Mémoire de Thèse

Présenté pour obtenir

Le grade de Docteur en Sciences - Mention Informatique

Par

Helmi Ben Hmida

From Quantitative Spatial Operators to Qualitative Spatial relationships - A new approach applied to the detection and the semantic qualification of 3D objects

Le 12 Décembre 2012

Composition du jury :

M. Frank Boochs, Professeur, Université de Mayence, **Co-Directeur de thèse**

M. Klaus Böhm, Professeur , Université de Mayence, **Examinateur**

M. André Borrmann, Professeur, Université de Munich, **Rapporteur**

M. Christophe Cruz, Maître de conférences, Université de Bourgogne, **Co-Encadrant de thèse**

M. Gilles Gesquière, Professeur, Université de Lyon 2, **Rapporteur**

M. Joerg Klonowski, Professeur, Université de Mayence, **Examinateur**

M. Christophe Nicolle, Professeur, Université de Bourgogne, **Directeur de thèse**

M. Andreas Nüchter, Professeur, Université de Jacobs- Allemagne, **Examinateur**

Doctorat préparé au sein de l'équipe Systèmes d'Information et Systèmes d'Images, dans l'équipe projet CHECKSEM du laboratoire LE2I de l'Université de Bourgogne.

Acknowledgements

The achievement of this thesis has offered me a wonderful opportunity to meet and exchange ideas with many people. This work could not have been realized without the intervention of a number of persons and their precious guidance and assistance.

First I would like to express my thanks to the Jury members who accepted to assess my thesis despite their occupations. Special thanks to the reviewers Prof Gill Gesquière and Prof André Borrmann who have used their precious time to evaluate the work and give their vital suggestions. Thanks also for their interest to our contribution and mainly for preparing the bases through their research on which this theses was built.

I would like also to especially acknowledge Prof Christophe Nicolle for his acceptance to supervise this thesis during the last three years. Likewise, I would like congratulate him for his focused vision and to think him for educating me how to think in a scientific manner, and how to shape scientific research along different steps.

I am also very grateful to Prof Frank Boochs, the leader of the i3mainz institute for giving me the opportunity to carry out my thesis in the best conditions, in parallel with my work in the institute.

I address my sincere thanks to HDR DR Christophe Cruz for his supervision and support which was necessary to carry out this work. He was always there to support me in dark moments. Without his faith, words of encouragement and mainly his support, I would not have been able to succeed. I will always be thankful to him and never forgot his care.

This thesis would have been very hard without the support of the German Ministry of Research and Education under contract No. 1758X09.

I would also like to acknowledge all the staff of the i3mainz and the Le2i, especially Dr Adlane Habed, Andreas Marbs, Ashish karmacharya and Hung Truong.

To finish, I want to thank all my family. I must first thank my Father for sacrificing everything to bring me to where I am. Likewise for my mother, sister, brother and my friend Dr Yahia for their words and love. I don’t know how to thank my wife, who always stood by my side and supported me during several critical months of the thesis. She always believed that I could do it, even when I doubted myself. Finally, during the course of time, we were given a very beautiful gift, our daughter. I will always be grateful to my wife for sacrificing lots of time from her thesis to take care of her.

Thanks to all of you…

Abstract

This work presents the **3D S**patial **Q**ualification tool (3DSQ) which was created to compute spatial data stored in OWL-DL ontology. By using the adjustment principle of an existing ontology, it is then possible to add 3D data to existing objects and compute their spatial relationships from their 3D models. The 3DSQ Platform makes an attempt to ensure the interaction between heterogeneous environments. Actually, such a semantic platform connects an adjusted OWL ontology structure, a 3D quantification engine, a visualization engine and a set of geometry via knowledge processing technology materialized via SWRL, SQWRL rules and SPARQL queries within its extended Built-Ins. The created Spatial Built-Ins are connected to the presented quantification engine and enable qualifying semantic spatial relationships. This will mainly help us to not just apply semantic queries selecting geometry based on such a qualified relationship, but also to benefit from the richness of the knowledge based schema, from a logical point of view. It includes the semantic definition and the implementation of the standard 3D spatial relationships and uses sophisticated geometry data structure like NEF Polyhedra. It further describes the implementation of the suggested bridge by the means of the NEF Polyhedra operation and the DLs definition of spatial relation.

In addition, this thesis presents an application of the 3DSQ platform. It is argued that the representation of spatial information is not a fundamental limitation of OWL, where linking top level semantic qualification with low level quantitative calculation is highly possible and efficient via the OWL-DL expressive power. This efficiency is carried out by the semantic rule system, and the geometry data structure required for the representation of spatial regions. In fact, such a semantic qualification based on description logic (DLs), and OWL ontologies enable much more efficient and intelligent spatial analysis semantically. To prove the feasibility and to validate the 3DSQ Platform within its quantitative and qualitative 3D spatial operators, real applied areas related to Building Information Model (BIM), IFC and especially 3D point clouds data were addressed. Given the complexity of the underlying problems, the suggested new methods resort to using semantic knowledge, in particular, to support the object detection and qualification. In this context, a novel approach which makes use of the 3DSQ platform and benefits from intelligent knowledge management strategies to qualify objects will be

discussed. It is based on the semantics of different associated domains to assist in knowledge formalization where Knowledge helps in the qualification process, and can be clearly palpable through the thesis.

