized data property `sa:hasSpatialAttribute` is introduced in the top level ontology. Other attributive properties as `sa:hasBufferDistance` (denotes the buffer distance of the buffer) are specialized properties of it.

Table 5.1 demonstrates the four geospatial processing functions with their relevant object properties.

Functions	Concept	ObjectProperty	Execution Method
Buffer	sa:sp_Buffer	sa:hasBuffer(x,c)	$sa{:}sp\_Buffer$ $\sqsubseteq \exists sa{:}hasBuffer.feat{:}Feature$ $\sqcap sa{:}hasBufferDistance.\ c$ c is of float value providing the buffer distance
Union	sa:sp_Union	sa:hasUnion(x,c)	$sa{:}sp\_Union$ $\sqsubseteq \exists sa{:}hasUnion.feat{:}Feature$ $\sqcap\ \geq 2\ hasUnion$

Intersection	sa:sp_Intersection	sa:hasIntersection(x,c)	$sa\!:sp\_Intersection$ $\sqsubseteq \exists sa\!:hasIntersection.feat\!:Feature$ $\sqcap (\geq 2\ hasIntersection)$
Difference	sa:sp_Difference	sa:hasDifference(x,c)	$sa\!:sp\_Difference$ $\sqsubseteq \exists sa\!:hasDifference.feat\!:Feature$ $\sqcap (\geq 2\ hasDifference)$

Table 5.1: The Geospatial processing Functions

The variables x and y are the domains and ranges of the property. The variables x and y are individuals of specialized classes of `feat:Feature` or `sa:spatialOperation`.

5.3.2. Geospatial relationship functions

These functions demonstrate the spatial relations between objects hence they are very straightforward when adjusting in ontology. They can be directly adjusted through object properties within the top level ontology. These functions are adjusted as specialized object properties of `sa:hasSpatialRelations`. The execution pattern of every function in this category is executed in similar. The table 5.2 illustrates the steps of every spatial function following OGC spatial operation standards but this research thesis utilizes four operations to demonstrate the argument. Those functions are Disjoint, Touch, Within and Overlap which are represented through `sa:hasDisjoint,` `sa:hasTouch,` `sa:hasWithin` and `sa:hasOverlaps` subsequently.

Functions	ObjectProperties	Characteristics
Disjoint	sa:hasDisjoint(x,y)	Symmetric
Touches	sa:hasTouch(x,y)	Symmetric
Within	sa:hasWithin(x,y)	Transitive
Overlaps	sa:hasOverlaps(x,y)	No characteristics
Equals	sa:hasEqual(x,y)	Symmetric, Transitive

Crosses	sa:hasCrosses(x,y)	Symmetric
Intersects	sa:hasIntersect(x,y)	Symmetric
Contains	sa:hasContain(x,y)	Transitive

Table 5.2: The Georelationtionship Functions

The variables x and y are the domains and ranges of the property. The variables x and y are specialized classes of `feat:Feature` or `sa:spatialOperation`.
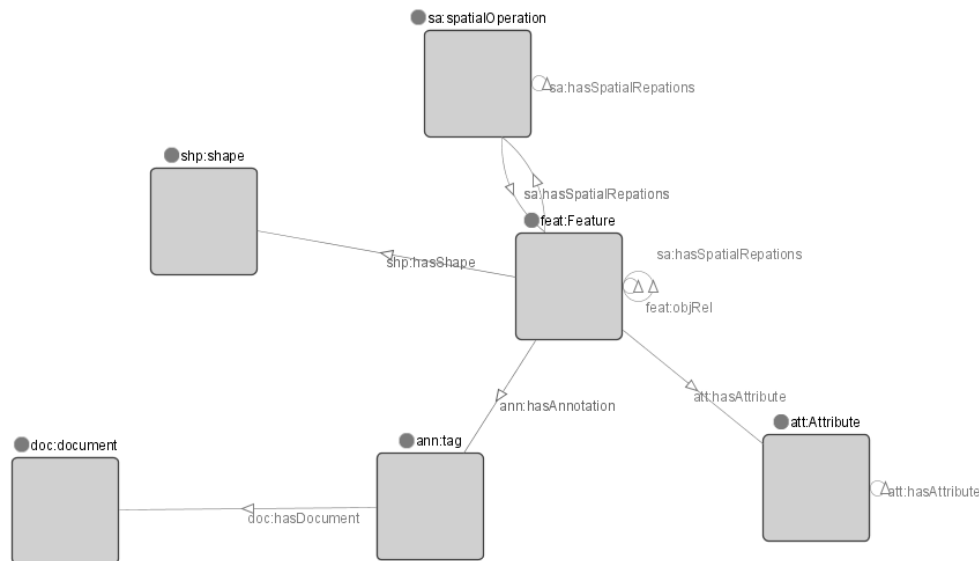

Figure 5.13: The modified Graphical representation of the ArchaeoKM framework

The graphical representation from figure 5.11 in the ArchaeoKM framework is thus modified to include the spatial components into it. This is illustrated in figure 5.13. A complete framework of spatial components into the top level ontology has been established. It is then left for the Translation engines to parse the spatial components and complete the spatial processing for the smooth operations of execution of knowledge technologies.

5.4.   Translation Engine

The translation engine is a part of the spatial facilitator that allows the computation of spatial SPARQL queries and spatial SWRL rules. In both cases, the translation engine interprets the statements in order to parse the spatial components. Once the spatial components are parsed, they are computed through relevant spatial functions and operations by the translation engine through the operations provided at the database level. The results are populated in the

knowledge base thus making it spatially rich. After that, the spatial statements are translated to standard statements for the executions through their respective engines. With the inference engine, the enrichment and the population of the ontology through the results of the inference process is stored in the ontology.
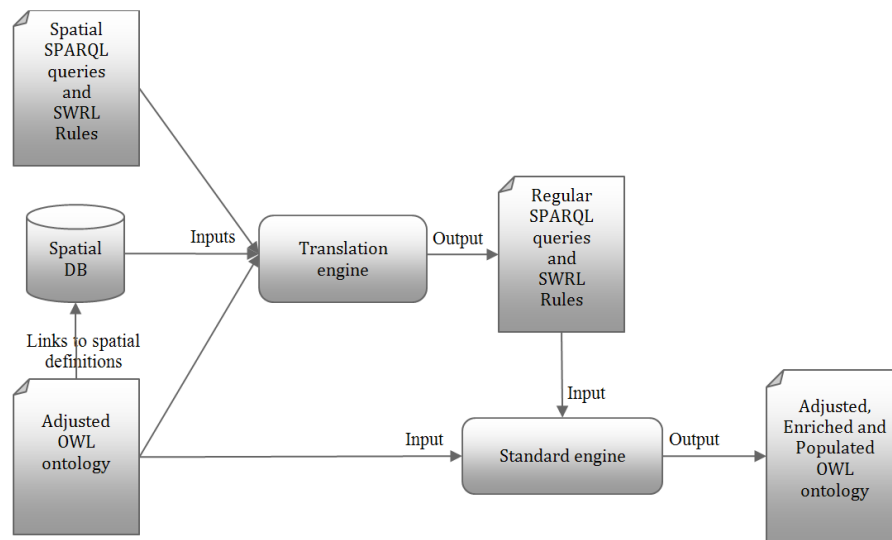
Figure 5.14: The spatial processing of the translation Engine

The next sections present in details the translation engine and more specifically the translation process of spatial SPARQL queries to regular queries. The following one presents the translation process of spatial SWRL rules to regular SWRL rules. These two processes have in common the use of SQL statements to query to the spatial database.

5.4.1. Spatial SPARQL Queries

The previous chapter introduced the notion of FILTER in SPARQL queries. FILTER can be used to compare strings and derive results. The functions like regular expression which matches plain literal with no language tag can be used to match the lexical forms of other literals by using string comparison function. In addition, SPARQL FILTER uses the relational operators as = or > or < for the comparison and restrict to the results that they return. From this idea, the FILTER principle is extended in order to process geospatial relationship functions.

5.4.1.1. Geospatial processing FILTER

Geospatial processing functions need to be addressed through enriching the knowledge base with the spatial operations which is related to them during the execution of the query. The enrichment process should be rolled back after the results are returned into its original form before the execution iff the SELECT statement is used under the filter. The optimization of the SPATIAL_FILTER is discussed later which highlights the management of the knowledge base during the execution of the SPARQL queries.

The following example demonstrates the syntax of geospatial processing filters in SPARQL. It could be seen that a new spatial filter through the keyword SPATIAL_FILTER is introduced which helps the translation engine during the parsing process. The SPARQL statement with spatial filters in the example returns names of all the buildings in class `feat:Building` which are intersecting with the buffer of 2000 meters of the rivers in class `feat:River` with their respective rivers names.

```
SELECT   ?name1 ?name2
WHERE
{
        ?feat1              feat:name    ?name1
        ?feat2              feat:name    ?name2
        ?feat1              rdfs:type    feat:River
        ?feat2              rdfs:type    feat:Building

        SPATIAL_FILTER [buffer (?x, 2000,?feat1)]
        SPATIAL_FILTER [intersection (?y,?x,?feat2)]
}
```

The term SPATIAL_FILTER is identified by the translation engine during the translation process. It then determines the nature of spatial operations and executes them. In this particular case, the geospatial processing functions of *buffer* and *intersection* are executed and the results are populated in the knowledge base first in classes of `sa:sp_buffer` and `sa:sp_intersection`. The class `sa:sp_buffer` has individuals with the buffer zones of 2000 meters around the rivers and the class `sp_intersection` has individuals of the intersecting buildings with this buffer. This intersecting buildings are then related to their corresponding rivers through the object property `sa:hasIntersection`. It thus completes the enrichment process. The engine then translates the spatial SPARQL statement to the standard SPARQL statement to execute the query. The translated SPARQL statement is illustrated below.

```
SELECT   ?name1 ?name2
WHERE
{
        ?feat1      feat:name               ?name1
        ?feat2      feat:name               ?name2
        ?feat1      rdfs:type               feat:River
        ?feat2      rdfs:type               feat:Building

        ?feat1      sa:hasBuffer            ?x
        ?x          rdfs:type               sa:sp_buffer
        ?x          sa:hasBufferDistance    2000

        ?y          rdfs:type sa:sp_Intersection
        ?y          sa:hasIntersection ?x
        ?y          sa:hasIntersection ?feat2

}
```

After the execution of the statement to return the results, the knowledge base rolls back to its former state. The syntax and translation procedures of these geospatial processing filters are given in table 5.3.

Function	Spatial SPARQL Syntax	Translation
Buffer	SPATIAL_FILTER [buffer (?x, b, ?y)]	?x rel:hasBuffer ?y
	Result: Populated in the knowledge base as	?y rdfs:type sa:sp_buffer
	individuals of class `sa:sp_Buffer`.	?y sa:hasBufferDistance distanceConstant
Union	SPATIAL_FILTER [union (?x, ?y1,?y2)]	?x rdfs:type sa:sp_Union
	Result: Populated in the knowledge base as	?x sa:hasUnion ?y1
	individuals of class `sa:sp_Union`.	?x sa:hasUnion ?y2
Intersection	SPATIAL_FILTER [intersection (?x, ?y1,?y2)]	?x rdfs:type sa:sp_Intersection
	Result: Populated in the knowledge base as	?x sa:hasIntersection ?y1
	individuals of class `sa:sp_Intersection`.	?x sa:hasIntersection ?y2
Difference	SPATIAL_FILTER [difference (?x, ?y1,?y2)]	?x rdfs:type sa:sp_difference
	Result: Populated in the knowledge base as	?x sa:hasDifference ?y1
	individuals of class `sa:sp_Difference`.	?x sa:hasDifference ?y2

Table 5.3: The spatial SPARQL syntax and its translation into SARQL syntax.

5.4.1.2. *Geospatial relationship FILTER*

In case of geospatial relationship filter it is straightforward as the enrichment process requires to enrich the object properties imitating spatial relationship between objects through the

results of the spatial operations at the database level. As with the previous case, the geospatial relationship filter uses the keyword SPATIAL_FILTER. This keyword parses the spatial components from the SPARQL statements. The following example illustrates the execution of SPARQL with these filters. The name of features couples are selected with this restriction. The first feature has to be a `feat:River` which is of kind of `feat:Feature`, and the second feature has to be a `feat:Building` which is also of kind of `feat:Feature`. The SPATIAL_FILTER selects the couples which are touching spatially.

```
SELECT   ?name1 ?name2
WHERE
{
            ?feat1              feat:name    ?name1
            ?feat2              feat:name    ?name2
            ?feat1              rdfs:type    feat:River
            ?feat2              rdfs:type    feat:Building

            SPATIAL_FILTER [touches (?feat1, ?feat2)]
}
```

This process is a selection process, and no inference process is engaged. The aim of the translate engine consists to compute the touches spatial process of the Cartesian production between the features of the kind `feat:River` and `feat:Building`. In the case of a positive result, this new link is populated in the knowledge base between the couple of feature with the help of a `sa:hasTouch` relationship which is of the kind of `sa:hasSpatialRelations`. Once the process is ended, the rule is translated to a standard given in the following example. It can be seen that the SPATIAL_FILTER is replace by the triple "feat1 `sa:hasTouch` ?feat2". Thus this rule can be processed by a standard SPARQL engine.

```
SELECT   ?name1 ?name2
WHERE {
            ?feat1              feat:name    ?name1
            ?feat2              feat:name    ?name2
            ?feat1              rdfs:type    feat:River
            ?feat2              rdfs:type    feat:Building

            ?feat1              sa:hasTouch ?feat2
    }
```

The knowledge base once again rolls back to its original form after the execution of the query. The table 5.4 shows the translation of geospatial relationship functions contained in SPATIAL_FILTER into standard triple component of a SPARQL query.

Functions	Spatial SPARQL Syntax	Translation
Disjoint	SPATIAL_FILTER [Disjoint(?x, ?y)]	?x sa:hasDisjoints ?y
Touches	SPATIAL_FILTER [Touches(?x, ?y)]	?x sa:hasTouch ?y
Within	SPATIAL_FILTER [Within(?x, ?y)]	?x sa:hasWithin ?y
Overlaps	SPATIAL_FILTER [Overlaps(?x, ?y)]	?x sa:hasOverlaps ?y
Equals	SPATIAL_FILTER [Equals(?x, ?y)]	?x sa:hasEqual ?y
Crosses	SPATIAL_FILTER [Crosses(?x, ?y)]	?x sa:hasCrosses ?y
Intersects	SPATIAL_FILTER [Intersects(?x, ?y)]	?x sa:hasIntersects ?y
Contains	SPATIAL_FILTER [Contains(?x, ?y)]	?x sa:hasContains ?y

Table 5.4: The spatial SPARQL syntax and its translation into SARQL syntax.

5.4.1.3. *Optimization*

The translation engine is time consuming for large spatial database. In order to select the context of execution four options can be given to the SPATIAL_FILTER.

- SPATIAL_FILTER_SELECT: No spatial operation is undertaken; the rule is translated without any spatial processing
- SPATIAL_FILTER_PROCESS: Spatial operations are processed only for the couples of features which don't have this relationship. If this relation already exists, this one is not computed.
- SPATIAL_FILTER_UPDATE: Spatial operations are processed only for the couples of features which have already this relationship in order to update these relationships.
- SPATIAL_FILTER_ALL: This is the option by default which consists to compute all relationship for the Cartesian product in order to process it if it doesn't exist or in order or update it.

The following example shows that the selection of features which have the touches relationship is done with the option SPATIAL_FILTER_UPDATE.

```
SELECT  ?name1 ?name2
WHERE
{
        ?feat1              feat:name   ?name1
        ?feat2              feat:name   ?name2
        ?feat1              rdfs:type   feat:River
        ?feat2              rdfs:type   feat:Building

        SPATIAL_FILTER [touches (?feat1, ?feat2)]
        SPATIAL_FILTER_UPDATE
}
```

In addition the spatial filter can be combined by the following manner. It consists to insert news filters and to use the same variable. The following example consists to select building which contains a chimney in order to see if it touches a river. Moreover, no spatial processing is done, only the existing knowledge in the ontology is used to process this query.

```
SELECT  ?name1 ?name2
WHERE {
        ?feat1              feat:name   ?name1
        ?feat2              feat:name   ?name2
        ?feat1              rdfs:type   feat:River
        ?feat2              rdfs:type   feat:Building
        ?feat2              rdfs:type   feat:Chimney

        SPATIAL_FILTER [touches (?feat1, ?feat2)]
        SPATIAL_FILTER [touches (?feat2, ?feat3)]
        SPATIAL_FILTER_SELECT
}
```

5.4.2. Inference Rules through SWRL

In an attempt to define the built-ins for SWRL, a list of eight built-ins was proposed during the research work. These eight built-ins reflect four geospatial processing functions and four geospatial relationship functions that are discussed previously. The built-ins reflecting geospatial processing functions are built up in combinations with the spatial classes adjusted in the ontology and their relevant object properties. The built-ins for geospatial relationship functions are object properties and using these object properties in collaboration to the spatial functions in database system.

5.4.2.1. Geospatial processing Built-ins

The first set of built-ins is the built-ins for geospatial processing functions. They are functions returning geometries and adjusted in the ontology through `feat:Feature` `sa:hasSpatialRelations sa:spatialOperation` sequence. This class-property series is

illustrated in table 5.5. The initial step consists of the spatial built-ins being parsed and processed by the translation engines. First the spatial built-ins are identified from the statement and parsed. Concurrently, the features on which these built-ins are applied are also identified. After that, the SQL statements with relevant spatial function on the relevant objects of the features are executed at the database level. The results are then populated in the knowledge base. Once, the knowledge base is populated, the spatial built-ins are broken down into standard `feat:Feature sa:hasSpatialRelations sa:spatialOperation` sequence to generate the standard SWRL statement which is executed through standard inference engines.

Functions	Class	Object Property	Data Property	Built-ins
Buffer	sa:sp_Buffer	sa:hasBuffer	sa:hasBufferDistance	Buffer(?x, b, ?y)
Union	sa:sp_Union	sa:hasUnion	-	Union(?x,?y$_1$,y$_2$)
Intersection	sa:sp_Intersection	sa:hasIntersection	-	Intersection(?x,?y$_1$,y$_2$)
Difference	sa:sp_Difference	sa:hasDifference	-	Difference(?x,?y$_1$,y$_2$)

Table 5.5: The Geospatial processing built-ins

In the table 5.5, ?x, ?y, ?y1, ?y2 are the variables and b is the constant.

The execution of every built-in can be elaborated through first running down the spatial operation and then translating the statements with spatial built-in into standard SWRL statements. Simplifying the explanations with an example of

```
feat:Feature(?x) ^ Buffer(?x, b, ?y).
```

The example highlights the usage of built-in *Buffer* on objects within the specialized classes of `feat:Feature` with the buffer distance. This statement is elaborated first through running the SQL statement with the spatial function buffer on each objects of the class to which it meant to run. That is if the statement is related to buffering walls, then each instance of class `feat:Wall` is taken and buffered through the execution of the SQL statement. The SQL statement with spatial function Buffer would look like:

```
SELECT Buffer(geom::Feature, bufferDistance)
```

Here, the geom are the geometries of the objects within specialized classes of `feat:Feature`. The result of this execution is then populated in the knowledge base. Primarily, the rows in

result are geometries which indicate the buffers of each object with certain buffer distance. The class sa:sp_Buffer is instantiated with objects representing every row and storing the buffer geometry and the buffer distance within them. Then after, it is time to translate the statement with the spatial built-in into standard form of SWRL statement which would be

```
feat:Feature(?x) ^ sa:hasBuffer(?x,?y) ^ sa:sp_Buffer(?y) ^
sa:hasBufferDistance(?y,b).
```

The statement thus converts the spatial built-in into `feat:Feature` `sa:hasSpatialRelations` `sa:spatialOperation` sequence of a standard SWRL statement. The complete list of spatial built-ins with their respective executions through SQL statements and breakdown in the translation engines is illustrated in table 5.6.

Built-ins	SQL Statements	Translated Built-ins
swrlbspatial:Buffer(?x, b, ?y)	SELECT Buffer(geom::Feature, bufferDistance) Result: Populated in the knowledge base as individuals of class `sa:sp_Buffer`.	feat:Feature(?x) ^ sa:hasBuffer(?x,?y) ^ sa:sp_Buffer(?y) ^ sa:hasBufferDistance(?y, b)
swrlbspatial:Union(?x,?y_1,?y_2)	Select Union(geom::Feature1, geom::Feature2) Result: Populated in the knowledge base as individuals of class `sa:sp_Union`.	sa:sp_Union (?x) ^ sa:hasUnion(?x, ?y_1) ^ sa:hasUnion(?x, ?y_2) ^ feat:Feature(?y_1) ^ feat:Feature(?y2)
swrlbspatial:Intersection(?x,?y_1,?y_2)	Select Intersection(geom::Feature1, geom::Feature2) Result: Populated in the knowledge base as individuals of class `sa:sp_Intersection`.	sa:sp_Intersection(?x) ^ sa:hasIntersection(?x, ?y_1) ^ sa:hasIntersection(?x, ?y_2) ^ feat:Feature(?y_1) ^ feat:Feature(?y2)
swrlbspatial:Difference(?x,?y_1,?y_2)	Select Difference(geom::Feature1, geom::Feature2) Result: Populated in the knowledge base as individuals of class `sa:sp_Difference`.	sa:sp_Difference(?x) ^ sa:hasDifference(?x, ?y_1) ^ sa:hasDifference(?x,?y_2) ^ feat:Feature(?y_1) ^ feat:Feature(?y2)

Table 5.6: The SQL statements executions of geospatial processing built-ins

5.4.2.2. *Geospatial relationship Built-ins*

Contrary to geospatial processing built-ins, geospatial relationship built-ins rely on the object properties and more straight forward. The built-ins and their linkage to the object properties are presented in table 5.7.

Functions	Class	Object Property	Built-ins
Disjoint	-	sa:hasDisjoint	Disjoint(?x, ?y)
Touches	-	sa:hasTouch	Touches(?x, ?y)
Within	-	sa:hasWithin	Within(?x, ?y)
Overlaps	-	sa:hasOverlap	Overlaps(?x, ?y)

Table 5.7: The Geospatial relationship Built-ins

However, it is necessary to determine the nature of built-ins from the statement to determine what spatial operation needs to be performed at database level. These statements are hence parsed to identify the spatial built-ins from the statement. After that, the SQL statement with related spatial operation is executed in the database level. The results are populated against their specified object properties in the knowledge base. Now, the statements are ready to get executed. The spatial built-ins are broken down into `feat:Feature` `sa:hasSpatialRelations feat:Feature` sequence by the translation engine which is now a standard SWRL statement so can be executed with existing Inference engines.

It would be helpful to elaborate with an example of built-in

```
Feat:Feature(?x) ^ feat:Feature(?y)^ Touches(?x,?y).
```

It is a spatial operation to determine whether an object is touching another. Generally, the geospatial relationship operations are binary operations and return Boolean values when are executed alone. However, when executed as a conditional parameter of the SQL statement, they yield results. That is if the statement

```
SELECT Touches(geom::Feature1, geom::Feature2)
```

is executed. It returns either *true* or *false* determining whether the geometry of feature1 touches geometry of feature2. But if the same operation is executed as

```
SELECT    Feature2    FROM    spTable    WHERE    Touches(geom::Feature1,
geom::Feature2),
```

then it returns all the feature2 which touches feature1. Here `spTable` is the table where the geometries of the features are stored in the database system and has been spatially annotated. The results derived through the execution of the statement with Touch operation is then populated against `sa:hasTouch` object property of the specified feature. The last step is to break down the `Touches(?x, ?y)` built-in into `feat:Feature sa:hasSpatialRelations feat:Feature` sequence to get the SWRL statement executed. The breakdown of the spatial built-in `Touches(?x, ?y)` is given as

```
feat:Feature(?x) ^ hasTouch(?x, ?y) ^ feat:Feature(?y).
```

It is a standard SWRL statement which can again be inferred by inference engines. The complete list of SQL statement execution of spatial built-ins is illustrated in table 5.8.

Built-ins	SQL Statements	Translated Built-ins
swrlbspatial:Disjoint(?x, ?y)	SELECT Feature2 FROM spTable WHERE Disjoint(geom::Feature1, geom::Feature2)	sa:hasDisjoint(?x, ?y)
swrlbspatial:Touches(?x, ?y)	SELECT Feature2 FROM spTable WHERE Touch(geom::Feature1, geom::Feature2)	sa:hasTouch(?x, ?y)
swrlbspatial:Within(?x, ?y)	SELECT Feature2 FROM spTable WHERE Within(geom::Feature1, geom::Feature2)	sa:hasWithin(?x, ?y)
swrlbspatial:Overlaps(?x, ?y)	SELECT Feature2 FROM spTable WHERE Overlap(geom::Feature1, geom::Feature2)	sa:hasOverlaps(?x, ?y)

Table 5.8: The SQL statements executions of geospatial relationship built-ins

5.4.2.3. *Example of a Spatial Inference*

The domain of archaeology benefits from this work and could surely be of benefit for lot of other domains. To show this we present a simple example to determine the location of possible flooding zone when the river bank bursts with excessive water during rainy season. This is a very common exercise for a flood management system in hydrology and it gives interesting clues for archaeology. In general with a common GIS, a set of activities are carried out which are mentioned in the following sequences:

- *Buffer the river by certain distance (e.g. 100 meters)*
- *Determine the elevation of land parcel inside the buffer zone*

- *Check whether the land parcel elevation is above the threshold (e.g. 25 meters)*
- *Select areas below the threshold area and determine them as flood liable zone.*

It should be understood that this example is provided just as a proof of the concept. Hence details on other hydrological factors are ignored on purpose. For a simple location analysis as such requires at least four steps of spatial analyses. This paper provides an alternative through the spatial extension of SWRL in one step. We combine the existing built-ins in existing SWRL and the spatial built-in mentioned in this paper to execute this analysis.

```
River(?x) ^ LandParcel(?y) ^ hasElevation(?y, ?Elv) ^
swrlb:lessThan(?Elv, 25) ^ spatialswrlb:Buffer(?x, 50, ?z) ^
spatialswrlb:Intersection(?z, ?y, ?res)  →
FloodingLandParcel(?y)
```

This statement execution rule provides an alternative to the above mentioned steps for spatial analysis carried out in current GIS. The spatial extensions of existing database systems are used to perform those spatial operations while executing the rule.

```
River(?x) ^ LandParcel(?y) ^ hasElevation(?y, ?Elv) ^
swrlb:lessThan(?Elv, 25) ^ sa:hasBuffer(?x, ?z) ^
feat:sp_Buffer(?z) ^ sa:bufDistance(?z, 50) ^
feat:Intersection(?res) ^ sa:hasIntersection(?res, ?y) ^
sa:hasIntersection(?res, ?z) → FloodingLandParcel(?y)
```

The result of this rule is that the individuals which respect the rule and belong to LandParcel, belong also to the concept `FloodingLandParcel`. In addition, if the concepts `SquareLandParcel` and `RectangleLandParcel` are specialized concepts of the concept `LandParcel`, consequently the individuals of these concepts which respect the rule belong also to the concept `FloodingLandParcel`.

5.5 Spatial Adjustment of existing Ontologies by example

This thesis work presents the possibility of spatial integration within knowledge technologies present in semantic web framework that could benefit the archaeologists working in the industrial archaeology domain in their data management process. However, the benefit of the results of this thesis work moves beyond the industrial archaeological domain. The results could be used almost in every application that has some connections to the spatial contexts. In an attempt to demonstrate its view, this thesis uses an example of the famous wine ontology.

This ontology is used to demonstrate the principle of spatial ontology adjustment which allows the computation of spatial data on any ontology whether it is under development or currently existing. The wine ontology is selected for several reasons. The wine ontology appears frequently in the literature as an example to define tutorials. In addition, this ontology is far from the industrial Archaeology which underlines the universal principle of the ontology adjustment for spatial processing.

5.5.1. The Existing Ontology Adjustment

Existing ontology can benefit from incorporating the spatial components proposed by the research. The spatial functionalities could be exploited with two essential steps. First, the top level ontology has to be integrated into the existing ontology. In this manner, all the components of the spatial layer are available for the existing ontology, which are based on the annotating and tagging principles of documents but more specifically the spatial definitions. The second step consists of specializing concepts of the existing ontology which could possibly have spatial signatures. The research puts forward the concept of implementation through the wine ontology. In the wine ontology, wine regions can be defined as spatial region or polygons in a GIS system. With the help of existing tools the regions could be digitized for their spatial signatures. Likewise, the wineries can be geo-localized as points. The existing ontology once adjusted through the integration process, the spatial signatures stored in the database system could be mapped to the individuals of the concepts respectively of wine region and wineries which are already defined in the ontology. For instance, it should be possible to define individual `vin:ClosDeVougeot` as a French winery situated in Burgundy region and is geo-localized through the coordinates 47.174835, 4.95544 in the WGS84 coordinate system.

The figure 5.15 illustrates the adjusted wine ontology in ontology editor Protégé (Research, 2010). On the left side, the tree viewer represents the hierarchy of concept with the ArchaeoKM top level ontology and `feat:Feature` concept with the wine ontology specialized concept `vin:Region` and `vin:Winery`. All the other concept of the wine ontology can by spatially defined. On the right side, the list of the `vin:Winery` individuals is given. The individual `vin:ClosDeVougeot` appears in this list. This list is composed of 43 individuals and the list of `vin:Region` is composed of 36 individuals.
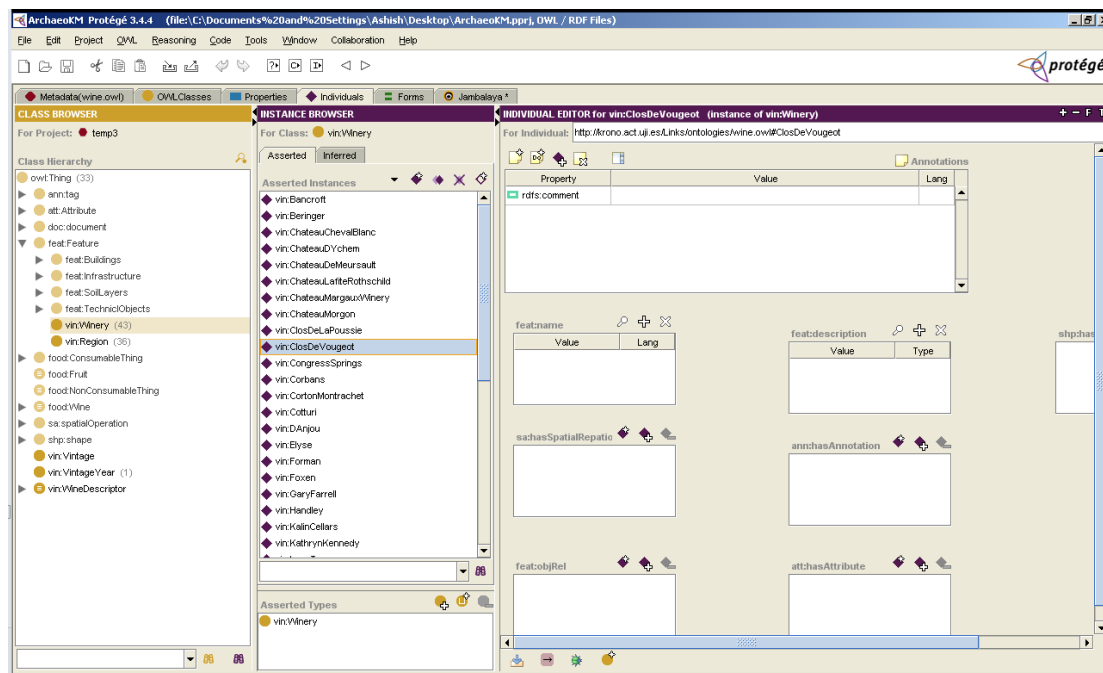
Figure 5.15: A snapshot of the wine ontology adjusted with spatial component.

5.5.2. Spatial Querying process

This section presents the benefit of spatial querying on spatial data composed of semantic definition. In the figure 5.16, the individual `vin:CoteDOrRegion` has a relationship `vin:adjacentRegion`. This relationship defines a symmetric relationship between two regions. In the wine ontology, this information is not feed. Currently, it is not possible to select adjacent regions and regions which are near to each other for instance as neighbors or 200km apart. The inclusion of spatial components through the top level ontology makes it possible. They are possible first through implementing the SPATIAL_FILTER with spatial queries in SPARQL. Two approaches can be suggested to map `vin:adjacentRegion` to the individuals of `vin:Regions` (It should be noted that the individuals of `vin:Regions` are spatially enriched). The first approach queries all the adjacent regions to `vin:CoteDOrRegion` with the touch operation. This could be perceived through the example below:

```
SELECT   ?adjacent
WHERE
{
      vin:CoteDOrRegion rdfs:type    vin:Region
      ?adjacent          rdfs:type    vin:Region

      SPATIAL_FILTER [touches (vin:CoteDOrRegion,?adjacent)]}
```

The second approach queries all the regions which lie inside 200km around region `vin:CoteDOrRegion` through buffer operation. It could be perceived through the example:

```
SELECT  ?region
WHERE
{
     vin:CoteDOrRegion rdfs:type    vin:Region
     ?region           rdfs:type    vin:Region

     SPATIAL_FILTER [buffer (?buffer,200000,vin:CoteDOrRegion)]
     SPATIAL_FILTER [intersection (?res,?buffer,?region)]
}
```

These examples implement SPARQL queries to the individuals of same class, the same can be implemented on individuals of different classes. Currently, no spatial relation is defined between regions and wineries. With the adjustment in the ontology and the formulating spatial definition of wine regions and wineries, query can be implemented - return all the wineries in a specific region.

```
SELECT  ?winery
WHERE
{
     vin:CoteDOrRegion rdfs:type    vin:Region
     ?winery           rdfs:type    vin:Winery

     SPATIAL_FILTER [within (vin:CoteDOrRegion,?winery)]
}
```
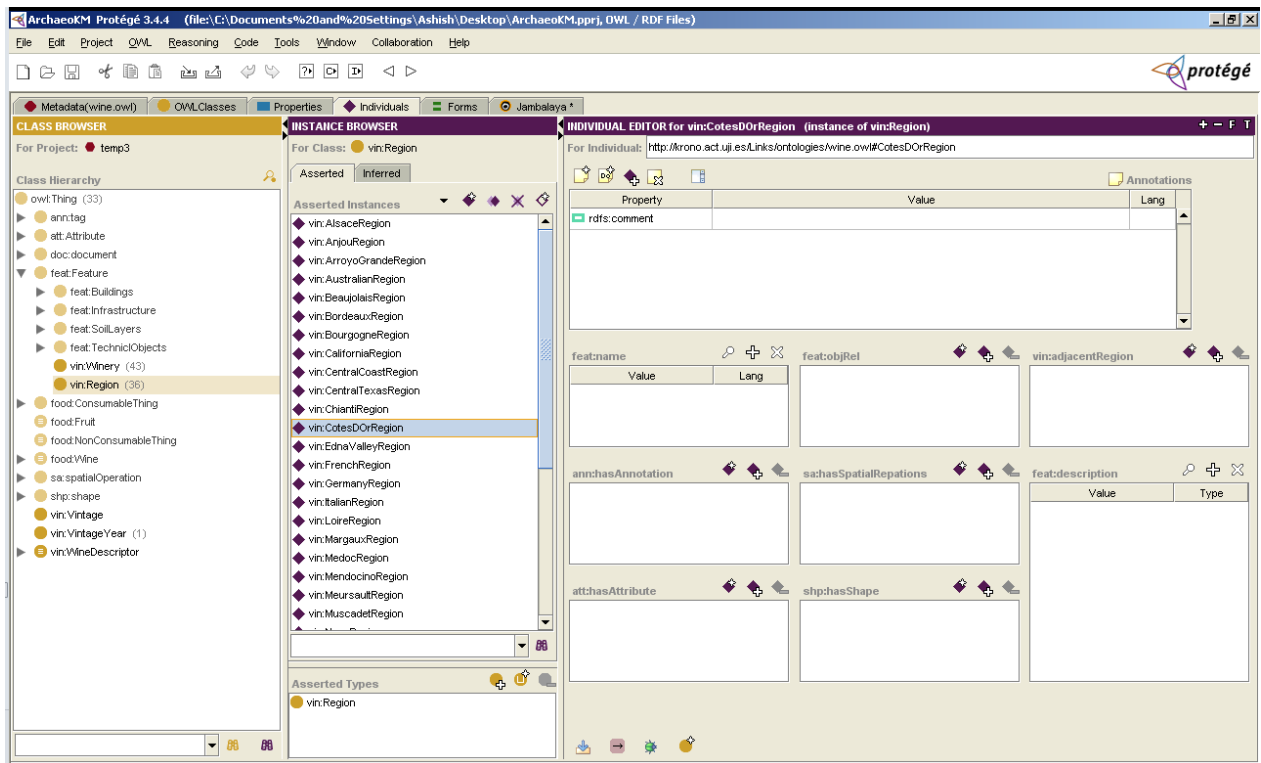
Figure 5.16: A snapshot of the wine ontology adjusted and the individual `vin:CotesDOrRegion`.

It is even helpful to check the validity and consistency of the relationships if those relationships are already defined and used to map in the knowledge base. Suppose the knowledge base consists the individual `vin:ClosDeVougeot` as a winery located in `vin:CoteDOrRegion`. The characteristic of the object property `vin:located` between the concept `vin:Region` and the concept `vin:Winery` requires the individual `vin:ClosDeVougeot` be related to the individual `vin:CoteDOrRegion` with through the object property. The following query is able to validate this relationship spatially.

```
SELECT  *
WHERE
{
      vin:CoteDOrRegion rdfs:type    vin:Region
      vin:ClosDeVougeot rdfs:type    vin:Winery

      SPATIAL_FILTER [within (vin:CoteDOrRegion, vin:ClosDeVougeot)]
}
```

In case it returns false, but still the spatial data is defined and correct, the ontology is inconsistent. The overlap between the semantic links and the spatial data permits to check the consistency of the knowledge base in the case that the links were not generated from the spatial processing.