Such a conception will bring solutions to the problem raised by the syntactic exchange level between CAD software packages, IFCs or 3D point cloud geometries. Moreover, all relations between the different geometries are defined by elements suggested in this thesis. In fact, these relations define how elements can interact. Such a semantic can only be synthetized, used and invested by OWL ontology structure with all the robustness of the Description Logics.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation, problematic and contribution

1.1.1 Motivation

Over the last few years, formal ontologies (Antoniou & Harmelen, 2009) have been suggested as a solution for several engineering problems, since they can efficiently replace standard data bases and relational ones with more flexibility and reliability. In fact, ontologies present a formal representation of knowledge by a set of concepts within a domain, and the relationships between those concepts. Well-designed ontologies possess lots of positive aspects, like those related to defining controlled vocabulary of terms, inheriting and extending existing terms, declaring a relationship between terms, and inferring relationships by reasoning on existent ones. Ontologies are used to formally represent the knowledge of a domain, where the basic idea was to present knowledge using graphs and logical structure to make computers able to understand and process it (Ben Hmida et al., 2011). The basic strength of formal ontology is their ability to reason in a logical way, based on Description Logics (DL) concepts (Baader, 2009) where a lots of reasoners exist nowadays like Pellet (Sirin, et al., 2007), and KAON (U. Hustadt, 2010). Despite the richness of the OWL set of relational properties, the axioms do not cover the full range of expressive possibilities for object relationships that we might find, since it is useful to declare relationships in term of conditions or even rules. These rules are used through different rule languages to enhance the knowledge possess in an ontology. In the next section, we will clarify the target engineering problem to be dealt with and resolved, with the help of the above introduced knowledge base technology.

1.1.2 Problematic

Recently, building modelling process, known as Building Information Modelling (BIM) (Eastman, et al., 2008), has come to occupy a wide area within the Architects, Engineers and Contractors (AEC) domain (Gu & London, 2010). Such an evolution has seen the light along with the normalisation of the BIM domain through new standards, mainly the Industry Foundation Classes (IFC) (Qin, et al., 2011) on one side, and through new efficient platforms managing such standards on the other. The presented format was introduced by the International Alliance for Interoperability (IAI) and considers the building elements as independent objects where each object is characterized by a 3D representation and defined by a semantic normalized label. Consequently, architects and experts are not the only ones who are able to recognize the elements, but everyone will be able to do it, even the system itself. For instance, an IFC door is not just a simple collection of lines and geometric primitives recognized as a door; it is an "intelligent" object door which has a door attribute linked to a geometrical definition. "*IFC files are made of objects and connections between these objects, where the object attributes describe its business semantic*" (Vanland, Cruz, & Nicolle, 2008). Relations between the different defined building elements are represented by "relation elements". Subsequently, building geometries are not modelled explicitly by means of a boundary (Haimes & Dannenhoffer, 2010) or CSG representation (Dempsey, 2010), but implicitly by using attributes with a geometric meaning.

In fact, one of the most important components in the BIM model is the Spatial Relation. It presents an important actor within the AEC domain, where the spatial relation and characteristics of geometries are able to characterise the building model semantically. Until today, existing technologies still suffer from the disability to interpret the geometric information presented within the building model, mainly those in concordance with the different relations between the presented building geometries. In fact, existing product model servers (Adachi, 2003) are restricted to the numerical evaluations of the spatial relations already predefined within the product model. Such a handicap will mainly reduce its expressivity. The lack of platform supporting spatial analysis added to the geometric one can be enlightened by the fact that research has mainly concentrated on bringing solutions for semantic object building modelling or building information model (BIM) (Schlueter & Thesseling, 2009) without thinking about integrating the spatial component analysis within the BIM model.

Not far from the Building Information Modelling, the technical survey of facility aims to build a digital model based on geometric analysis from different data sources. Such a process is becoming more and more tedious, especially in the case of terrestrial laser scanners as main data sources where a huge amount of 3D point clouds are generated. Within such a scenario, new challenges have seen the light, where the basic one is to make the survey process automatic and more accurate. Thus, early works on 3D point clouds have investigated the reconstruction and the recognition of geometrical shapes (Pu & Vosselman, 2007) to resolve this challenge. Unfortunately, most of these approaches are data-driven and concentrate on specific features of the objects being accessible to numerical models. These problems can be solved when further supplementary and guiding information is integrated into the process chain for object detection and recognition through its geometric and spatial characteristics, enabling to support the validation process. Such information can be derived from the context of the object itself and its behaviour, with respect to the data and/or other objects, or from a systematic characterization of the parameterization and effectiveness of the process to be used. However such a domain is characterized by specific vocabulary containing different type of objects. In fact, the assumption that knowledge will help the improvement of automation, accuracy and the quality result is shared by specialists of the facility model creation from point cloud processing.

1.1.3 Contribution

The main above discussed issues will be looked at during this paper, where we will highlight the current challenge in the field of 3D geometric spatial relations and the impact of semantics on it (Ben Hmida et al., 2012). In fact, based on the different observations, we predict that more standard and flexible representations of facility objects and more sophisticated guidance for object model creation, by modelling the geometric and spatial knowledge within an ontology structure, will open the way to significant improvement in facility modelling capability and generality, since it will enable us to create a more dynamic process based on object characteristics and make the qualification process more robust.

Within the actual research, domain ontologies are used to define the concepts, and the related necessary and sufficient conditions. These conditions are of value, because they are used to populate new ones. In addition, the rules are used to compute more complex results such as the 3D spatial relationships between objects. For instance, the relations between objects are used to obtain new efficient knowledge about it. To do so, and in order to reduce this technological gap between both domains, we develop throughout this thesis concepts and techniques for 3D spatial information and query language for Building Information Models that we will call the “3D Spatial Qualification” approach (3DSQ). It makes qualifying specific building component relations possible, by means of spatial semantic constraints. Such an idea can be used in lots of applied areas and applications for Building Information Models range, from verifying construction rules to extracting partial models that fulfil particular spatial constraints. Likewise, such an initiative will be involved in different applied areas, for example, the detection and qualification of objects in 3D point clouds Data. Once achieved, this thesis takes a second step forward from the spatial qualitative geometric relation qualification, to the geometry qualification and its management from the 3D point clouds data. As a matter of fact, the second goal of our paper is to develop efficient and intelligent methods for automated object qualification. The principle of our solution is a knowledge-based detection and qualification of objects in point clouds for AEC (Architecture, Engineering and Construction) engineering. In contrast with existing approaches, our approach consists in using prior knowledge about the context and the object itself. This knowledge is extracted from databases, CAD plans, Geographic Information Systems (GIS), technical reports or domain experts. Therefore, this knowledge is the basis for a selective knowledge-oriented detection and qualification of objects in several data sources. Particularly, the presented research and contribution will be applied within the WiDOP project, (Ben Hmida, 2010).

1.1.4 WiDOP project

The WiDOP project (Knowledge-based detection of objects in 3D point clouds for engineering applications) presents a new research project founded by the German government. This project, as its name suggests, is about the knowledge integration for 3D point clouds processing, object detection and scene reconstruction. The created prototype behind this is based on semantic web technology and 3D processing algorithms. The created application will be able to facilitate the reconstruction process, and make it easy and mainly automatic. The mentioned created application aims at replacing the engineer's efforts, by managing the 3D processing algorithms and the engineer's knowledge automatically. The German Railway company and the Frankfurt airport, as the main partners, are the main associates for the project. In fact, the Fraport company's main concerns are the building and furniture management of the airport. The position of the furniture, relative to the security gates and the trash, is constantly moving. In addition, updates are done on buildings such as new walls, destruction of walls, new holes in a wall, new windows, etc. This could be undertaken by technical employees, in order to reorganize storerooms for instance. In fact, it is very difficult to keep the plans of the airport up to date. On the other hand, the main concern of the German railway company is the management of railway furniture. The issue is close to the Fraport one, because they have to handle the management of the furniture, which is constantly changing. Moreover, the cost of keeping these plans up to date is increasing. The key solution consists in fixing a 3D terrestrial laser scanner on a locomotive, and monitoring the surrounding landscape. After the first monitoring, the resulting data will be considered as a reference for comparisons with future monitoring in order to detect changes.

1.2 Suggested approach

Qualitative spatial relationships are used in many areas of computer science. Indeed, reasoning on such relationships is fundamental, so as to infer graphical depiction through logic mechanisms. In addition, these relationships facilitate the access to data by a query processing mechanism that refers to objects and their relationships.

1.2.1 Integration of 3D Spatial Processing with knowledge processing

Our current efforts centre on suggesting a complete solution for engineering building modelling implementation in the easiest manner, thus enabling the combination of geometric analysis and the spatial one in a more qualitative manner, separating the real quantitative knowledge from the qualitative one. Recently, the qualitative spatial relations have been used to carry out inference, and to identify inconsistencies on these relations. In our current work, we will focus more on the $\mathbb{R}^3$dimension environment. In addition, the 3D spatial relation computation will be carried out by external libraries, which makes the execution process more optimal, and where the standard semantic platform will be extended with new 3D spatial relation built-ins.

To fill this technological gap, we propose a 3D Spatial Qualification approach (3DSQ) throughout this paper. The model, architecture and the language of 3DSQ is based on semantic technologies, and was designed with a generic query language that is applied for Building Information Models. The 3DSQ tool enables to select specific building components by means of qualitative spatial constraints. These constraints form an intermediate level of abstraction between the technical views on building geometry, using specific geometries structure coordinates; the way human's reason about buildings and the relations between their components, Figure 1-1.