5.5.3. Spatial Inference process

With the help of the SWRL rules, the spatial enrichment in the ontology is possible. The following simple example underlines this idea. The winery Clos de Vougeot `vin:ClosDeVougeot` is located in the region of Côte D'Or `vin:CoteDOrRegion`, and subsequently a region located in France `vin:FrenchRegion`. Consequently, the winery Clos de Vougeot `vin:ClosDeVougeot` is located in France `vin:FrenchRegion`. The transitive nature of object property `vin:hasSubRegion` allows the definition of relationships between regions `vin:Region`.

The following spatial SWRL rule enriches the ontology with vin:hasSubRegion relations between regions.

```
vin:Region(?x) ^ vin:Region(?y) ^ spatialswrlb:Within(?y, ?x)

  → vin:hasSubRegion(?x, ?y)
```

Likewise the following spatial SWRL rule enriches the ontology with vin:isLocatedInRegion relations between wineries and regions.

```
vin:Region(?x) ^ vin:Region(?y) ^ vin:Winery(?z) ^

vin:hasSubRegion(?x, ?y) ^  vin:isLocatedInRegion (?z, ?x)

 → vin:isLocatedInRegion (?z, ?y)
```

Lastly the following spatial SWRL rule combines both the rules to enrich the knowledge base through the relationships `vin:isLocatedInRegion` and `vin:hasSubRegion`

```
vin:Region(?x) ^ vin:Region(?y) ^ vin:Winery(?z) ^

swrlbspatial:Within(?y, ?x) ^ swrlbspatial:Within(?z, ?y)

  → vin:isLocatedInRegion (?z, ?y) ^ vin:hasSubRegion(?x, ?y)
```

The execution of this last rule results in creation of new relationships `vin:isLocatedInRegion` and `vin:hasSubRegion` which map the respective individuals. Consequently, the ontology is enriched with these new relationships.

5.6.   Discussion

The discussions that were presented in last chapters focus mainly on the implication of spatial technology within the Semantic Web. The Semantic Web technologies are relatively new technologies and have not matured enough to address every issue and specifically the spatial requirement. Consequently, this work raises the issue of the spatial technology integration within the Semantic Web technology. Primarily designed to address the problem of heterogeneity of information in Web in terms of their structure and semantics, the Semantic Web technologies were proposed to use collaborative approach of human-machine interaction to determine the interoperability through capturing the underlying semantics behind. This requires machine to learn the semantics behind the information and to human to understand them. Though the knowledge technology existed for some time, it flourished with the advancement in the Semantic Web technologies. The Semantic Web utilizes the tools and technologies based on the components of knowledge technology to allow machine to learn and interact with human on various issues. This has given opportunities like geospatial technology which is not primarily based on semantics to take the advantage of incorporating knowledge in its analysis. Secondly, the non-conventional data type like the temporal data needs to be addressed within the Semantic Web framework in order to prove its effectiveness as an alternate solution for the existing technologies. Such data types have been ignored in the researches. This can be clearly derived through the lack of these data types being mentioned in the Semantic Web stack. This chapter attempts to address the lack of geospatial inclusion in the Semantic Web technologies through discussing the conceptualizations and their implementations of the integration of spatial technology.

This chapter discusses the top level ontology including its spatial components with backdrop of the ArchaeoKM application. The ArchaeoKM framework is primarily designed to facilitate archaeologists to manage their information of the excavated objects through the knowledge possessed by them. It is a Web platform where archaeologists can work collaboratively to acquire, manage and visualize knowledge. Spatial aspects of the excavated objects need to be addressed for efficiency of the tool and hence the integration of spatial components in the application was important. The ArchaeoKM framework being based on Semantic Web technologies has ontology in its core. The top level ontology provides the base for functionalities of the application. The ArchaeoKM framework is based on 4Ks:

Knowledge Acquisition, Knowledge Management, Knowledge Visualization and Knowledge Analysis. These 4Ks can be semantic knowledge or the spatial knowledge. The top level ontology in the application is designed to support the 4Ks. The ontology first supports the semantic knowledge possessed by the archaeologists through its functionalities.

In addition, the Semantic Web technologies are not yet matured enough to carry out spatial operations independently and hence requires supports of spatial functionalities from database systems in order to carry out these operations. Thus for a required domain, it is necessary to adjust the existing ontology with the top level ontology including the spatial components in order to simulate the spatial operations in the database system. The operations are classified according to their nature of executions and results. This research though classifies them in four major groups, focuses on two groups containing maximum number of these operations to argue its concepts. The first group of operation returns geometries which need to be taken into account. The next one is the spatial relationships between the features. Both groups need to be adjusted accordingly. The top level ontology addresses the first group through accommodating classes to represent the operations while the next group is addressed through the object properties.

Spatial knowledge processing follows the 4Ks principle as well. Spatial knowledge acquisition is carried out first through the tagging process and then through the spatial annotation which provide spatial signatures for spatial operations. The classes and properties in the top level ontology are adjusted accordingly. Then the inference capabilities of knowledge tools within the Semantic Web technologies are taken advantage of to run the spatial rules and spatial queries. The research work came out with spatial filters and built-ins for querying and inferring spatial knowledge through the query language of SPARQL and the rule language of SWRL.

The spatial adjustment presented in the last section is one of the main principles of this work. Actually, by adjusting the existing ontology which consists to integrate the top level ontology and its spatial components, the spatial queries and rules are executed on this existing ontology if the spatial signatures of the individuals are fed in a defined spatial database. The wine ontology is used as an example of the possibilities given by this principle. That shows the generality of the solution which allows its application on all kind of domains which handle spatial elements.

Finally, the Semantic Web technologies are maturing as time goes by and it is inevitable to integrate spatial components within it. The dependency on database system might be cut down through the technical advancements in spatial algorithms developments within database systems. The possibilities and the effectiveness of the spatial integration could already be realized through the results of this work (though it is currently dependent on the database system). Next chapter will cover the results of this conceptualization. It will cover the ArchaeoKM platform and its functionalities. The spatial integration and the results of spatial knowledge manipulation are discussed independently through a customized solution for spatial knowledge processing.

References

Herring, J. R. (2010). *OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.* Open Geospatial Consortium Inc.

Research, S. C. (2010). *User Documentation*. Retrieved November 15, 2010, from Protege: http://protege.stanford.edu/doc/users.html

Chapter **6**

THE ARCHAEOKM WEB PLATFORM

This chapter presents the application prototype to demonstrate the applicability of the principles that were presented in the previous chapter. The application prototype is given the name ArchaeoKM to reflect the nature of the application. The name ArchaeoKM is the abbreviation of "**Archaeo**logical **K**nowledge **M**anagement" which facilitates the archaeologists to manage their information through their knowledge. Additionally, this chapter covers the technical frameworks on which the ArchaeoKM platform is developed.

Last chapters have provided enough backgrounds behind the route map of this research thesis. They have also provided the arguments on the implementation of archaeological knowledge in managing the dataset collected during the excavation process. The benefit of collaborative approach of archaeologists through the ArchaeoKM platform is characterized by the two following points. The knowledge sharing through the ArchaeoKM platform saves not only the redundancy in the activities it also provides to the archaeologists a better understanding of the objects. The underlying knowledge base operates as the knowledge hub where archaeologists can extract and attach new knowledge generated through managing it. In addition, the integration process of spatial component into the application has opened a new scope within the broader domain of the Semantic Web technologies. This chapter demonstrates the applicability of the integration of spatial components within the platform ArchaeoKM through a tailored made tool for spatial knowledge processing. It covers the current state of the application and the functionalities that are provided to facilitate the work of archaeologists in order to manage their findings.

This chapter lays its foundation on definition of the requirement in general and how ArchaeoKM fulfills the need. The basic requirement behind developing the application is to provide a platform to the archaeologists to interact and use their knowledge to manage the huge dataset. This requires the interfaces to be easy to interpret and use. In other hand the backend processing should be sophisticated enough to process complex knowledge archaeologists possess. It should not only handle the data interpretation of the information collected on field but also should manipulate them long after they are collected. It is hence necessary to use the machine to learn and interpret information and assist human to manage them through their knowledge. ArchaeoKM is developed with this intension. Carrying forward the legacy of knowledge processing through semantic interpretation, ArchaeoKM initiates the collaboration of spatial data within semantic framework to evoke the complete knowledge processing of the archaeological data.

This chapter is divided into 4 parts. The first part presents the ArchaeoKM Web platform as a tool for the industrial archaeology domain and its use of the 4Ks (Knowledge Acquisition, Knowledge Management, Knowledge Visualization, and Knowledge Analysis). The second part presents the spatial tool as a proof of concept. The third parts present the technologies

used to develop the ArchaeoKM Web platform. The last section is a discussion about the chapter.

6.1. The ArchaeoKM Web platform

The project user interface of the ArchaeoKM Web platform is shown in figure 6.1. It also provides the basic impression of the application through various sections of functionalities incorporated within. As the case study involves German archaeologists, the primary language used within the application is German.
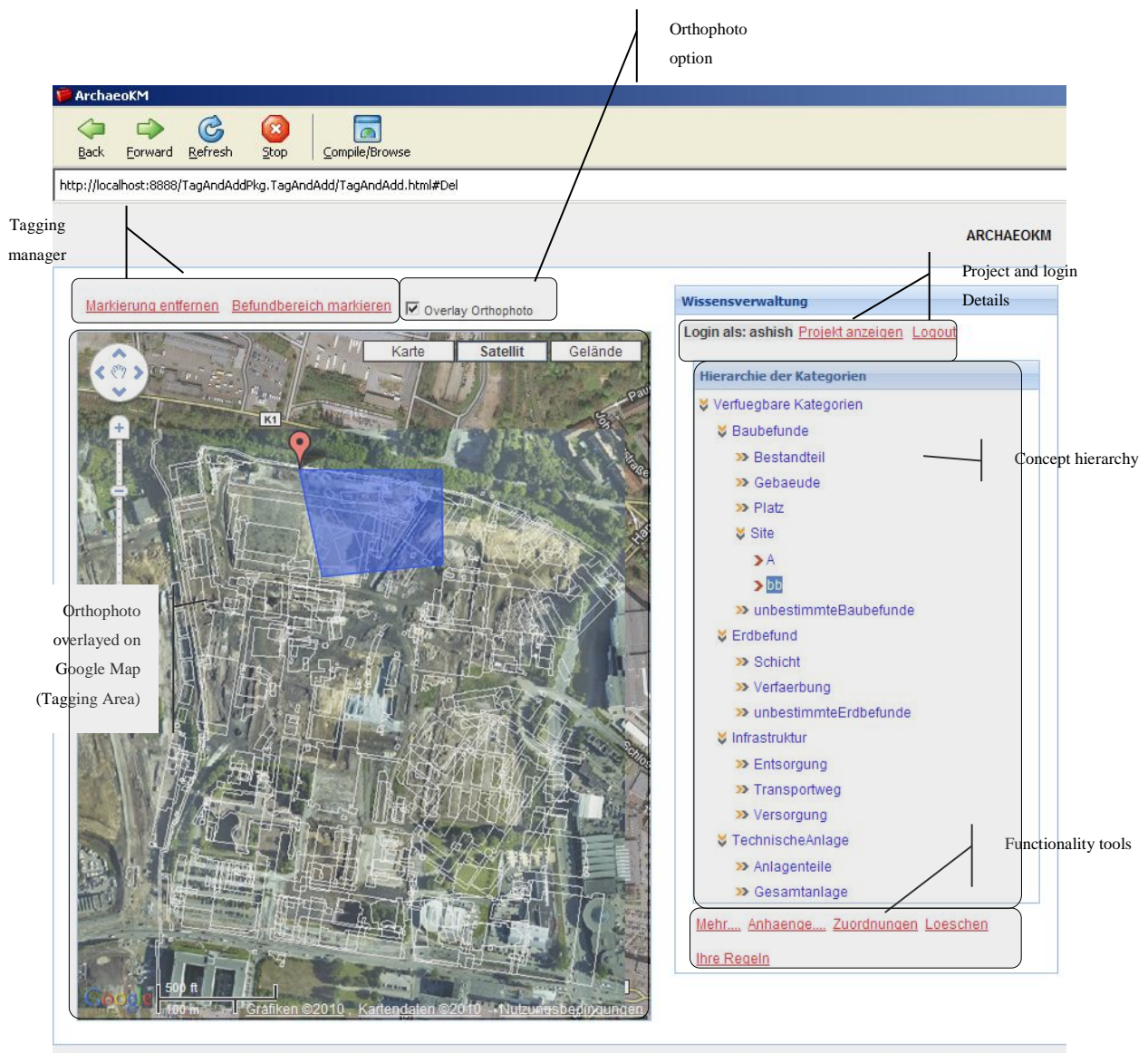

Figure 6.1: The project User Interface of the ArcheaoKM Web platform

The project page is composed of six major parts. The main area is covered by a map from Google Map and is overlayed by an orthophoto of the area (ArchaeoKM currently handles 2D data as it is only a demonstrating tool and intends to be extended in handling 3D datasets). The area is the tagging area where objects are spatially identified and tagged. The sections over it are dedicated for the tagging manager which controls the tagging process and orthophoto option. The orthophoto can be toggled on and off in the orthophoto Option section. The sections on the right are mostly dedicated to knowledge management and visualization process. The section on Project/Login Detail provides information details on the project site as the area, the location and so on. However, the section below is the concept hierarchy which is a hierarchical structure of the ontology that defines classes and their individuals prefixed by a red horizontal arrow heads (textual symbol) in the figure 6.1. The bottom section (Functionality Tools) is mostly dedicated to knowledge processing functionalities through the tools presented in the section.

The coming sub sections demonstrate the implementation of the concepts through the 4Ks principle presented in previous chapters (Knowledge acquisition, Knowledge Management, Knowledge Visualization and Knowledge Analysis). They also provide the continuation to the principles of the functionalities discussed in last chapter. The section also presents an application of a working example of industrial archaeology.

6.1.1. Knowledge Acquisition

The process of knowledge acquisition consists in identifying the object on the archaeological site and to enrich the ontology with the new kind of object, or to populate the ontology with this new object. The ArchaeoKM platform provides a tool to identify and tag the object on its Tagging Area section. The tagging process not only helps in identifying the object in the map, it populates the ontology. This populated ontology acts as the knowledge base for knowledge processing activities.
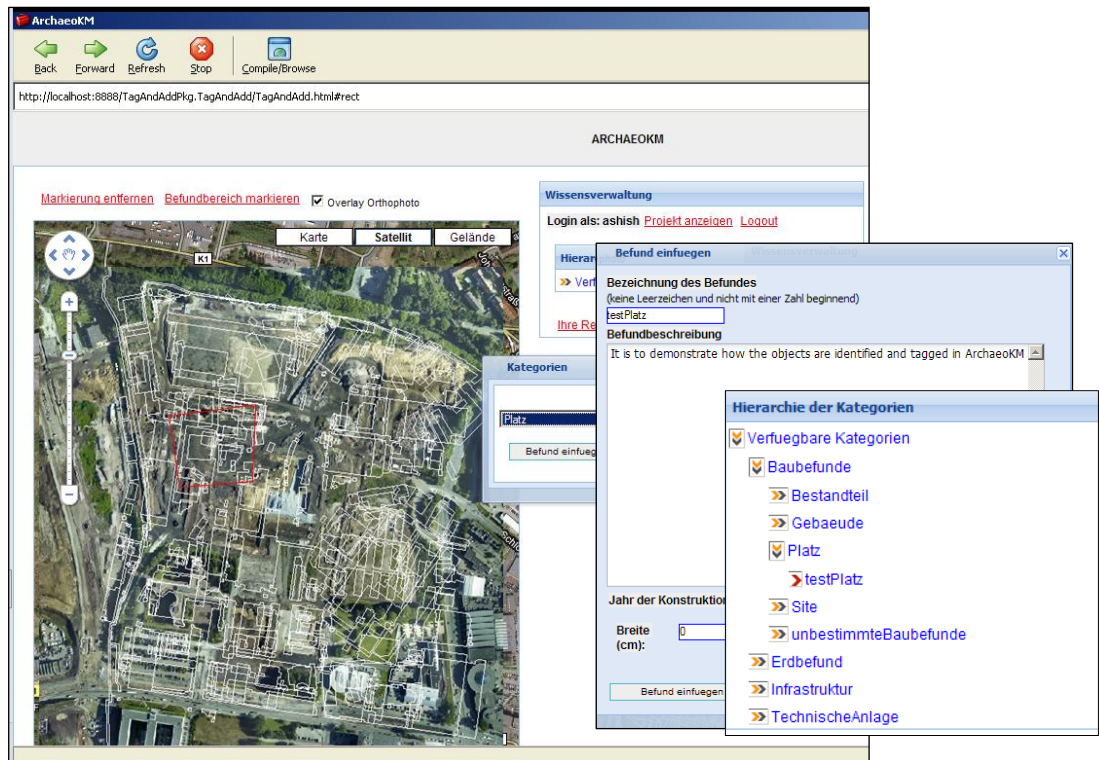
Figure 6.2: The identification, tagging, enrichment and populating process

From the user interface point of view, in order to tag the map with an object one has to click on the hyperlink *Befundbereichmarkiern (Tag/Mark the object)* in *Tagging Manager* section. This activates the tagging option and one could move around the map and draw either a line or a point or a polyline to tag the object. For simplicity, the tagging process with polygon is used here to demonstrate the whole process. Once, the tagging option is activated, a polygon is drawn inside the Tagging Area to outline the boundary of the object identified. As soon as the polygon is closed, the application asks the appropriate class in which the object is belonging to prompt up. One has to choose in which class the identified object belongs. This process instantiates the classes through the enrichment of individuals. During this instantiation process, the details of the object need to be entered which provide first semantics to the object. Figure 6.2 illustrates the tagging process and the enrichment of the ontology.

From the ontology point of view, in the example of figure 6.2, the individual `feat:testPlatz` is instantiated into the class `feat:Platz`. The object `feat:testPlatz` is requiered to be attached with its semantics definition in order to define it. The first set of semantics are attached through describing the basic properties of the object such as description `feat:description`, date of construction `feat:dateCons`, the polygon placement

`feat:localPlacement` and so on. The coordinates are stored in `feat:localPlacement` as soon the instantiation process is completed and the coordinates are automatically controlled by the application. As mentioned in chapter 5, an individual of the nature of shape, here a polygon, is created in the relevant class, here `shp:polygon`, which has its own data property `feat:localPlacement` in which the coordinates are stored. This individual `shp:polygon` then gets mapped to the `feat:testPlatz`, an individual of `feat:Platz`, through the object property `feat:hasIndex`, thus linking the object to its coordinates. In addition, it is also important to give the object its connection to relevant data and documents that are collected during the excavation process. This is carried out through semantic annotation process. As already been mentioned in the last chapter, the ArchaeoKM platform provides the semantic annotations to three categories of data and documents: images, archives and geometry through Image Annotation, Archive Annotation and Spatial Annotation.