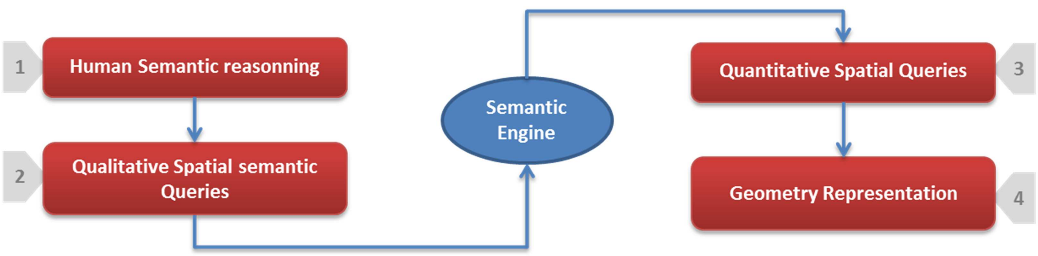


Figure 1-1. Qualitative relation provided by the quantitative one

The spatial operators available for the spatial types are the most important part of the algebra. They consist of

- Metric operators (distance, closerThan, fartherThan, etc.),
- Directional operators (above, below, northOf, etc.) and
- Topological operators (touch, within, contains, etc.),

1.2.1.1 *3DSQ platform- Demonstration through 3D CAD/IFC geometries*

Via the suggested semantic qualification of spatial relation in 3D data, the method makes it possible to build a semantic global diagram in an OWL ontology structure. First, such a CAD element can be presented as an IFC containing the scene geometries. Once converted to OWL Ontology structure, the executed process unifies all knowledge generated during each step of the building's management, beginning from geometries arriving to spatial relation, and finally to geometry qualification. The result is a rich semantic graph which contains geometries, spatial relation and element´s semantics. Such a concept will bring solutions to the problem raised by the syntactic exchange level between CAD software packages or IFC ones. Moreover, all relations between the different geometries are defined by elements suggested in this thesis. For instance, 'contain' defines a relation between a wall element and a window. Consequently, only elements having windows inside can be qualified as walls. Such a semantic can only be synthetized, used, and invested by OWL ontology structure with all the robustness of the Description Logics. In fact, OWL concepts define the semantic of elements, their relations and resources. Thanks to the presented extension via Spatial SWRL Built-Ins, new semantic relations can be easily added to the building management system materialized with the ontology structure. To do so, the suggested approach aims to use the original 9 Intersection model for the 3D topologic relation qualification instead of the dimensionally extended version, since we don´t see any need to use the last one to bring a solution to the presented problematic. Moreover, such a DEM-9IM has to be also supported by the suggested solution. The presented 3DSQ platform has to take into account the geometry structure and its ability to specify the internal, external and the boundaries of each one of the geometries. Once done, it has to suggest formal logic expressions able to satisfy the intersection model in each case, to be mapped later on to the semantic level. Likewise, in this paper, the semantic qualification will be linked to the quantitative one, where no further complex modification on the Standard $\mathcal{SHIQ}$ language (Horrocks, et al., 2003), and neither in any reasoners, will be achieved. On the other hand, we aim to avoid complex computation while qualifying the spatial relation based on DLs language. Finally, it is highly recommended that such a solution separates the low level quantification from the high level qualification, while always with ensuring a communication bridge between both of them. We follow a totally different approach in this paper, since we are based on semantic spatial, which is dynamically qualified via our knowledge base and the presented 3DSQ engine.

1.2.1.2 *Semantic technology and 3DSQ*

The presented concept requires efficient methods of knowledge, handling the different geometric, spatial and algorithms. Efficient knowledge-handling tools are available from the Semantic Web framework, which expresses knowledge through the Web Ontology Language (OWL). The encapsulation of semantics within OWL through Description Logics (DLs) axioms has made it an ideal technology for defining knowledge from almost any discipline. We use the OWL to define expert knowledge about the scene of interest, its geometry and spatial relation. With OWL ontology, we are able to describe complex semantics of a scene. For instance, the statement “A railway track is a linear feature with two linear structures running parallel to each other within a certain distance” can be expressed through logical statements. Likewise, in the case of the 3DSQ extension for the WiDOP project, we define the semantics of algorithmic processing within OWL. For example, the “Check parallel lines” algorithm is designed for detecting a “Signal,” which may contain parallel linear structures. As additional technology, the Semantic Web Rule Language (SWRL) is available. It is a program which infers logic from the knowledge base, to derive a conclusion based on the observations and hypothesis. For instance, the following rule asserts that a 3D geometry that has a distance from Distant_Signal of 1000m, has a height equal to or greater than 4m, and that has a linear structure, will be inferred as a Main_Signal, where Main_Signal and Distant_Signal are semantic objects within the railway scene. Most importantly, SWRL built-ins are keys for any external integration. They help in the interoperation of SWRL with other formalisms and provide an extensible infrastructure for knowledge based applications. They are essential in that they allow entry to a different world of processing. In the context of this solution, it bridges knowledge management and geometry processing.

1.2.2 Created prototype

The created 3DSQ prototype takes into consideration the adjustment of the old methods and, in the meantime, benefits from the advantages of the emerging cutting-edge technology. From a main point of view, the developed system still retains the storing mechanism within the existent 3D geometry processing. In addition, it suggests a new field of qualification, where we get a real-time support from the created knowledge. Added to that, we suggest a collaborative Java Platform based on semantic web technology (OWL, RDF, and SWRL) and knowledge engineering, in order to handle the information provided from the knowledge base and 3D geometries. The process enriches and populates the ontology with new individuals and relationships between them. In addition, the created platform offers the opportunity to materialize the qualification process by the generation and the visualization of the qualified geometries based on a VRML structure, (W3C, 1995), powered from the knowledge base. It ensures an interactive visualization of the resulting qualified elements beginning from the initial state, to a set of intermediate states arriving finally at an ending state, once the set of rules are totally executed. The resulting ontology contains enough knowledge to feed a GIS system, and to generate IFC file (Vanland, et al., 2008) for CAD software, Figure 1-2.

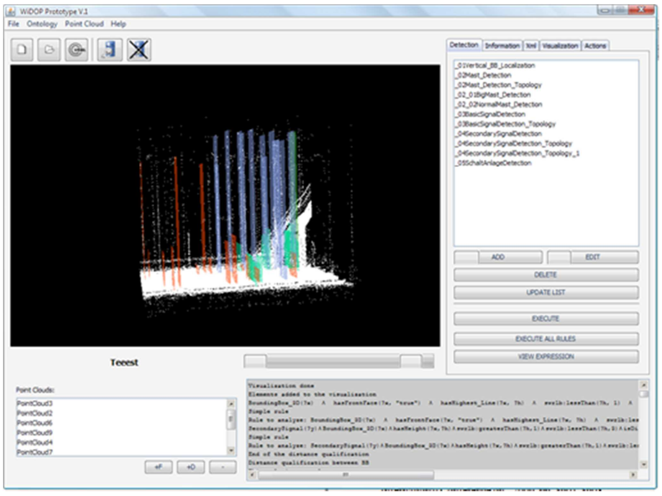


Figure 1-2. 3DSQ platform overview

1.3 Thesis Plan

This thesis puts forward the views, contribution and results of the research activities within the backdrop of Semantic Web technology and the knowledge management aspect within it. The suggested system is materialized via 3DSQ platform and its extension (Ben Hmida, et al., 2011) and applied in the context of the WiDOP project. Furthermore, the created platform is able to generate an indexed scene from unorganized geometries visualized within the virtual reality modelling language (W3C, 1995).

The paper is structured into different chapters. Chapter 2 presents the basic material for our contribution and gives an overview of the semantic web and the related technology: mainly, the knowledge engineering domain, the Description logic and its impact on the knowledge modelling. Second, more precise background knowledge on ontology web language and the Semantic Web rule and their relation to this thesis will be presented. In chapter 3, we present the first step, where we highlight the state of the art related to the 3D Quantitative spatial relation qualification, and suggest a more optimal and accurate 3D geometry structure for the qualification process. In chap 4, we discuss the suitability of Semantic web concepts and their related technology to the problem of 3D spatial relation qualification, and present the integration process of 3D Spatial processing with knowledge processing through the created 3DSQ, linking the qualitative spatial relation to the qualitative one, Figure 1-3 top. Chapter 5 demonstrates the impact of the presented semantic platform on the CAD/IFC geometry qualification, through a first semantic extension of the 3DSQ platform, Figure 1-3 Middle. Chapter 6 implements the second extension of the created semantic approach through its application to the problematic of 3D object detection and qualification in 3D point clouds data. Meanwhile, it presents the second semantic extension of the 3DSQ platform to support the new domain requirement, Figure 1-3 below.