6.1.1.1. Image Annotation

From the user interface point of view, the identified object can be semantically annotated by clicking on the hyperlink Anhaenge…(Appendices…) on the Functionality Tools section. The annotation window constitutes of three tabs dedicated to each of three categories of annotation. The first tab is Image Annotation shown as Bilder (Image) in the figure 6.3. The tab consists of list of images in the local server, their previews when clicked and their properties as format, source (Quelle) and date of capture (Erstellungsdatum). The semantic of the image with respect to its nature to the object is given in Bildbezeichnung (Image Title). Once all the details are given and press Einfuegen (Annotate) button, the image is semantically annotated to the object through its semantics given in Bildbezeichnung.

Technically, the process of image annotation needs an image as an object within the knowledge base. It is hence required to instantiate the image document inside the appropriate class. As the document is of type image, it creates an instance of the image within class `doc:image` (sub type of `doc:document`). The mapping process between the individual of image document and the feature `feat:testPlatz` is carried out with the help of intermediate class of `ann:tag` and its subclasses. First an individual is created within respective subclass of `ann:tag` and this class is related to `feat:testPlatz` with object property `ann:hasImageAnno`. Then, the intermediate individual is related to the instance of image

through the object property `doc:hasImage` thus completing the annotation process. The complete background procedure of annotation is hidden from the users.
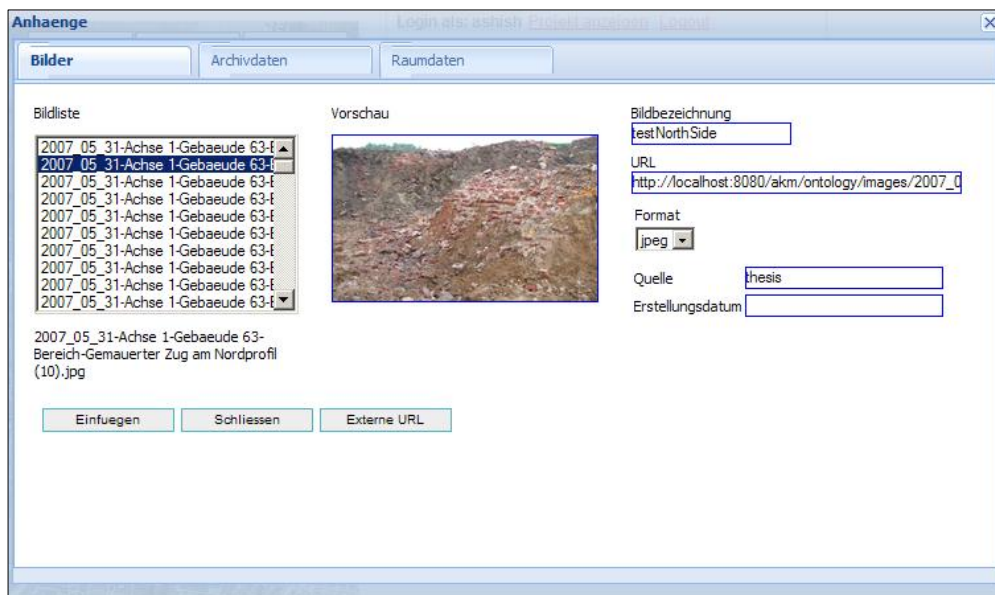

Figure 6.3: The Image Annotation of the object testPlatz

It is also possible to annotate images from external sources through clicking Externe URL (External URL) button. This will open a window to navigate to the external image source and enter the external URL. Figure 6.4 shows the interface to navigate to external image source.
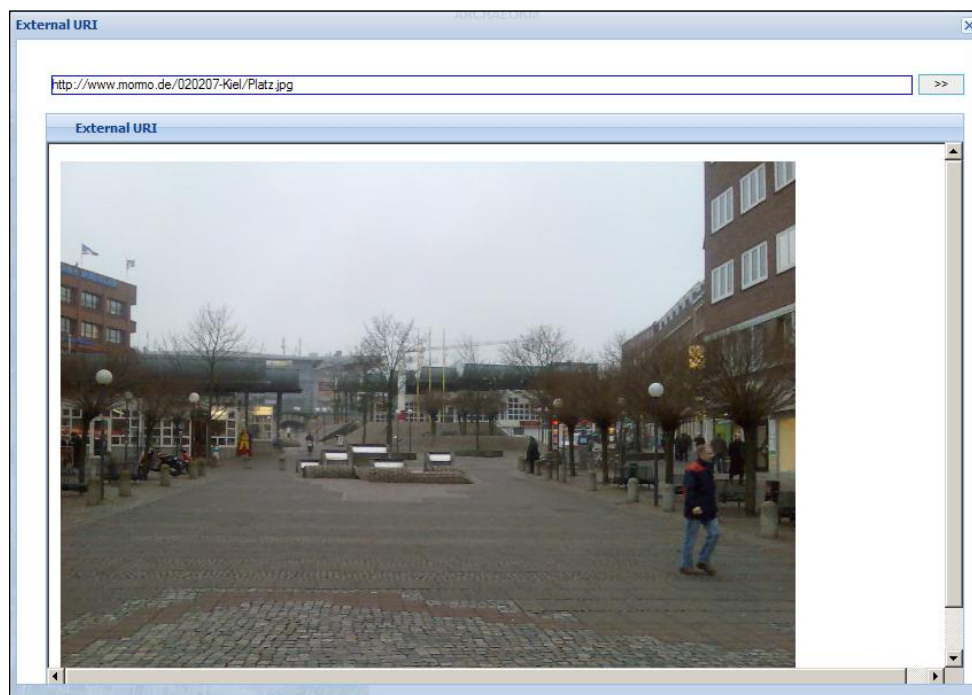

Figure 6.4: The navigation to an external image source for the image annotation

To demonstrate the application two images were annotated for the object `feat:testPlatz`. The first annotation `doc:testNorthSide` is the instance of image in the local server and `doc:testPlatz` is the instance of image of the external server (www.mormo.de). The visualization of both annotations is presented in Knowledge Visualization section.

6.1.1.2. Archive Annotation

The second tab in the Annotation window constitutes the annotation functionality for archiving data annotation presented by Archivdaten. The procedure that is opted in annotating images is again opted annotating archive. The difference here is that the documents are annotated with a wider ranges of document types which range from simple text files to CAD documents in dxf files. As with the image annotation, the tab for archive annotation consists of provision to attach semantics of the annotated documents in terms of its nature related to the object and properties of the documents to themselves. The interface is given in figure 6.5. As with the image annotation the field Bezeichnung provides the semantic of the annotation by giving relevant terminology to define it. Likewise the field Lesezeichen/Seite provides the section of the document where there is reference to the object in the document. Besides that other data properties like the format, type, publication date etc. provide information about the document themselves. The tab also consists of the list of archive documents present in the local server. It also provides functionality to annotate external documents through Externe URL (External URL).
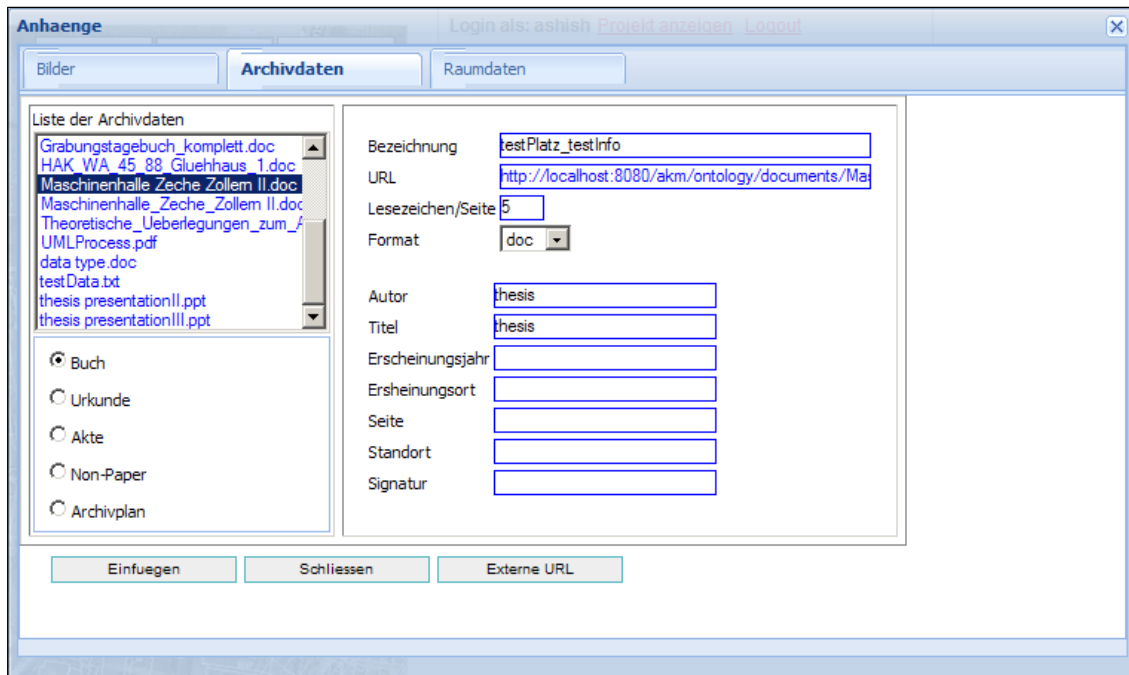
Figure 6.5: The Archive Annotation of the object testPlatz

From the ontology point of view, the button Einfuegen (Annotate) annotates the document to the object through the annotation *testPlatz_testInfo*. In the process, the new individuals are created under `doc:archive` and `ann:tag`. The individual of `ann:tag` is related to `feat:testPlatz` through `ann:hasArchiveAnno`. Similarly the individual of `ann:tag` is related to the individual of `doc:archive` through `doc:hasArchive`. These processes of relating object first with the corresponding tag and then the tag to the archive provide the mapping of the documents to the identified object.

6.1.1.3. *Spatial Annotation*

The last tab in the Annotation window is for the Spatial Annotation. The spatial signatures of the identified objects are already been defined in the spatial database while tagging them on the orthophoto. It is also possible to spatially annotate point cloud. However, it has to be noted that these spatial signatures cannot be used directly to annotate the spatial contents for two reasons. The first reason is that sptial signatures defined with Google Map are in WGS84 coordinate system and the geometrical information in the spatial signature of point clouds is in the GK zone II coordiante system. Without converting the coordinates, links between these two kinds of spatial signatures cannot be undertaken. The second reason is that using the

boundary to extract the coordinates inside the point cloud provides problem in terms of computational performance as they need to be extracted in Web environment.

Hence, a simple and effective method is implemented in the ArchaeoKM platform in order to retrieve spatial signature of the object from clouds of points. First, the Minimum Bounding Rectangles (MBRs) are extracted from the coordinates of the identified objects. The tagging coordinates from Google Map interface are used to define these MBRs. Actually, the MBRs coordinates are transformed into GK Zone II in order to extract the points within the MBRs. Finally, the extracted points from the point cloud regarding the MBR are stored in a file and annotated to the object providing thus a spatial signature for spatial processings.

From the user interface point of view, the spatial annotation tab Raumdaten in the Annotation window contains the scan of the site or at least the scan of the area where the object has been tagged. Again through Bezeichnung the semantic nature of the spatial annotation in relation to the object is defined. The spatial annotation process provides the functionality of choosing the MBRs of objects manually or automatically. Through the automatic process, the transformmed coordinates MBRs are loaded into the fields for XminYmin and Xmax Ymax. The Zmin and Zmax needs to be entered manually as the MBRs of tagged objects are based on 2 dimensional tagging. It also provides the details about database entry of these MBRs. These spatial entries on database are used during spatial operations and functions while enriching the knowledge base spatially. The first step is to create the point cloud file for the object which is an extract of the original scan. The figure 6.6 illustrates the spatial annotation tab where all the details are filled in for extraction of the point cloud file and database entry of the MBR of object testPlatz.
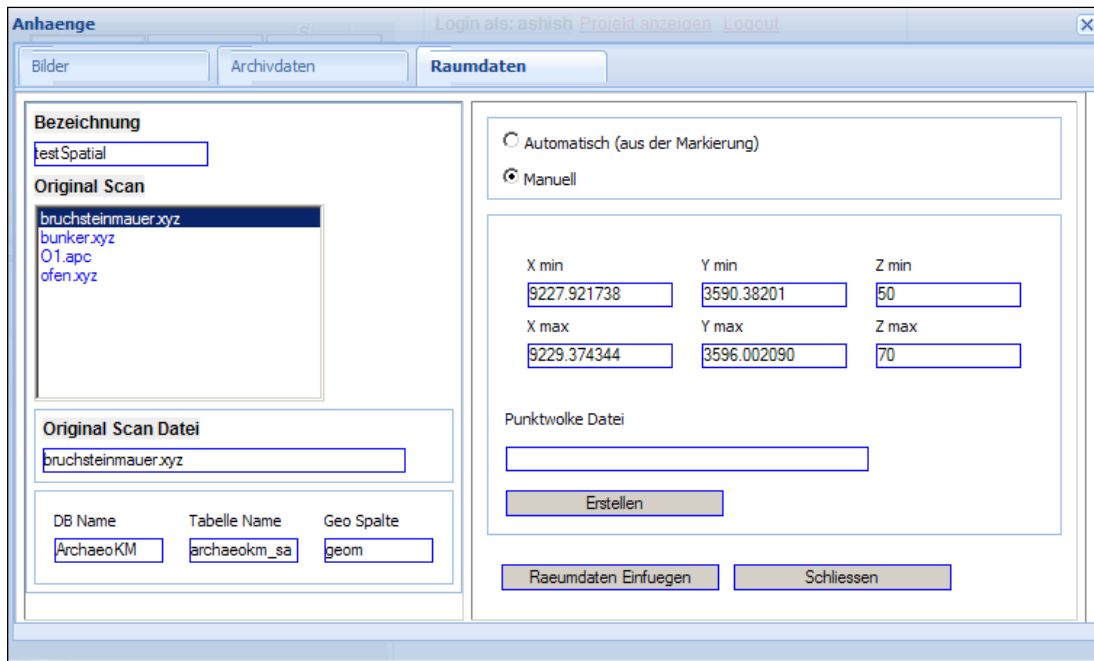
Figure 6.6: The generation of a point cloud file and the spatial database entry for the testPlatz objects

Once the original scan file is chosen for extracting the point clouds of the object from it and the details of database entry is filled, the Erstellen (Generate) button is pressed and this generates a point cloud file with the name testPlatz.apc. Additionally, it enters the MBR of the object into the database system as spatial coordinates. The generation of .apc file and entry in database table is shown in figure 6.7.
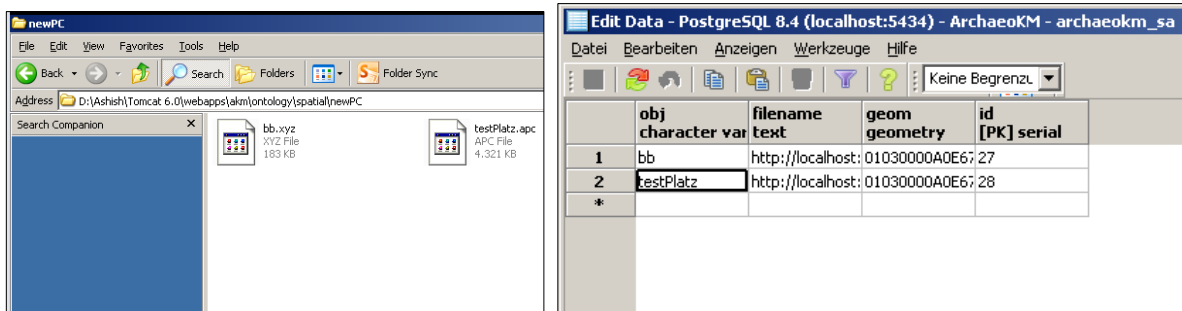
**Figure 6.7:** The point cloud file and the database entry of MBR

The creation of a point cloud document of the identified object starts up the annotation process. As it is document file, it follows the annotation steps of image and archive annotation. Along with the creation of point cloud file, the spatial signature of the object is created in the database system through the MBR storage in it. The annotation procedure of point cloud file have first instantiating the file in the class `doc:spatialDoc` and then mapping to `feat:testPlatz` through the intermediate individual of class `ann:Tag` and object

properties `doc:hasSpatial` first and then `ann:hasSpatialAnno`. It is almost the same procedure that is implemented in archive annotation. In case of the database entry of the spatial signature, a new instance is created `doc:spDatabase` and this individual stores the information about the database like the database name, table name and the geometry column through the data properties of `doc:dbName`, `doc:tableName` and `doc:spName` respectively. Now, once again this individual is mapped to the object `feat:testPlatz` through the same procedure implemented in other annotations. That is an individual under `ann:tag` maps the individual under `doc:spDatabase` through `doc:hasSpatial` and to object `feat:testPlatz` through `ann:SpatialAnno`, thus linking the object to its spatial entry in the database.

6.1.2. Knowledge Management

Knowledge is represented through the link in the ontology and the link is the relationship between two nodes. It has been mentioned the ArchaeoKM platform provides the relationships primarily through object properties `feat:objRel` and `att:hasAttribute` to relate within each other's and to their attributive properties respectively. In addition, the platform provides also an interface to manage those relationships.

To demonstrate the relationship, a new object is tagged within `feat:testPlatz` and instantiated it as an individual M001 of class `feat:Mauer` (Wall). Figure 6.8 a. presents the tagging process. Once it is populated in the ontology, `feat:M001` can be related to `feat:testPlatz` through certain relationship.
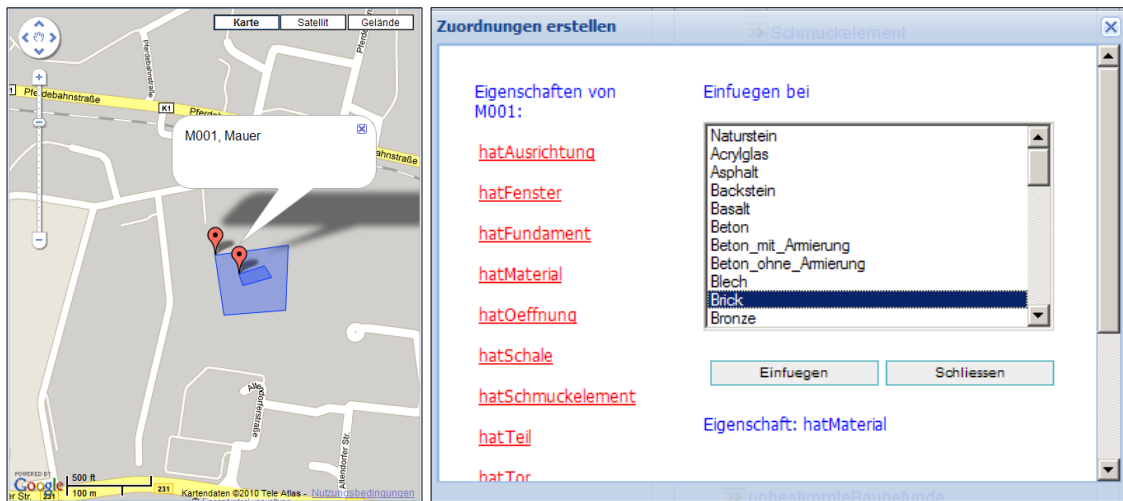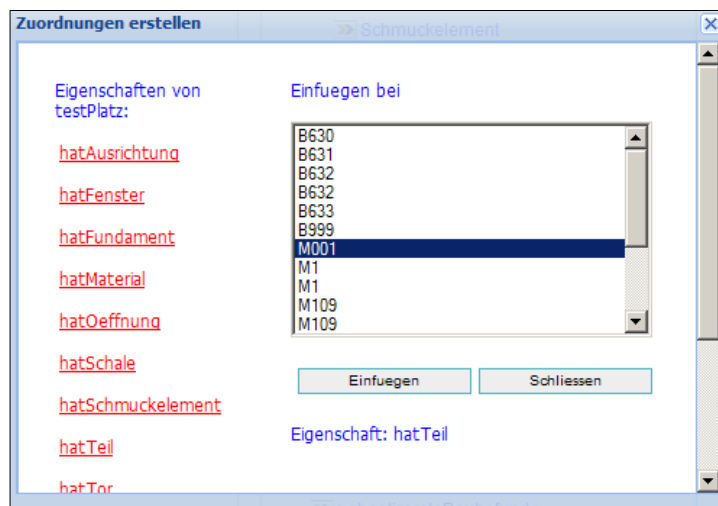
Figure 6.8: a) The new object M001 b) The relation of M001 to Brick c) The Relation of testPlatz to M001

The relationship window is prompted up through clicking the hyperlink Zuordnungen (Relationship). First, figure 9 b. shows the mauer `feat:M001` is made up of brick i.e., `feat:M001` is related to `att:Brick` through `att:hatMaterial` object property. This is to demonstrate that the certain attributive properties that might be necessary to evaluate in themselves are kept as individual of separate class (`att:Attribute`) and linked through specialized property of `att:hasAttribute` object property. The individual `att:Brick` is an instance of specialized class `att:Material` subclass of `att:Attribute`, and `att:hatMaterial` is specialized object property of the object property `att:hasAttribute`. Lastly, figure 6.8 c. demonstrates the relationship of testPlatz to M001 through `feat:hatTeil` object property which is specialized object property of `feat:objRel`.

6.1.3. Knowledge Visualization

The Knowledge Visualization window displays the knowledge that has been fed into the ArchaeoKM platform. The window gets prompted by clicking hyperlink *Mehr* (More) in the main interface of the application. The window includes of three major sections as shown in figure 6.9.
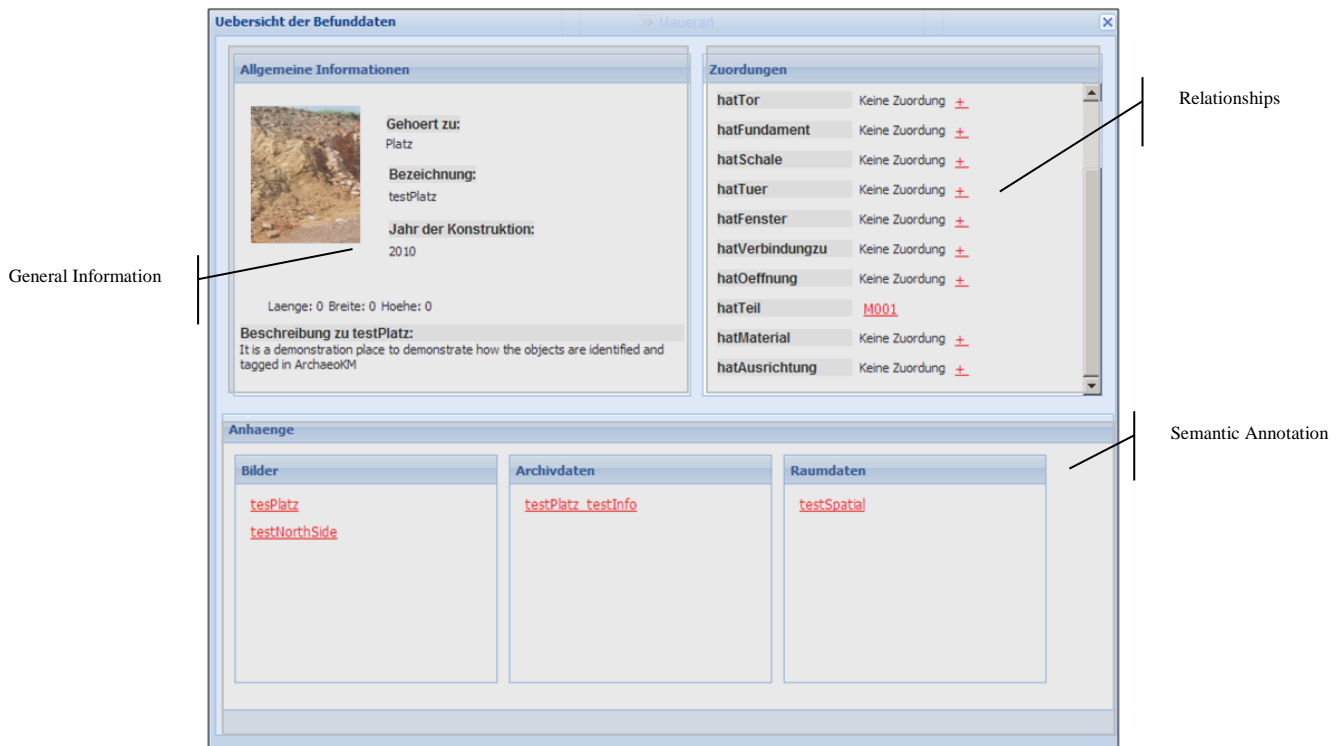

Figure 6.9: The Knowledge Visualization Interface

The General Information section provides general description of the object. They mostly reflect data properties of the object. The Relationships section provides relationships the object has with other objects and the attributive properties. This section is a navigational section providing the details about the related objects through the hyperlinks within. It contains every possible relationship that the object can have. In case there is no object related to then the hyperlink provides navigation to the *Relationship* window where one can define new relationships. The bottom section is the Semantic Annotation section. This section has three different sections to display the three different annotation types.

The first part is for Image Annotation shown here in *Bilder*. As could be seen, there are two annotations: *testPlatz* and *testNorthSide* representing two annotation activities performed in

the previous section: one for the images in the local server and another for the one from external source. These annotations can be navigated to open details about them. This could be seen in figure 6.10.
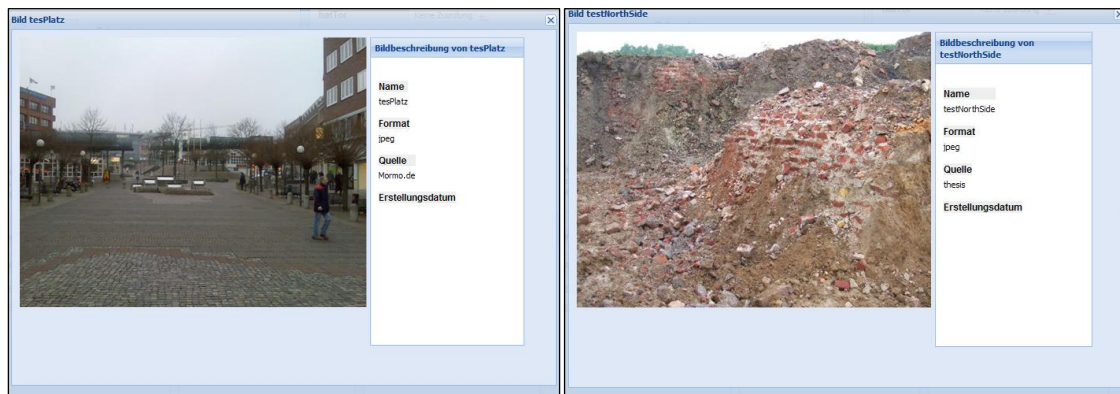

Figure 6.10: The visualization of image annotations of testPlatz

The middle part is for Archive Annotation shown by *Archivedaten*. There is only one annotation *testPlatz_testInfo* in this column as only one document was annotated to demonstrate the annotation process in last section. The annotation is a hyperlink which could be clicked to open detailed window about it. Figure 6.11 shows this detailed window of annotation *testPlatz_testInfo*. As could be seen the window consists of details about the annotation and the document. It also contains a hyperlink *Bittehierklicken* (Please click here) to visualize or download the annotated document.
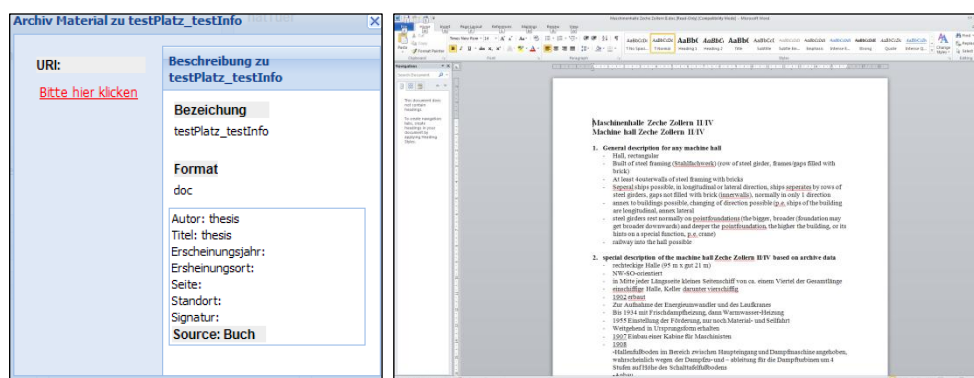

Figure 6.11: The Archive annotation details window with the annotated document

The last part is for the spatial annotation shown by *Raumdaten*. As with the previous two annotations it is populated with an object *testSpatial* denoting the annotation performed in last section. It is presented as a hyperlink in the visualizing window. This hyperlink too prompts up to a window displaying the details of spatial annotation of this type. Figure 6.12

shows the prompted window of the spatial annotation *testSpatial.* It could be seen that the detailed window consists of two parts: Scan Data part shown by *Scan Datei* and Database Entry part shown by *Raeumliche Databank*. The first part presents the scan data files that were annotated. As could be seen it has two files: *Original Scan* presents the original scan file containing the point cloud of complete site and *Scan Datei* which is the generated point cloud file during the annotation process. Both of them could be downloaded and opened in point cloud editing software.
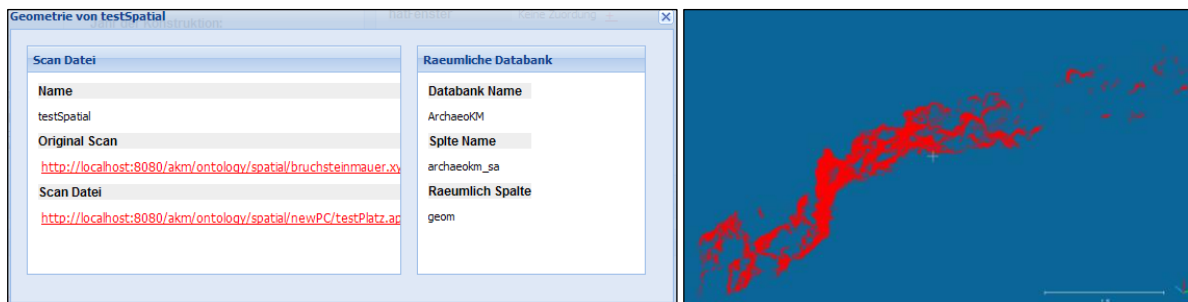

Figure 6.12: (a) The Spatial Annotation detailed window (b) The Scan data of the object

Next is the spatial data entry part. This part provides the information about the databank, the table in which the geometry of the object is stored and the geometry column. This information provides the spatial signature of the object.

6.1.4.   Knowledge Analysis

The ArchaeoKM platform is a rule based system and rules are used to analyze knowledge within the application. The SWRL language is used to infer knowledge in the context of industrial archaeology. The rule interface can be opened through clicking hyperlink *IhrRegeln (Your Rules)* in the main window. The interface for applying rules contains the list of classes within `feat:Feature` which can be navigated through their subclasses. Once a class is chosen to apply the rule, the interface projects show every relationship that the class can have. For example, in the case presented here the class `feat:Platz` is selected as `feat:testPlatz` is the individual of the class. When `feat:Platz` is selected, the interface provides a list of relationships that are possible with the class. Then the relevant relationship needs to be chosen. This again prompts the classes that are within the range of the property with domain as the class first chosen (class `feat:Platz` in this case). The object property `feat:hatTeil` is selected to continue the execution of the example presented in this chapter.

This will enlist the list of classes that are range of the property `feat:hatTeil`. Class `feat:Mauer` is one of such classes, it is chosen to extend the rule statement *Platz hatTeil Mauer (Place has Part Wall).*

In the continuation of the example, once class `feat:Mauer` is selected as range of object property `feat:hatTeil`, it again lists all the properties class `feat:Mauer` can have. The `hatMaterial` is then selected and this lists material types which are basically individuals of class `att:Material`. As `att:Material` is a attributive class it lists the individuals directly. The individual *Brick* is selected to complete the rule statement *Platz hatTeil Mauer hatMaterial Brick* (Place has Part Wall has Material Brick). Clicking *Deutet auf* (Implies to) button, one get all the possibilities that the rule can project. For the case, class *Raum* is selected completing the rule statement *Platz hatTeil Mauer hatMaterial Brick Deutet auf Raum* (Place has Part Wall has Material Brick implies to Room). Once executed, the resulted individual *testRaum* will be populated in class *Raum.*

This interface provides expert users to enter or to interact with the rules directly from the IhrRegeln entry option. The rules are executed through SWRL but the interface hides the complications of converting the clicking hyperlink sequences into inferring and enriching the result in the knowledge base. Figure 6.13 illustrates the execution of the rule and enrichment of the result.
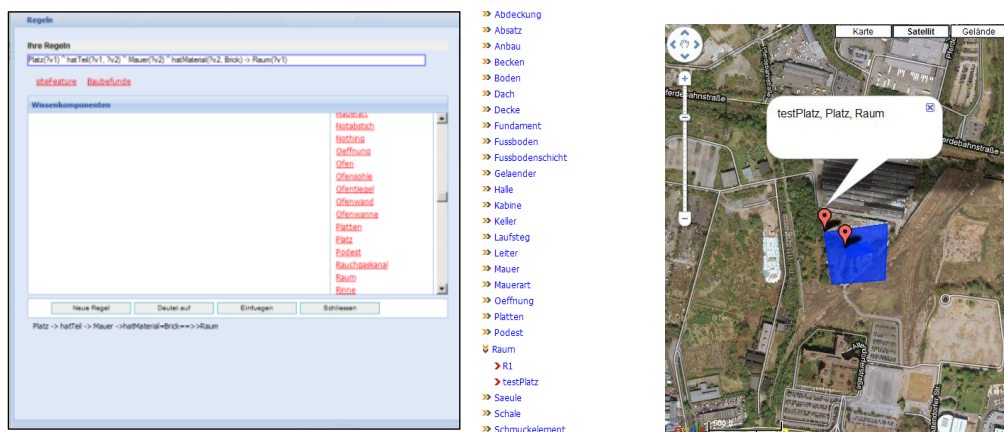

Figure 6.13: The execution of the rule

6.1.5. A Working Example

The research activities include testing the ArchaeoKM platform in a real environment. To demonstrate the test is successful, a section of a model created by archaeologists is

implemented within the ArchaeoKM platform. The result of the testing highlights the implacability of the application in real sites. This section presents the section of the model tested and the results obtained.

Identified objects B631, B632 and B633 as *Mauer* (Wall) and B634 as *Verfaerbung* (Discoloration) with their properties

- `B632` **`hatAusrichtung`** `O-W`
- `B633` **`hatAusrichtung`** `N-S`
- `B631` **`hatAusrichtung`** `N-S`
- `B634` **`hatFarbe`** `rot`

The relationships with other objects are determined through the relationship *hatVerbindungzu* (*hasConnectionTo*) and implemented as

- `B632` **`hatVerbindungzu`** `B631`
- `B632` **`hatVerbindungzu`** `B634`
- `B631` **`hatVerbindungzu`** `B632`
- `B633` **`hatVerbindungzu`** `B632`
- `B633` **`hatVerbindungzu`** `B634`

The relationship *hatInnenschale* (*hasInnershell*) is implemented as

- `B631` **`hatInnenschale`** `Innenschale` **`hatMaterial`** `Schamotteziegel`
- `B632` **`hatInnenschale`** `Innenschale` **`hatMaterial`** `Schamotteziegel`
- `B633` **`hatInnenschale`** `Innenschale` **`hatMaterial`** `Schamotteziegel`

Again with relationship *inGewachsenemBoden* (*inNaturalGround*)

- `B631` **`inGewachsenemBoden`**
- `B632` **`inGewachsenemBoden`**
- `B633` **`inGewachsenemBoden`**

Finally,

- `B633` **`hatVerbindung`** `zu B634`
- `B634` **`hatFarbe`** `rot`

The knowledge acquired and managed through the relationships presented above could now be analysed through the rule shown in table 1:

Rule	Implementation/Result	Intermediary SWRL
If a finding has walls orientate to *N-S* and these walls have connection to walls having orientation *O-W* it hints that the finding may be a room	*Objekt* (Object) *hatTeil (*hasPart) *Mauer* (Wall, *result -* **B632**) *hatAusrichtung (hasDirection)* O-W and *hatVerbindungzu (hasConnectionTo)* Mauer (Wall, result -**B633** and **B633**) *hatAusrichtung* N-S →Implies to: *Kammer (Room)*	Objekt(?x) ^ hatTeil(?x, ?y) ^ Mauer(?y) ^ hatAusrichtung (?y, O-W) ^ hatVerbindungzu(?y, ?z) ^ Mauer(?z) ^ hatAusrichtung(?z, N-S) → Kammer(?x) Object(?x) ^ hasPart(?x, ?y) ^ Wall(?y) ^ hasDirection (?y, O-W) ^ hasConnectionTo(?y, ?z) ^ Wall(?z) ^ hasDirection(?z, N-S) → Room(?x)
If the finding have walls having inner side made of fire resistant bricks it hints that the finding may be an oven	*Objekt* (Object) *hatTeil* (hasPart) *Mauer* (Wall, result - **B631, B632, B633**) *hatInnenschale* (hasInnerShell) Innenschale (InnerShell) *hatMaterial*(hasMaterial) *Schamotteziegel*(Fireclaybrick)→Implies to: *Ofen* (Oven)	Objekt(?x) ^ hatTeil(?x, ?y) ^ Mauer(?y) ^ hatInnenschale (?y, ?z) ^ Innenschaler(?z) ^ hatMaterial(?z, Schamotteziegel) → Ofen(?x) Object(?x) ^ hasPart(?x, ?y) ^ Wall(?y) ^ hasInnerShell (?y, ?z) ^ InnerShell(?z) ^ hasMaterial(?z, Fireclaybrick) → Oven(?x)
If the findings are beneath the earth and are ovens then they are deep ovens	*Ofen* (Oven) *hatPosition* (hasPosition) *GewachsenemBoden* (NaturalSoil) *unter* (Under) → Implies to: *Tiefofen* (DeepOven)	Ofen(?x) ^ hatPosition(?x, unter) → TiefOfen(?x) Oven(?x) ^ hasPosition(?x, under) → DeepOven(?x)