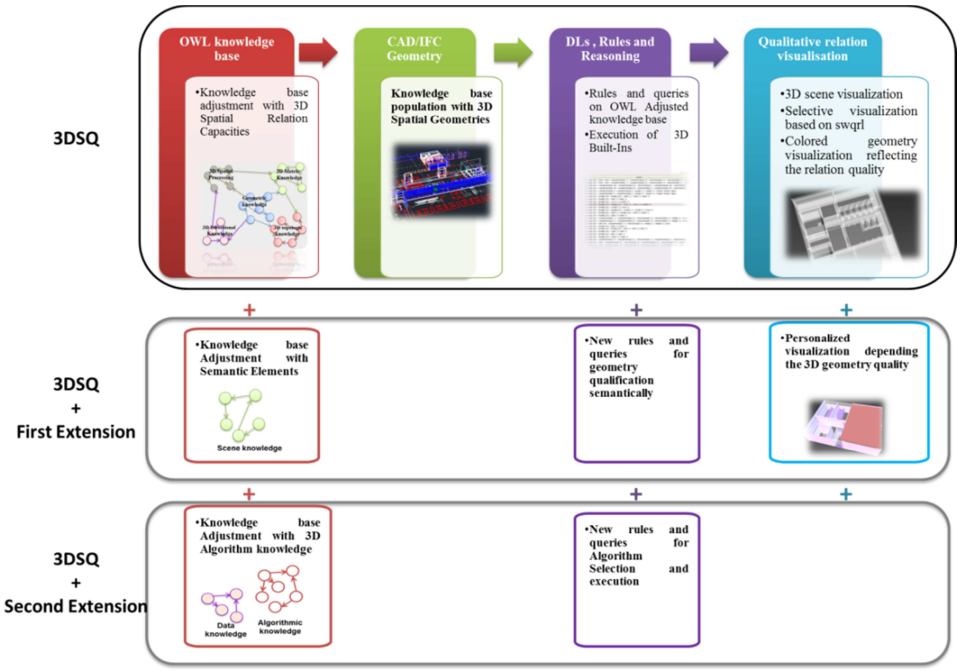


Figure 1-3. 3DSQ platform evolution

1.4 Conclusion

The present paper aims at building a bridge between semantic modelling and 3D geometric processing. The knowledge will be structured in ontologies structure, containing a variety of elements, such as already existing information about objects of the scene, for example data sources, information about the objects' characteristics, hierarchy of the sub-elements, geometrical spatial, etc. Throughout this paper, an approach on achieving object detection and qualification within these inference engines will be presented. The major context behind the current chapter is the use of knowledge in order to manage the engineering problem in question, based on heterogeneous environment. It primarily focuses on 3D geometry and its management through the available processing technologies incorporated through the knowledge. As the Web technologies mature through their approach in the Semantic Web, the implementation of knowledge in this domain seems even more appropriate.

References

Adachi, Y., 2003. Overview of partial model query language. In Proc. of the 10th Int. Conf. on Concurrent Engineering. pp.549--555.

Antoniou, G. & Harmelen, F., 2009. Web ontology language: OWL. Handbook on ontologies, pp.91--110.

Baader, F., 2009. Description logics. Reasoning Web. Semantic Technologies for Information Systems, pp.1--39.

Boochs, F., Marbs, A., Ben Hmida, H., Truong, H., Karmachaiya, A., Cruz, C., Habed, A., Nicolle, C., Voisin, Y., 2011. Integration of knowledge to support automatic object reconstruction from images and 3D data. In International Multi-Conference on Systems, Signals & Devices. Sousse Tunisia, 2011.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2012. From Quantitative Spatial Operator to Qualitative Spatial Relation Using Constructive Solid Geometry, Logic Rules and Optimized 9-IM Model. Proceeding ieee international conference on computer sciences and automation engeneering, Zhanjiajie China(453), p.2.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2011. From 3D point clouds to semantic object. In KEOD, the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. Paris, 2011.

Dempsey, C., 2010. What is GIS? [Online] Available at: http://gislounge.com/what-is-gis/ [Accessed 2010 August 2010].

Eastman, C.M., Teicholz, P., Sacks, R., Liston, K., 2008. BIM handbook. Wiley Online Library.

Gu, N. & London, K., 2010. Understanding and facilitating BIM adoption in the AEC industry. Automation in construction, 19, pp.988--999.

Haimes, R. & Dannenhoffer, J., 2010. Control of Boundary Representation Topology in Multidisciplinary Analysis and Design. In 48th Aerospace Sciences Meeting.

Horrocks, I., Patel-Schneider, P.F., Van Harmelen, F., 2003. From SHIQ and RDF to OWL: The making of a web ontology language. Web semantics: science, services and agents on the World Wide Web, 1, pp.7--26.

Pu, S. & Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. Intl Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36, pp.12--14.

Qin, L., Deng, X. & Liu, X., 2011. Industry foundation classes based integration of architectural design and structural analysis. Journal of Shanghai Jiaotong University (Science), 16, pp.83--90.

Schlueter, A. & Thesseling, F., 2009. Building information model based Energy Performance Assessment in Early Design Stages. Automation in Construction, 18, pp.153--163.

Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y., 2007. Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web, 5, pp.51-53.

U. Hustadt, B.M.U.S., 2010. KAON2. [Online] Available at: http://kaon2.semanticweb.org/.

Vanland, R., Cruz, C., Nicolle, C., 2008. IFC and building lifecycle management. Automation in Construction, 18, pp.70--78.

W3C, 1995. VRML Virtual Reality Modeling Language. [Online] Available at: http://www.w3.org/MarkUp/VRML/.

Chapter 2

Knowledge Modelling and the Semantic Web

2.1 Introduction

The growth of the World Wide Web has been tremendous since its evolvement both in terms of content and technology. The first Web generation was mainly presentation-based, providing information through the Web pages but not allowing users to interact with them. In short, Web Pages contained 'read only information' since they were only text pages and did not contain multimedia data. The main drawback of these Web sites is that they have higher dependency on the presentation languages, mainly the Hypertext Markup Languages (HTML), (Vaughan-Nichols, 2010). With the introduction of e**X**tensible **M**arkup **L**anguage (XML) (Decker et al., 2000), the information within the pages became more structured. In fact, those XML based pages could hold up the contents in a more structured method but still lacked the proper definition of semantics within the contents, (Berners-Lee, 1998). For this reason, the need of intelligent systems which could exploit the wide range of information available within the Web was widely felt. The Semantic Web was envisaged to address this need.

The term "*Semantic Web*" was coined by Tim Berners-Lee (Lee et al., 2001) proposing the inclusion of semantics for better enabling machine-people cooperation for handling the huge amount of information that exists on the Web. The term "*Semantic Web*" has been defined numerous times. 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 about data or metadata*" (Lee et al., 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*" (Huynh et al., 2007). "*The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise and community boundaries*" (Decker et al., 2000).

Not so far, the Description Logics (DL) (Baader, 2009) is a family of knowledge representation languages which can be used to represent the concept definitions of an application domain in a structured and formally well understood way. The name description logic refers to a concept used to describe a domain and to the logic-based semantics which can be given by a translation into first logic order predicate (Ertel, 2011).

This chapter introduces the basic concept of the semantic web and its related technology; mainly the web ontology language (Antoniou & Harmelen, 2009). To do, we began by a general overview on the knowledge modelling and the different formal language, especially the description logic theory in section 2. In section 3, we highlight the reasoning process and the inference capacities on the DLs language. We discuss in section 4 the different issues related to the definition of the semantic web where we focus on its related technology; mainly the ontology web language (Antoniou & Harmelen, 2009), and the semantic web rule one (Horrocks et al., 2004). Finally, we conclude the chapter with a discussion clarifying the manner in which such a technology can be used for current engineering problems.

2.2 Description logics and Knowledge modelling

Description logics (DLs) (Baader & Sattler, 2001) 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 two terms: description and logic. The former describes the real world scenario with the real world objects and the relationships between those concepts. These objects are more formally grouped together through unary predicates defined by atomic concepts while their relationships are gathered over binary predicates defined by atomic roles. The term ‘logic’ adds the fragrance of logical interpretations to the description. One could reason on the descriptions for generating new knowledge from the existing one through these logics. Definitions are used to introduce symbolic names for complex descriptions. The following example defines a Mother as a Woman who has at least one child. By inference, it means that every individual type of Women which has at least a relation with a Person and the type of the relation is “hasChild”, will be qualified as a Mother.

$$Mother \equiv Woman \sqcap \exists hasChild.Person \tag{1}$$

The description logics are a subset of first order logic offering a family of knowledge representation formalism. These logics inherit from the semantic networks of Sowa (Sowa, 2006) and differ from several other formalisms since it provides a precise semantic characterization of a modelling language. It is based on concepts or classes, and also named roles and relationships. These concepts are defined using unary predicates representing the class of an object with similar characteristics. Roles denote binary predicates representing relationships between objects. Likewise, DLs concepts can also define attributes characterizing an object semantically. It should be noted that recently, the description logics have become the cornerstone of the Semantic Web technology and the Ontology web Language definition (Horrocks & Bechhofer, 2008). This is achieved due to the amount of research over several decades in the area of specification languages for description logics, and validation of algorithms resolving problems and reducing complexity.

2.2.1 Description Logic families

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 (McGuinness & Patel-Schneider, 2003), (Calvanese et al., 2005). Description logics contain the formal logic-based semantics, which present the major reason for its choice of Semantic Web languages over its predecessors. The reasoning capabilities within the DLs have added a new dimension to it. Having these capabilities as a central theme, inferring implicitly represented knowledge becomes possible. Currently, web languages such as XML or RDF(S) (Decker et al., 2000) could benefit from the DL approach to formalize the structured knowledge representation (Lassila, 2007). This has arranged a background behind the emergence of Description Logic languages in the Web. Nowadays, an agreed method to encode these operators using an alphabetic letter to denote expressivity of DLs has seen the light. 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. In the next subsection, we introduce the 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 of:

$$C \sqsubseteq D\,, R \sqsubseteq S \; Or \; C \equiv D, R \equiv S \tag{2}$$

Where C and D are concepts while R and S are roles. Axioms of the first kind are called inclusions, while axioms of the second kind are called equalities. It is used to introduce symbolic names for complex descriptions e,g.

$$RailWorker \equiv Person \sqcap \exists haswork.RailWork \tag{3}$$

Complex descriptions can be built through the above mentioned elementary descriptions of concepts and roles. These descriptions are given different notations over time. The Attributive Language ($\mathcal{AL}$) was 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 ($\mathcal{ALC}$) to allow any concepts or roles to be included and not just atomic concepts and atomic roles which were the previous elements of the descriptions. $\mathcal{ALC}$ is the important notation format to express Description Logics, Table 2-1.

Notation	Syntax	Semantics	Read-as
$\top$	$C, D \rightarrow \top$	$\top(x)$	Universal concept
$\bot$	$\bot$	$\bot\,(x)$	Bottom concept
$\sqcap$	$C \sqcap D$	$C(x) \sqcap D(x)$	Intersection
$\sqcup$	$C \sqcup D$	$C(x) \sqcup D(x)$	Union
$\neg$	$\neg C$	$\neg C(x)$	Negation
$\exists$	$\exists R.C$	$\exists R(x,y) \cap C(y)$	Existential Quantification
$\forall$	$\forall R.C$	$\forall y.R(x,y) \rightarrow C(y)$	Value Restriction

Table 2-1. The syntax and semantics based on $\mathcal{ALC}$

$$C, D \longrightarrow A | \top \,| \bot \,| \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R.\top \tag{4}$$

The presented syntax in the above equation allows defining a set of concepts as the following concept of "A person married to a doctor whose children are all doctors and professors." Such an expression can be formally written as seen in the following equation.

$$Human \sqcap \neg Woman \sqcap (\exists isMarried.Doctor) \sqcap (\forall hasChild.(Doctor \sqcap Professor)) \tag{5}$$

Returning to Table 2-1, several restrictions between concepts and rules are used. Within the DLs language, such restrictions can be classified as:

- **The Quantifier restriction**

It is again classified as the ***existential quantifier*** (*at least one*, or *some) and* ***universal quantifiers*** *(every).* The existential quantifier 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. To simplify, 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*. From the other side, the ***universal quantifier*** (*every)* constraint links a restriction concept to a concept description or a data range. This restriction asserts that the property or relation holds all the member of the domain. To simplify, 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*.