Table 6.1: Rules formulated and their results

Summarizing the above rules we get "the objects which have these *mauers* having *innenschale* with material *Schamottziegel* and connected to *verbaeurbung* with red color and under the ground is *TiefOfen*". This rule could be implemented in the ArchaeoKM platform through *"Objekt hatMauer und Mauer hatInnenschale Innenschale hatMaterial Schamottiziegel + Mauer hatVerbindungzu Verfaerbung hatFarbe Rot + Mauer inGewachsenemBoden → TiefOfen(Objekt)"* roughly translated as <u>Object with walls having inner shell and material as fireclay brick and the walls connected to a discoloration area with color red and is in natural ground is deep oven</u>. The rule is written in SWRL as

```
unbekannteObjekt(?v1)  ^  hatMauer(?v1,  ?v2)  ^  Mauer(?v2)  ^
hatAusrichtung(?v2,  N-S)  ^  hatSchale(?v2,  ?v3)  ^  Innen(?v3)  ^
hatMaterial(?v3,  Schamottziegel)  ^  hatVerbindungzu(?v2,  ?v4)  ^
```

```
Mauer(?v4) ^ hatAusruchtung(?v4, O-W) ^ hatVerbindungzu(?v2, v5) ^
Verfaerbung(?v5) ^ hatFarbe(?v5, Rot) → Tiefofen(?v1)
```

It can be translates in English as

```
unknownObject(?v1)     ^     hasWall(?v1,     ?v2)     ^     Wall(?v2)     ^
hasDirection(?v2,  N-S)  ^  hatShell(?v2,  ?v3)  ^  Inner(?v3)  ^
hasMaterial(?v3,  Fireclaybrick)  ^  hatConnectionto(?v2,  ?v4)  ^
Wall(?v4)  ^  hasDirection(?v4,  O-W)  ^  hatConnectionto(?v2,  v5)  ^
Discoloration(?v5) ^ hacColor(?v5, Red) → Deepoven(?v1)
```

The knowledge processing of the above example implemented in the ArchaeoKM platform is illustrated in figure 6.14 and 6.15. Figure 6.14 illustrates the identification of the objects in the excavation site with the allocation of relationships of one of the objects. Likewise it illustrates the detail description and semantic annotation in the form of image annotation. Figure 6.15 illustrates the execution of rules through in the ArchaeoKM platform. A structure is digitized covering all the walls and connecting the *Verfaerbung* B634. This structure is identified as an object *u0001* under class *unbekannteObjekt* and then related to objects within classes *Mauer* and *Verfaerbung* as through *hatMauer* and *hatVerbindungzu* relationship respectively shown in figure 6.15 a.
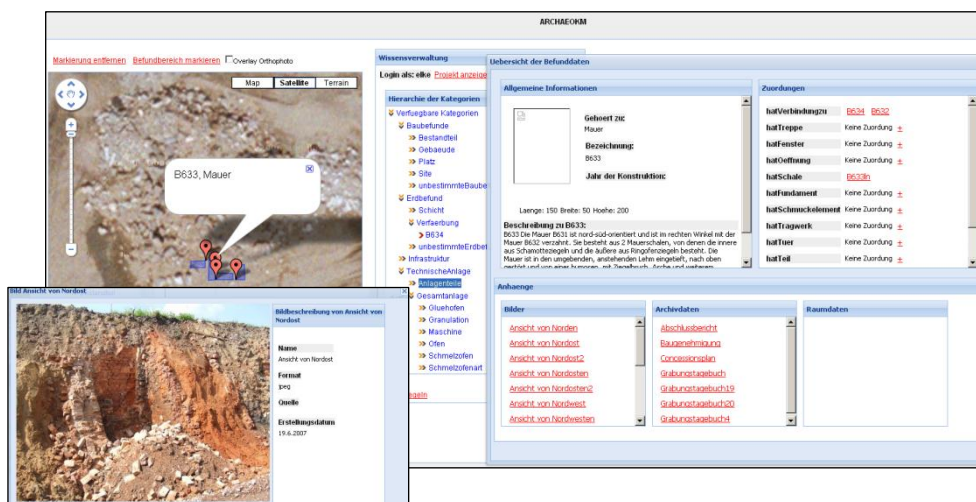

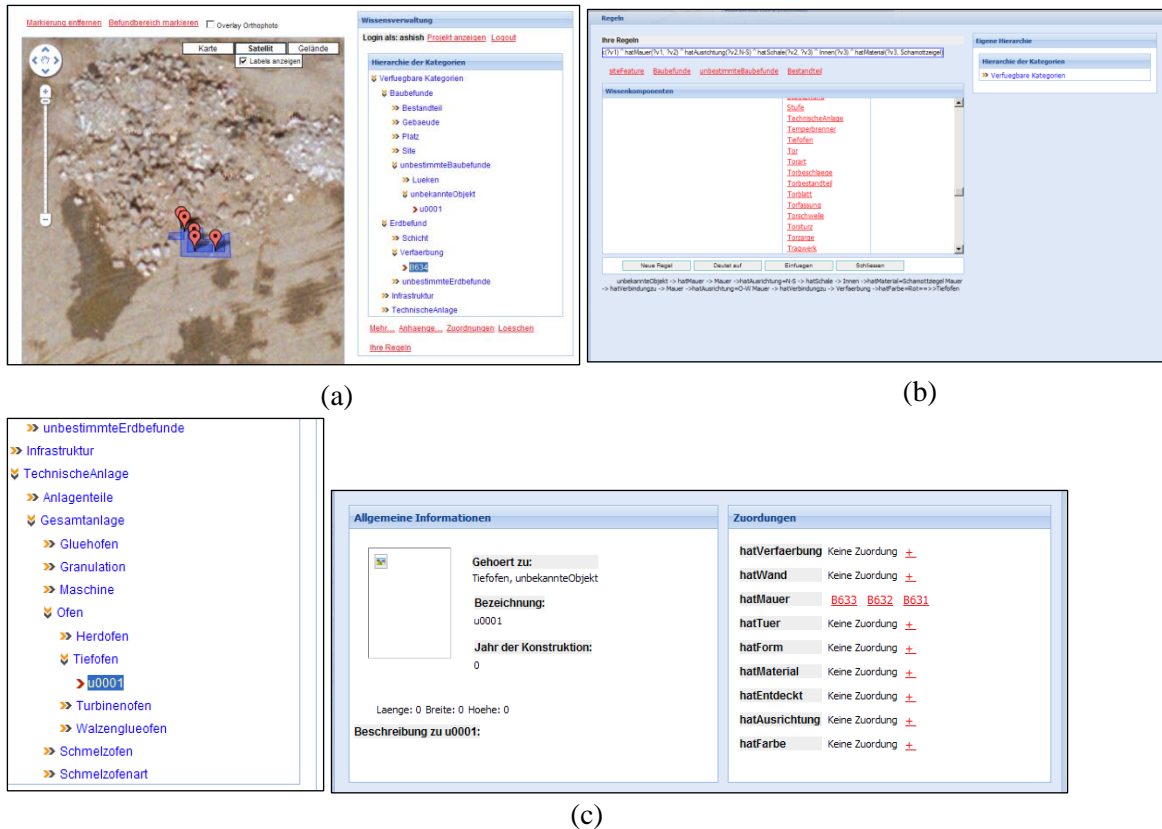**Figure 6.14:** The implementation of the model in the ArchaeoKM platform

Figure 6.15: (a) The structure covering the features (b) the execution of the rule (c) the result

The rule is then executed through Rule window which is shown in figure 6.15 b. Figure 6.15 c shows the result and it could be seen that object u0001 is listed now in the class *TiefOfen* and shares all the relationship of the class.

6.2.   The Spatial Integration Tool – A Proof of Concept

The integration of spatial component within the Semantic Web technologies were carried out through the application developed specifically to demonstrate the core concept of this thesis work. This section presents the results of inferences with spatial rules through spatial built-ins proposed in that section.

Staying with the industrial archaeology the thesis presents an example of spatial analysis in the industrial archaeological sites. Figure 6.16 illustrates the class hierarchy of `feat:Feature` to accommodate the instances that will be used during the demonstration of the example. This example is based on the two categories of findings that are commonly excavated in industrial archaeological sites: part of railway track and remains of the buildings. The buildings are identified within three categories: Machine Halls, Furnaces and Bunkers and populated in the

knowledge base likewise. Similarly, the finding of the part of the railway track is populated too. The instantiation of the individuals for classes `feat:MachineHall`, `feat:Furnace` and `feat:Bunker` and for class `feat:RailTrack` could also be seen in figure 6.16.
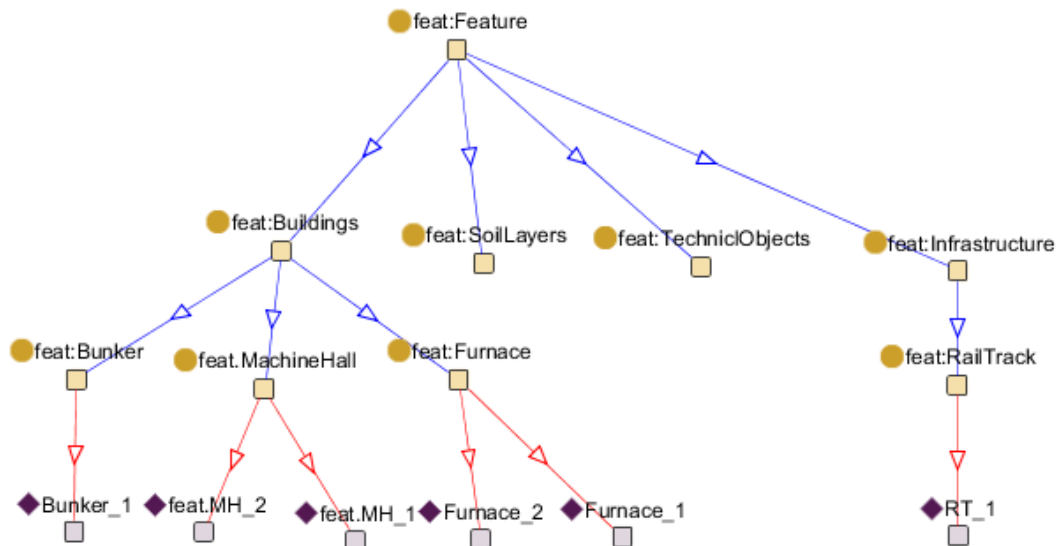

Figure 6.16: The class hierarchy for demonstration of the example

The populations of the individuals within these subclasses are done manually through Protégé (Research, 2010) which could easily be replaced through applications like the ArchaeoKM platform. The coordinates of these individuals are digitized and transferred to the database as spatial data. These individuals are then mapped to the coordinates through the process of spatial annotation. The knowledge base thus is spatially rich and the application tool can infer spatial rules through the spatial built-ins within it.

As already mentioned, a total of four classes were populated with the individuals: `feat:RailTrack`, `feat:Bunker`, `feat:MachineHall` and `feat:Furnace`. The individuals of each class could be seen in figure 6.16 through the leaf nodes of respective classes in the class hierarchy. The individuals are mapped to their spatial signature through the digitization process. The location map of them is illustrated in figure 6.17. It should be noted that they are hypothetical interpretation and only resembles real world scenario but are not from any real locations.
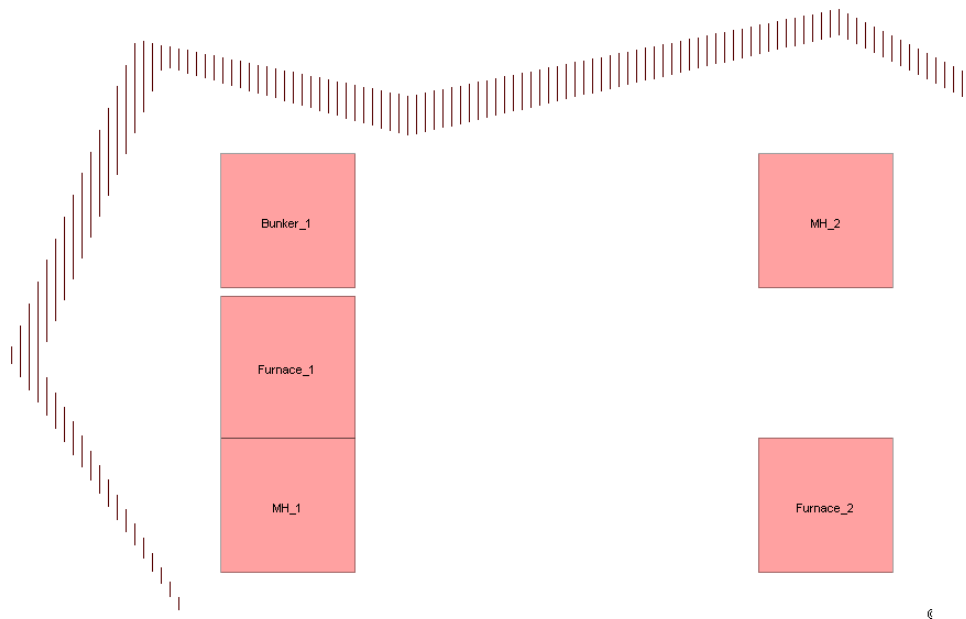
Figure 6.17: The map representing spatial objects in the knowledge base

Figure 6.17 illustrates the remains of buildings and remains of the rail track in the excavation site. A hypothetical GIS analysis of determining the buildings that might have used the rail track to get supplies of the raw materials during the production period of the industrial site is presented as an example here. In order to draw the example closer to the case study presented in this research thesis, the location map is a section of the industrial archaeology site that produces steel. Likewise, the machine halls represented through individuals `feat:MH_1` and `feat:MH_2` consists of machines and one of those machines processes iron ore (one of the raw material needed to produce steel). So, the GIS analysis should determine the machine hall that uses railway track to receive its supply for processing in steel production. In general with a common GIS, a set of activities are carried out and are mentioned in the following sequence:

- *Buffer the Rail Track by certain distance (e.g. 50 meters) to establish a valid possibility to receive the supply from that distance*
- *Determine the machine halls within the buffer region*
- *Determine the raw material needed for the processing in each machine hall to determine which ones are used to produce steel. In this case, the raw material used within the building should be iron ore*
- *The machine hall should be connected to a furnace in order to process the iron ore. So it needs to be checked whether the halls inside the buffer region are touching*

> *furnace*

- *Display the machine halls satisfying all these*

It is a simple example to demonstrate the concept and hence details about other processing steps of steel production are ignored on purpose. For a simple analysis as this it requires at least four steps of spatial analyses as listed above. Here the possibility of an alternative is provided through the spatial extension of SWRL language. It can execute these four steps in one statement. The combination of existing built-ins in existing SWRL and the spatial built-in mentioned in chapter 5 is used to execute this analysis.

```
feat:RailTrack(?x) ^ feat:MachineHall(?y) ^ feat:Furnace(?f) ^
att:hasRawMaterial(?y, att:ore_iron) ^ spatialswrlb:Buffer(?x, 50,
?z) ^ spatialswrlb:Intersection(?y, ?z, ?res) ^
spatialswrlb:Touches(?y, ?f) → feat:hasSupplyLine(?y, ?x)
```

Here the relationship `feat:hasSupplyLine` is an object property with domain and range are the class axiom of `feat:Feature`.

The spatial SWRL rule is processed in the tool to illustrate the applicability of the concept. It includes a platform to enter this SWRL statement. It has a translation engine that first parses the spatial built-ins to execute them at database level and then enrich the ontology with the result. It then converts the spatial statement to standard statement and infers it through an inference engine. Figure 6.18 presents the interface of the demonstration tool.
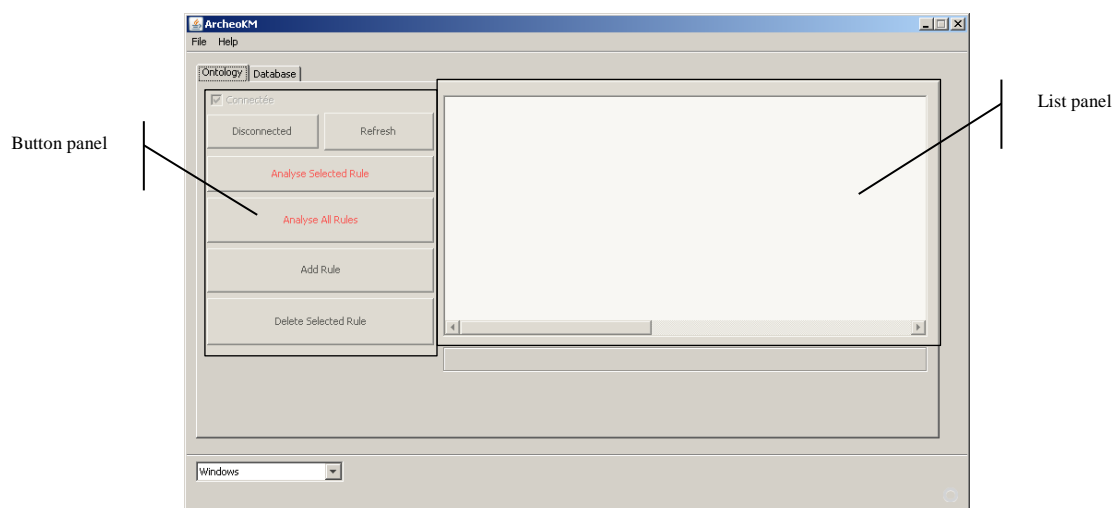

Figure 6.18: The demonstration tool

A new rule could be entered by clicking on Add Rule button. It prompts a window where the rule could be added. On clicking Validate button, the rule gets listed on the List Panel. The process is shown in figure 6.19.
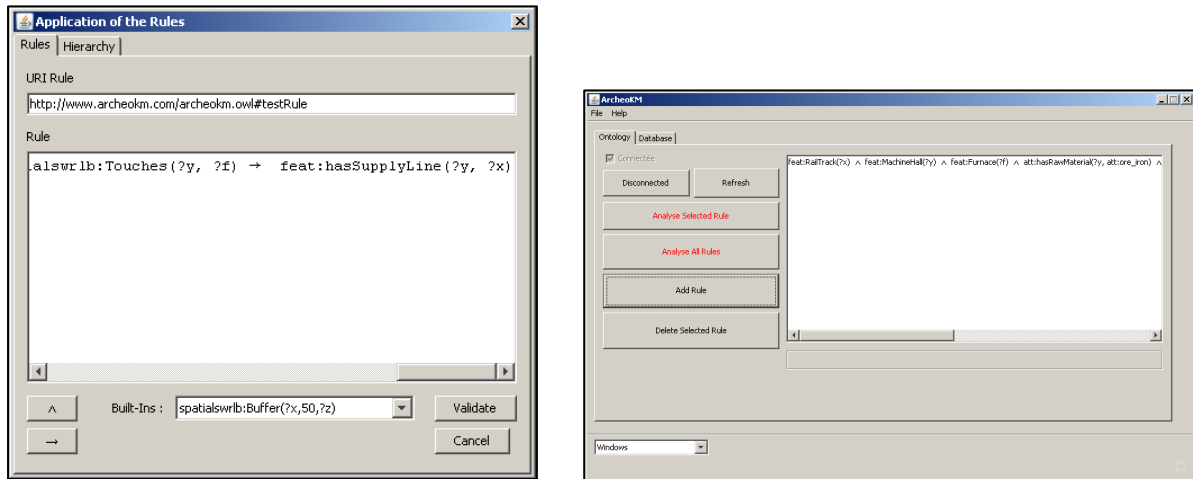

Figure 6.19: Adding a spatial rule the rule list panel

The rule is now to be executed. It could be executed by pressing *Analyze Selected Rule* in *Button Panel.* Once the rule gets executed, it enriches new individuals for classes `sa:sp_Buffer` and `sa:sp_Intersection` as could be seen in figure 6.20.
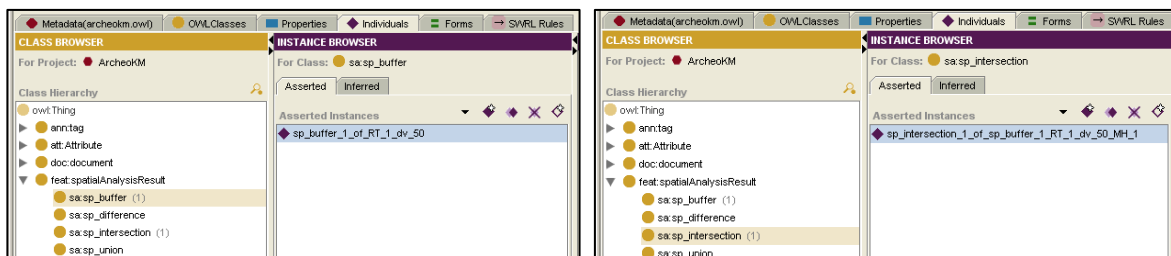
Figure 6.20: The enrichment of spatial components in the knowledge base

It could be seen an individual for class `sa:sp_Buffer` is created which presents the buffer geometry through spatial operation Buffer. Likewise, an individual for class `sa:sp_Intersection` is populated indicating that within the specified buffer zone only one machine hall is intersected. This is true when the result is observed. The result shows Machine hall `feat:MH_1` has raw material `att:ore_iron` and touches the furnace `feat:Furnace_1` and lies within 50 units of `feat:RT_1` satisfying all conditions. The knowledge base is thus populated with `feat:RT_1` in `feat:hasSupplyLine` object property through the execution of the rule. It could be seen in figure 6.21.
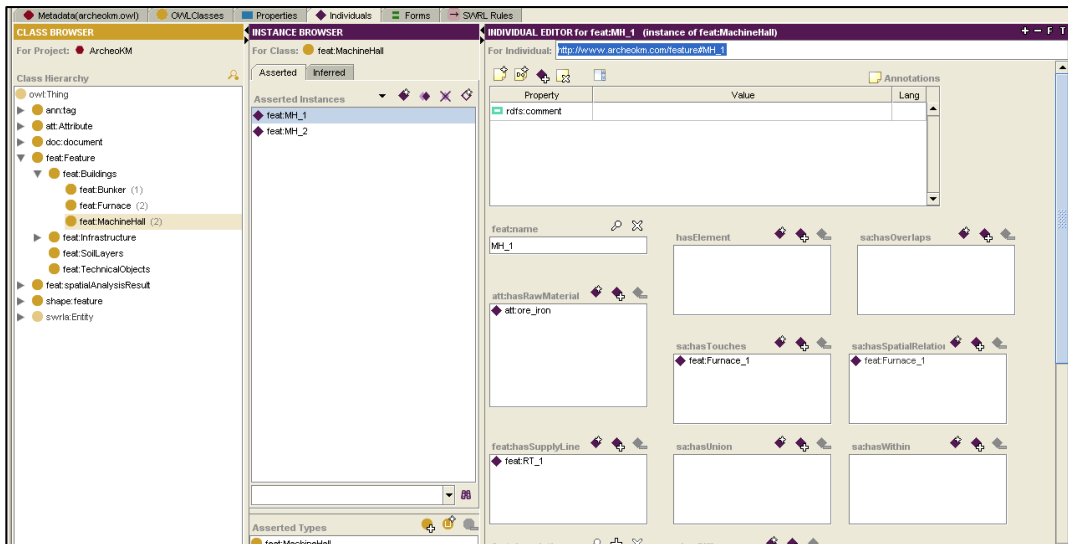
Figure 6.21: The enrichment through the execution of spatial SWRL rule

The machine hall `feat:MH_1`, as shown in the figure is the only machine hall that is within buffer zone of 50 units and is touching a furnace with ore iron as raw material. It is hence populated with `feat:hasSupplyLine` as `feat:RT_1` through the rule. When the same analysis was carried out in GIS (Quantum GIS (Project, 2002)), it was observed that only two buildings are intersecting the buffer geometry of the rail track and only one machine hall - MH_1. This can be seen in figure 23. This is thus evident that the underlying knowledge technologies within the Semantic Web framework could be extended with the spatial features to provide more efficient alternative of current GIS.

Figure 6.22: The result in Quantum GIS

It could thus be concluded that the result obtained from the spatial rules through the spatial built-ins resembles the result obtained from a contemporary GIS. By the same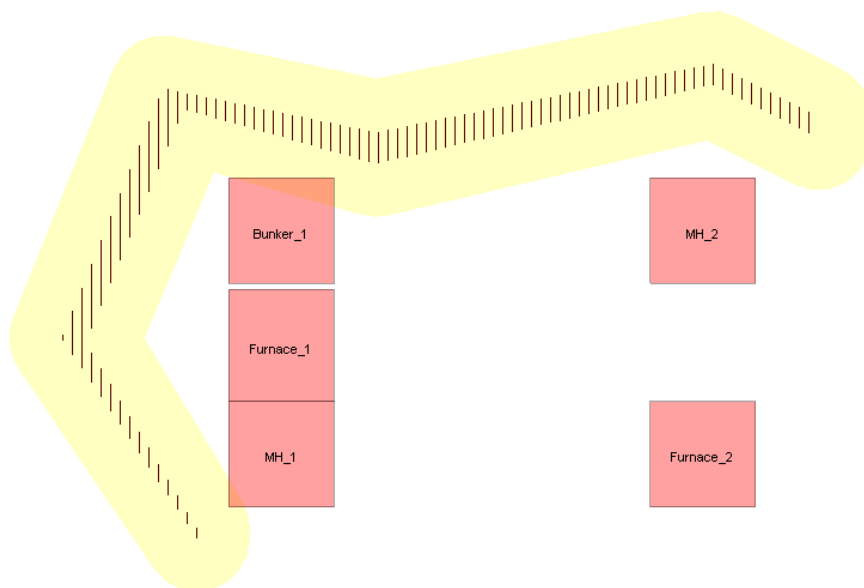 manner, more complex rules can be undertaken on spatial data by using knowledge from the ontology which cannot be undertaken on standard GIS system.

6.3. Application Programming Interfaces

The ArchaeoKM platform is a Web based application designed and developed with client-server architecture. The client uses the potentiality of Ajax (Asynchronous JavaScript and XML) to interact with users at the client side and retrieves data from the server without interfering each other performances. Server side uses Jena framework along with inference engines to interact with the components of Semantic Web tools and techniques. The whole system is designed and developed in Google Web Toolkit (GWT). This section presents short overview of all these technical tools on which the ArchaeoKM platform is based on.

6.3.1. Google Web Toolkit

Google Web Toolkit or simply GWT (Google, 2010) is a Java based framework to create Ajax applications on the Web. Ajax or Asynchronous JavaScript and XML is a technique combining the tools to develop client-side interactive Web application. Ajax is not an independent programming language but a new way of using existing standards. It incorporates several technologies to provide the capabilities of each. Ajax incorporates (Garrett, 2005)

- standards-based presentation using XHTML and CSS;
- dynamic display and interaction using the Document Object Model;
- data interchange and manipulation using XML and XSLT;
- asynchronous data retrieval using XMLHttpRequest;
- JavaScript binding everything together.

GWT brings easiness in writing Ajax applications ignoring the complications that are normal in writing these applications. It provides the Java API which provides the functionalities to build the user interfaces avoiding the complexities of JavaScript and underlying protocols and models. They are all managed through the compiling capability of GWT compiler. GWT also

consists a Remote Procedure Call (RPC) mechanism that provides a simple approach of transferring the object back and forth from the clients and their server.

The ArchaeoKM platform is entirely developed in GWT. The client side provides the user interfaces to interact with the application. They primarily provide the visualization tools or the tools to manage knowledge. These interfaces are designed in Java framework with the features provided through GWT API but they are compiled as Ajax based interfaced when uploaded in the Internet. The RPC provides an interface between the clients and the server. It provides the inputs from the users to manage the knowledge base or requesting results from the knowledge and the server manipulates these inputs and provides the result through java objects.

The overall client/server model of the ArchaeoKM platform is illustrated in figure 6.23. In general every GWT application that is developed to build Ajax application follows similar server-client model. This is the application based modification of Ajax Web application model presented in (Garrett, 2005).
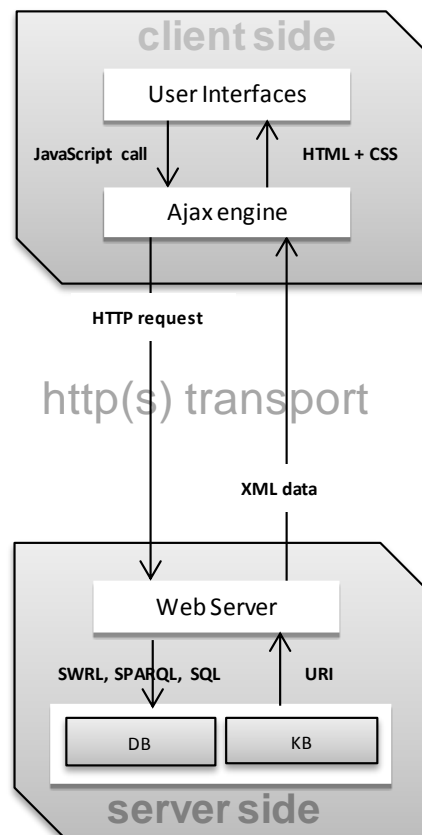

Figure 6.23: The client/server model of the ArchaeoKM platform

6.3.2. Google Maps API

Google Maps API for GWT provides the JavaScript API to allow map functionalities in Ajax based applications developed with GWT. Google Maps API allows embedding Google maps directly on the Web application. It also allows customizing the maps according to the need of the application. The maps are shown through general perspective view which means they are shown from the birds' eye angle. They are not same as orthorectified images as they are images directly obtained from satellites or the aerial flights. The internal coordinate system of Google Map API is geographic coordinate system based on World Geodetic System of 1984 (WGS84) and its baseline resolution in most of the western European countries is 1 meter or less.

Google Maps API comes in different versions but the API supported by GWT through gwt-maps API is Google Maps API version 2. It has been used in the ArchaeoKM platform to facilitate the archaeologists with the impression of the sites they have excavated and the objects they have discovered. The map forms the base to identify and tag the objects thus enriching the objects in the knowledge base. It also highlights the flexibility of the application in handling various sites through the same platform. The maps can be overlayed with the orthorectified images of different resolution than that of Google maps thus providing the flexibility of zooming down to different levels. However, the maximum level of zooming in zooming scale is controlled by Google maps. Nevertheless this overlaying of orthophotos in the Google map provides better understanding of the site area for the archaeologists.

One of the major issues while using the Google map for identification and enriching process is the incompatibility of coordinate systems. As already been mentioned in chapter 2, the coordinate system of the geometries of the excavated objects are in Gauss Krüger Zone II coordinate system and that of Google map is WGS84. It is hence not possible to use the coordinates of the tags directly to process the spatial analysis. Moreover, the extraction of geometries from the point cloud storing geometries of the site is impossible without having certain transformation mechanism from WGS84 to GKII. This transformation mechanism is implemented within the ArchaeoKM platform and used during using this spatial signature for semantically linking to the object.

6.3.3. Ontology Handling Techniques

The research work handles the ontology in two different methods. First one is the design and development of the top level ontology and the second to manage them through the enrichment and population. Design and development of top level ontology is done through Protégé, an open source editor for creating and editing ontologies and knowledge base (Knublauch, 2003), (Research, 2010). It is extensible and platform independent making it good tool to add plugins for effective visualizations of the knowledge base. Its capability to load its frames at the backend database in demands provides high level of performance. The most effective side is the easiness to understand user interface making it ideal tool to start building ontology for any level of experience.

The top level ontology is managed through the Jena API. The Jena API is a Java framework to build Semantic Web applications. It provides a programmatic environment to work with tools and technologies in Semantic Web like RDF, RDFS, OWL and SPARQL (Dickinson, 2009). It is an open source framework for Java that extracts and writes data from and to RDF graphs. Jena provides support to OWL which makes it different from other similar frameworks like Sesame (B.V., 2009). It provides the support to ontology through the extension of ontology model that are capable of handling ontologies.

Jena itself contains inference and reasoning engine and can accommodate third party inference and reasoning engines as well. These engines providing reasoning capabilities could be turned on or off while using the framework to avoid unnecessary details provided through these reasoners. The ArchaeoKM platform uses Jena framework at its server end to manage the ontology. The top level ontology works as base ontology and through their axioms provides framework for the knowledge base. The base ontology is then enriched through different functionalities of knowledge processing activities within the application thus changing it to the knowledge base where archaeologists could interpret and execute their knowledge models.

ArchaeoKM is a rule based application and hence uses inference engines to infer rules within. Inference engines are computer applications that infer the knowledge base to deduct answers from it. They are brains in any expert systems. Jena provides Jena2 inference subsystem that allows a range of inference engines and reasoners to be plugged into Jena. Such engines are

used to derive additional RDF assertions which are entitled from the base RDF together with any optional ontology information and axioms and rules associated with the reasoners (Reynolds, 2010). Jena provides inference through its extensions of inference API. The inference API provides reasoners which can be configured. The configuration of reasoners would instantiate them and the instances can be attached to RDF data to create inference models. These inference models access the RDF statements to deduct additional statements that are entailed from the bound data by means of the reasoners. Jena includes RDF reasoner to support all RDF entailments and the OWL reasoner to implement rule-based reasoning on OWL documents. Jena provides rule-based reasoned that implement both RDF and OWL reasoners. The ArcheaoKM platform uses jena rule, a rule based reasoner of Jena to execute semantic rules with ArchaeoKM. It however uses SWRL to implement spatial rules within research activities.

6.4. Discussion

The ArchaeoKM platform is a Web platform for facilitating archaeologists to manage their information. It is a collaborative tool where archaeologists can share their knowledge. The implication of knowledge is used in its full extent within the application through the concept of 4Ks. The ArchaeoKM platform being a Semantic Web application uses the Semantic Web tools and techniques within it. The knowledge technology which has become an important and stands out tool within the Semantic Web framework is implemented through these 4Ks. The collaborative knowledge sharing through the ArchaeoKM platform not only helps to manage the data within but also generates new knowledge for the archaeologists to evaluate.

The benefits of the emerging Semantic Web technology through its knowledge tools are quiet visible over the convention technologies that rely heavily on database systems. The ArchaeoKM platform has just laid a stepping stone to affirm the benefits. The shift in technology is quite eminent now and the benefits with the new technology are just speeding up the process. The benefits that have been experienced during the design and development of the ArchaeoKM platform are quite strong. The flexibility nature of ontology based system allows integrating new components at any time of development and even implementations. This is quite in contrast to the systems developed in database environment. Additionally, the acceptance of the application in very strict discipline of archaeology also provides a strong

argument for the continuation of usage of applications based on Semantic Web for wider audiences.

This research thesis aims at contributing the ongoing enhancement of the Semantic Web technologies through focusing on the possibility of integrating spatial components within its framework. It makes an attempt to cross the boundary of using semantics within the geo-ontology researches to provide interoperability and takes it a step forward in using the underlying knowledge technology to provide spatial analysis through knowledge. The direction that led the research centered on the same theme of integrating spatial components in knowledge technologies of semantic web framework. In the other hand it was necessary to develop a package that utilizes all the components of Semantic Web and provides at least the same degrees of performance that conventional technologies provide. The ArchaeoKM platform is primarily designed as an argument to the direction. It was also intended to include the spatial part within the application to provide even stronger base for argument. It however was not completely possible because of time constraints, yet the demonstration tool that was developed as a proof of concept demonstrated that the concept is applicable. Ironically, it also showed that the principle of integration could be applied in any discipline. Moreover, the top level ontology of the ArchaeoKM project could be base for other domain to manage the knowledge in a collaborative approach.

References

B.V., A. (2009). *User Guide for Sesame 2.3*. Retrieved November 15, 2010, from openRDF.org: home of Sesame : http://www.openrdf.org/

Dickinson, I. (2009, February 4). *Jena – A Semantic Web Framework for Java* . Retrieved November 12, 2010, from Jena Ontology API: http://jena.sourceforge.net/ontology/index.html

Garrett, J. J. (2005, February 18). *adaptive path*. Retrieved November 12, 2010, from Ajax: A New Approach to Web Applications: http://www.adaptivepath.com/ideas/essays/archives/000385.php

Google. (2010). *Google Web Toolkit*. Retrieved November 12, 2010, from Google code: http://code.google.com/Webtoolkit/overview.html

Knublauch, H. (2003, July 20). *JavaWorld: Solutions for Java Developers*. Retrieved November 15, 2010, from An AI tool for the real world: Knowledge modeling with Protégé: http://www.javaworld.com/javaworld/jw-06-2003/jw-0620-protege.html

Project, O. G. (2002). *Quantum GIS*. Retrieved November 23, 2010, from Quantum GIS: http://www.qgis.org/

Research, S. C. (2010). *User Documentation*. Retrieved November 15, 2010, from Protege: http://protege.stanford.edu/doc/users.html

Reynolds, D. (2010, March 28). *Jena 2 Inference support* . Retrieved November 15, 2010, from Jena – A Semantic Web Framework for Java : http://jena.sourceforge.net/inference/

Chapter 7

CONCLUSIONS

This chapter concludes the presentations of activities and results of the research. It discusses the relevance of the research working in the growing influence of the Semantic Web technologies in the domain on Information Technology. The chapter also presents the contribution of the research work in context of similar work in this area. Lastly, it concludes highlighting the future areas in the research that need to be carried out for providing strong base of implementation of spatial technology in the Semantic Web framework.

This research has made an attempt to contribute through including the functionalities of spatial analysis within the Semantic Web framework. Moving beyond the semantic information, it has opened the chapter of inclusion of other form of information. It is important in the sense of development of the technology itself. The world is witnessing a shift in technology and the Semantic Web is the direction the shift is moving towards. This would mean that the technology including that of GIS is moving towards the flexible solutions through knowledge based systems from static solution through current database systems. Hence, it is important to raise issues of integrating non-typical semantic data into it. This research thesis should at least provide certain vision towards the direction the technology should take to integrate these forms of data. It discusses the direction in terms of spatial integration. There are other data patterns like temporal data which need to be addressed too.

This concluding chapter begins with summarizing the work contribution that has been presented in previous chapters. It then discusses the scope of current research in respect to a research work that resembles closely and highlights the results of the current work against the previous one. Lastly, the chapter concludes the future prospect and the direction of the research work in this field.

7.1. Contribution

This research attempts to highlight the possibilities to integrate spatial technology in Semantic Web framework. It moves beyond the scope of data interoperability while presenting the concept and makes efforts to utilize the potentiality in other areas of the Semantic Web technologies. The underlying technologies of knowledge processing provide to the Semantic Web the capabilities to process the semantics of the information through close collaboration with the machine. It makes not only the understanding of data easier for achieving interoperability among different data sources, but it also provides valuable knowledge which could enrich the knowledge base in order to equip it with new knowledge. This helps the users understand the data better. The underlying knowledge technology makes stand out among its contemporaries.

7.1.1. In the Industrial Archaeology Domain

This research benefits from the advancement in Semantic Web technologies and its knowledge representation formalization tools and techniques. The primary principle of 4Ks processing is based on the knowledge formalization techniques. The research uses the case study of the industrial archaeology to demonstrate the possibility of implementation of application based on Semantic Web and utilizes the knowledge possessed by the archaeologists to manage the information excavated. This turns out to be an ideal case for the experimentation as the site for industrial archaeology is available for short duration of time. With the conventional technology it is difficult to manage the information due to share volume of data and the limitation of available time. It is however seen that with 4Ks implemented within the application prototype of the ArchaeoKM framework, the information could be managed. There has always been active involvement of archaeologists in every phase of design and development. The domain ontology and its axioms and theorems are based on their experiences. The enrichments of domain ontology through the identification of objects are carried out by them. It is the first K, Knowledge Acquisition. The knowledge acquired through the identification process is managed through defining relationships. It is again the archaeologists with the ArchaeoKM platform to manage knowledge through adjuring proper relationships (which reflects archaeologists view of the world) to the objects and semantically annotating to the data and documents collected during the excavation. The process is second K: Knowledge Management. The third K is Knowledge Visualization which generally means that knowledge identified and managed could be visualized through the interfaces of the ArchaeoKM platform. The knowledge base enriched and managed through the collaborative approach of archaeologists could be analyzed through inferring the knowledge base with rules formulated by archaeologists. These rules are inferred through SWRL – a rule language for Semantic Web standardized by W3C (Horrocks, Schneider, Boley, Tabelt, Grosof, & Dean, 2004). It is the last K, Knowledge Analysis.

7.1.2. In the Geospatial Domain

The 4K processing principle is implemented during the integration of spatial technology. The domain ontology is modified to adjust the spatial components into it. The research work considers the advancement in spatial technology in modern database systems. It implements the notations standardized by OGC simple feature specification (Herring, 2010) during the

inclusion of the spatial components as axioms into the ontology. This has been discussed in chapter 5. The spatial technologies provide spatial functions and operations to perform spatial analysis. These functions and operations are categorized into four major categories as mentioned in chapter 4. However, the research implements functions under geospatial processing and geospatial relationship functions as these two categories consist of mostly all the spatial functions. Geospatial processing functions are implemented as class axioms which relates to the classes containing features through the respective object properties. Likewise the geospatial relationship functions are treated as object properties relating the classes containing features spatially to each other.

The knowledge acquisition process comprises of acquiring spatial signatures of the object. In general they are acquired during the identification process. However, the spatial signatures are formalized during spatially annotations of the objects which are then stored in database as spatial data type. The spatial operations and functions which are encoded as classes and object properties within the ontology provide the management of spatial knowledge. The ontology was spatially enriched through the spatial operations and functions at the database level. This enriched knowledge base can be inferred spatially through the spatial built-ins for SWRL proposed in the research. The research also proposes the spatial built-ins for query language of the Semantic Web (SPARQL) (Harris & Seaborne, 2010).

The benefits to geospatial community are prominent. The shift from data oriented to knowledge oriented GIS gives the GIS an edge. The flexibility of knowledge based systems should add the flexibility to GIS in terms of data acquisition, data management and data analysis. The data acquisition process though remains the conventional digitization techniques; the possibility to link it up to its semantics adds knowledge to it. This added knowledge then could be utilized for different purposes including semantic interoperation between other data from other sources. However, here it is discussed in terms of knowledge management and analysis. The knowledge query through SPARQL or knowledge inference through SWRL to the spatially rich knowledge base generates new knowledge which is more authentic in a sense that this new result is the manipulation of knowledge base through the existing one. It is not just data any more. The semantic behind the results provides support to their authenticity.

This research has provided GIS community an alternative to conventional spatial data analysis through spatial rules. It can be opined that the proposed approach of knowledge analysis is apparent and less complicated to the conventional one. As the spatial rules could be combined with general rules they have wider implications. Additionally, the rules are based on formal logics which relate to day-to-day human interpretations; they should be easy to understand and implement. Consequently, the research proposes a rule based approach for spatial analysis and provides an evidence of possibilities through the experimentation performed.

7.1.3. In the Semantic Web Domain

A spatial layer in the Semantic Web stack presented in chapter 5 is not enough to address the overall problems of non-semantic data within the framework but at least there is something to start with. The full potential of underlying knowledge techniques through the reasoning or inferring capabilities within Semantic Web has not been identified in Geospatial community. The primary focus on these technologies is to achieve data interoperability within different data sources [(Cruz, Geospatial Data Integration, 2004), (Cruz, Sunna, & Chaudhry, Semi-Automatic Ontology Alignment for Geospatial Data Integration, 2004)]. Even W3C concentrated its priority in proposing comprehensive geospatial ontology acceptable to all through its Geospatial Incubator Group (Lieberman, Singh, & Goad, 2007). All these research works show that the emphasis on using geospatial ontology lie in achieving data interoperability and thus ignores the capabilities of underlying knowledge techniques for carrying out complex spatial analysis. This research presented a concept to carry out spatial analysis through inferring knowledge base spatially.

The realization of spatial integration into Semantic Web framework is demonstrated through a demonstration application. The application demonstrates that through a suitable translation engine, it is possible to infer the spatially enriched knowledge base in order to deduce spatial knowledge. The translation engine developed within the demonstration application translates the spatial built-ins and enriches the knowledge base through results of spatial operations of these built-ins making the knowledge base ready to be inferred. This has been discussed in chapter 6.

7.2. A very similar work

The work presented in the research work (Klien, 2008) resembles to the approaches embraced in this research work to a certain extent. However, the applications and the focuses of both researches vary entirely. The goals of research presented were entirely focused on providing solutions for information discovery through finding the relevant geo-data and to access them to retrieve the information in existing heterogeneous data sources. Nevertheless, the similarity exists in form of using the spatial rule to achieve the targets. The activities in (Klien, 2008) are mainly motivated by the heterogeneity problems in Spatial Data Infrastructure (SDIs). Spatial Data Infrastructures are basically frameworks of spatial data and represented through the metadata defining the data, their nature and in some degrees their intent uses and users or the tools to use them. The research concentrates the accessibility of spatial information through Web services in Web Feature Services (WFS) and the SDIs attached to such services.

The research project (Klien, 2008) in certain degrees follows the pattern of existing studies in geo-ontology research domain by focusing on the use of ontology for achieving data interoperability. It follows this pattern of research through defining the problems of data discovery in WFS and conceptualizing a mechanism to understand the semantic differences in the data even though the features associated to the data have the same naming conventions. The problems follow even after the data discovery due to nature of the information that data represents is not explicitly stored. The research hence plans a mechanism of match making of different SDIs through the mediation of semantic rich domain ontology designed through the consultation of the experts. The Domain Ontology as it terms contains explicit information which capture the meanings of real world entities.

The users are required to query the WFS for the data in the conventional way which transforms the feature type's application schema of the SDI of the WFS into feature type ontology. This ontology does not possess explicit information and hence only reflects the data schema of the features. Mapping of concepts through the match making mechanism between domain ontology and feature type's ontology provides explicit semantic information to the concepts. This mapping is done initially through the string base matching but to verify semantically that both the concepts are dealing with same world entity, spatial analysis method is put forward in the extraction process. It is here the approach between two researches converge. As with the case of this research, the research (Klien, 2008) utilizes the

inference capabilities of the description logics in the ontology representation language of OWL DL through inference rules. It (Klien, 2008) uses a simple hydrological example to semantically annotate the data through the spatial rules. The SWRL representations of the rule are given:

```
Region(?x) ^ hasSlope(?x, Flat) → Lowland(?x)                    (1)
```

```
Lowland(?x) ^ River(?river) ^ adjacentTo(?x, ?river) ^           (2)
hasAltitude(?x, ?xAlt) ^ hasAltitude(?river, ?riverAlt) ^
swrlb:subtract(?diffAlt, ?xAlt, ?riverAlt) ^
swrlb:lessThan(4, ?diffAlt) → Floodplain(?x)
```

Region, Lowland, River and Floodplain are the concepts and `hasSlope`, `adjacentTo` and `hasAltitude` are the object properties in both feature type's ontology and domain ontology. The idea is to semantically annotate the concept Floodplain with the rules. The first rule represented by equation 1 forms the lowland if the slope of a region is flat. There are many constraints of a region being lowland but the research uses this rule to demonstrate the usability. A Digital Elevation Model (DEM) is used in background which intersects the inferred information and the dataset is annotated as lowland. In short the object property `hasSlope` is intercepted and run through an algorithm which combines the DEM dataset to determine the flat slope. `Regions` inferring these flat slops are then annotated as lowland. Extending the rule to equation 2, it uses object property `adjacentTo` and built-ins of SWRL to annotate the `floodplain`. The object property `adjacentTo` again needs to run an algorithm in collaboration to the spatial dataset to provide the result. This result again infers with the other axioms in the knowledge base to enrich itself. The `adjacentTo` object property utilizes buffer operation to determine the objects close to it. However, the operation is hidden from the users and is executed inside the algorithm. This execution enriches the knowledge base which could be inferred through standard rule of SWRL. The execution of buffer or any spatial operations are carried out through the spatial operations of ArcGIS. The semantic annotation through these rules is carried out to enrich the Domain Ontology thus negating any short coming of explicit semantics in feature type's ontology.

The method of inferring the rules first through execution of spatial operations at database or application level and then enriching the knowledge base matches with the current research

work. However, the implications in both researches are different. The approach that current research undertakes is to enhance the Semantic Web technologies through integrating spatial components into the technology. It differs significantly with the former research (Klien, 2008) as it was conducted to use Semantic Web tools and techniques to answer specific GIS problems. Hence, the scale of application of the Semantic Web techniques is relatively low in the previous research. In other hand, it could be seen that the spatial operations and functions are used implicitly through object properties like `hasSlope` or `adjacentTo` which are terms of natural language. This might give ambiguity to the interpretations of these terms. For example the term `adjacentTo` can have two or more meanings as rightly quoted in the thesis report. It can be near to each other either through touching or not touching. So, the utilization of spatial operation should be based on these factors. If the adjacent to means that the objects are touching then the spatial operation `Touches` could be directly used instead of `Buffer` which is more resource dependent.

The thesis provides the example of `Floodplain` to demonstrate the applicability. The terms which are used in example are commonly used terms to explain the relation. This however also could provide short coming in bringing a larger community into same understanding as people prefer to use different terms to interpret same thing. It is also not possible to anticipate all the terms that could form a spatial relation and enrich them in the ontology as there could be infinite terms. The current research presented in this report has taken the works forward to address these concerns. Instead of using the commonly used terms, it uses the spatial operations and functions terminology standardized by OGC. It has proposed standard terms to formulate rules rather than using different terms.

The equation 3 illustrates the adjustment of object property `adjacentTo` directly through SWRL rules through spatial built-ins.

    River(?x) ^ Lowland(?y) ^ Buffer(?x, ?y, 50) →adjacentTo(?x, ?y)

                                                                        (3)

It could thus be seen that there is much more flexibility of implementation of spatial rules through standard spatial built-ins proposed here. Besides the spatial built-ins for SWRL, this research adds on spatial built-ins to SPARQL, the query language of Semantic Web tools which is not explicitly researched before.

7.3.   Way Forward

This research thesis has highlighted the benefits of tools and techniques of the Semantic Web and especially underlying knowledge technologies within spatial technologies for the efficient management of spatial information. It has also been discussed that the approach presented here benefits both the Semantic Web and spatial technology. It should also be noted that the research activities has just initiated the integration of spatial technology into the Semantic Web framework and still has long way to go to have complete integration. This section presents few areas where the research work could be continued in this area.

Researches in the field of spatial technology within the Semantic Web framework have not moved beyond geospatial ontology and the possibility of semantic interoperability between different sources. This research attempts to break that trend and used knowledge to manage the spatial data through knowledge. In the process, it provided the mechanisms to infer spatial rules through spatial built-ins for SWRL. This was done first through populating domain ontology with the spatial components so that spatial knowledge could be enriched into it and this spatially rich knowledge inferred through SWRL. It could also be queried through SPARQL. However there are number of issues that need to be addressed in future work. The first one is about the dependability on the database systems to conduct the spatial operations and functions. This research uses the spatial operations and functions provided by PostGIS, the spatial extension of PostgreSQL to enrich the knowledge base through their result. Future works should make an attempt to free them with such dependency through providing such functionalities within spatial built-ins themselves.

Another area where the research could concentrate is the area of using current reasoning engines to reason the spatial knowledge base and deduce the implicit spatial knowledge. In other words instead of using the inference engine to infer the rules through SWRL, the constraint axioms should be introduced within the ontology which automatize the enrichment of knowledge base through reasoning mechanism. The constraint axioms in particular should be able to include the spatial built-ins and run through the respective spatial operations and functions to automatize the enrichment process while reasoning the knowledge base. It can be clarified with one of the typical examples in industrial archaeology: *"chimney should be 5 meters around an oven and round"*. Currently it is possible to execute this through SWRL rule.

```
feat:Object(?x)  ^ feat:Oven(?y)  ^  spatialswrlb:Buffer(?y, 5, ?x)
^ att:hasShape(?x, round) → feat:Chimney(?x)
```

This infers the spatial knowledge base to annotate the result to the class `feat:Chimney`. However an alternative of using the restriction of

$$feat:Chimney \sqsubseteq Buffer\ feat:Oven, 5\ \sqcap\ \exists hasShape.\{round\}$$

can be thought upon. The existing reasoning engine then reasons every object with round shape around 5 meters of every oven and terms them as individuals of chimney.

It is important to have standard terms for every built-in that will be developed to process spatial knowledge. With other built-ins in the tools standardized by W3C, the spatial built-ins should also get standardized by the consortium. In addition to W3C, OGC should also get involved in standardizing the built-ins. An effort in this direction should be carried out.

References

Cruz, I. F. (2004). *Geospatial Data Integration.* Chicago: ADVIS Lab, Dept. of Computer Science, University of Illinois.

Cruz, I. F., Sunna, W., & Chaudhry, A. (2004). Semi-Automatic Ontology Alignment for Geospatial Data Integration. *GIScience 2004* (pp. 51 - 66). MaryLand: LNCS.

Harris, S., & Seaborne, A. (2010, January 26). *SPARQL Query Language 1.1*. Retrieved September 2, 2010, from W3C Working Draft: http://www.w3.org/TR/2010/WD-sparql11-query-20100126/

Herring, J. R. (2010). *OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.* Open Geospatial Consortium Inc.

Horrocks, I., Schneider, P. f., Boley, H., Tabelt, S., Grosof, B., & Dean, M. (2004, December 21). *SWRL - A Semantic Web Rule Language - Combining OWL and RuleML*. Retrieved May 22, 2009, from SWRL: A Semantic Web Rule Language - Combining OWL and RuleML: http://www.w3.org/Submission/SWRL/

Klien, E. M. (2008). *Semantic Annotation of Geographic Information.* Essen: University of Muenster.

Lieberman, J., Singh, R., & Goad, C. (2007). *W3C Geospatial Ontologies - W3C Incubator Report.* W3C Incubator Group.

PUBLICATIONS

KARMACHARYA A., Proceeding of Wissenmanagement für räumliche Daten – Fallstudie mit industrearchäologischen Befunden, Denkmäaler3.de – Industriearchäologie, Essen, Germany 5-7 November 2008, Pages 71 – 82

KARMACHARYA A., CRUZ Ch., MARZANI F., BOOCHS F. (2008), Industrial Archaeology: Case study of Knowledge Management for Spatial Data of Findings, Adaptive Hypermedia, Workshop 4 Proceedings, Hannover, July 2008, Pages 35 – 39

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2008), Managing Knowledge for Spatial Data – A Case Study with Industrial Archaeological Findings, Proceedings of the Int. Conference "Digital Heritage in the new knowledge environment: shared spaces & open paths to cultural content", Athens, November 2008, Pages 150 – 153

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2009), ArchaeoKM: Toward a Better Archaeological Datasets Management. "Proc. CAA09", Williamsburg, Virginia, March 2009

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2009), Support of Spatial Analysis through Knowledgebase – A new concept to exploit spatial information shown for Industrial Archaeology, International Cartography Conference, Santiago, Chile, November 15 – 21

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2009), ArchaeoKM: Managing Data through knowledge in Industrial Archaeological Sites, The International Committee for the Conservation of the Industrial Heritage, Freiberg, Germany, August 30 – September 5 2009

KARMACHARYA A.; CRUZ Ch.; BOOCHS F.; MARZANI F. (2010), ArchaeoKM: Realizing Knowledge of the Archaeologists, Proceeding: Von Handaufmaß bis Hightech III, Cottbus, Germany, 24 – 27 February 2010, Philipp von Zabern, pp 102 - 108

KARMACHARYA A.; CRUZ Ch.; BOOCHS F.; MARZANI F. (2010), Integration of Spatial Technologies and Semantic Web Technologies for Industrial Archaeology, 6th International Conference on Web Information Systems and Technologies, Valencia, Spain, 7 – 10 April 2010

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2010), ArchaeoKM: A tool to Manage Knowledge from Archaeological Data. "CAA10", Granada, Spain, April 2010

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2010), Use of Geospatial Analyses for Semantic Reasoning, 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems KES 2010, Cardiff, Wales, UK, September 8 – 10, 2010, LNCS Volume 6276/2010, LNAI 6276, pp 576-586,.

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2010), Spatial Rules Through Spatial Rule Built-Ins in SWRL, Journal of Global Research in Computer Science, Vol. I No. II, ISSN-2229-371X

KARMACHARYA A., CRUZ Ch., BOOCHS F., MARZANI F. (2011), Integration of Spatial Processing and Knowledge Processing through the Semantic Web Stack , 4th International Conference on GeoSpatial Semantics, Eds. Christophe Claramunt, Sergei Levashkin, Michela Bertlotto, Brest, France, Springer LNCS 6631 pp 200-216