- **The Value restriction**

It links a restriction concept directly to a value which could be either an individual or data value. For example, the $\mathcal{AL}$ $\mathcal{ALC}$ logic is extended with a complete negation. Another example, the $\mathcal{SHIQ}$ description logic is an $\mathcal{ALC}$ logic extended with cardinality quantifier restrictions, inverse roles and relationships. The $\mathcal{AL}$ Description logic also called minimal logic was defined by Schmidt-Schaub and Smolka, (Schmidt-Schauss & Smolka, 1991).

As the DLs language matures, the $\mathcal{SHIQ}$ logic is created. In fact, the derivation of $\mathcal{SHIQ}$ logic with respect to naming the convention of the Description Logic is given as:

- *S*: Used for all $\mathcal{ALC}$ with transitive roles R+
- *H*: Role inclusion axioms R1⊑ R2 (is_component_of ⊑ is_part_of)
- *I*: Inverse Role R-(isPartOf = hasPart-)
- *Q*: Qualified number restrictions

In fact, the $\mathcal{SHIQ}$ clauses authorize the use of the conjunction ⊓, the disjunction ⊔, the negation ¬, the existential quantifier ∀ and the universal quantifier ∃. This logic is extended with transitive roles ($\mathcal{S}$), inverse roles ($\mathcal{I}$), role hierarchy ($\mathcal{H}$), nominal class or enumeration by individuals ($\mathcal{O}$) and restrictive role quantifier ($\mathcal{Q}$). There are a wide variety of description logics describing the authorized operators. However, their naming convention is informal. As in the previous example, $\mathcal{SHIQ}$ expressiveness is encoded in the name using different letters, Table 2-2.

Letter	Definition
$\mathcal{F}$	Functional properties
$\mathcal{E}$	Complete existential Quantification
$\mathcal{U}$	Union
$\mathcal{C}$	Complete Negation
$\mathcal{S}$	Abbreviation of $\mathcal{ALC}$ with all transitif role equivalent to $\mathcal{ALUE}$ since the Union and the completre existential quntification are presented with a complete negation and vis versa
$\mathcal{H}$	Hierarchy of roles and sub properties
$\mathcal{O}$	Nominal Classes or individual Enumeration
$\mathcal{I}$	Inverse properties
$\mathcal{N}$	Role cardinality restriction
$\mathcal{Q}$	Quantifier cardinality restriction on roles
$\mathcal{S}$	Transitive roles
$\mathcal{D}$	Add support for primitives types (Integer, Character chain….)

Table 2-2 DLs Expressivity Definition

To summarize, the following table presents all the available commands in description logics for defining a particular logic.

Constructor	Syntax	Symbols
Universal Concept (top)	$\top$	$\mathcal{AL}$
Empty Concept (bottom)	$\bot$	$\mathcal{AL}$
Conjunction	$C_1 \sqcap \ldots \sqcap C_n$	$\mathcal{AL}$
Disjunction	$C_1 \sqcup \ldots \sqcup C_n$	$\mathcal{U}$
Negation	$\neg C$	$\mathcal{C}$
Universal Quantification	$\forall R.C$	$\mathcal{AL}$
Limited existential quantification	$\exists R.\top$	$\mathcal{AL}$
Existential quantification	$\exists R.C$	$\mathcal{E}$
Transitive Role	R^+	$\mathcal{S}$
Inverse Role	R^-	$\mathcal{I}$

Hierarchical Role	$C \sqsubseteq D$	$\mathcal{H}$
Functional Role	$\leq 1R$	$\mathcal{F}$
Complexes Inclusion of Roles	$RoS \subseteq R$, $RoS \subseteq S$	$\mathcal{R}$
Unqualified number restriction (at least)	$\geq nR$	$\mathcal{N}$
Unqualified number restriction (at most)	$\leq nR$	$\mathcal{N}$
Unqualified number restriction (exactly)	$= nR$	$\mathcal{N}$
Qualified number restriction (at least)	$\geq nR.C$	$\mathcal{Q}$
Qualified number restriction (at most)	$\leq nR.C$	$\mathcal{Q}$
Qualified number restriction (exactly)	$= nR.C$	$\mathcal{Q}$
Nominal (Individuals)	$\{a\}$ or $\{a_1, ..., a_n\}$	$\mathcal{O}$

Table 2-3. Constructors, syntaxes and symbols of the Description Logic

The description of complex concepts is performed using atomic concepts and primitive roles. The meaning of a description is defined by an interpretation function $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the interpretation domain and $\cdot^{\mathcal{I}}$ is the interpretation function. It infers to the correspondence between the intention of concepts and roles with their extension, where atomic concept corresponds to a term representing a subset of the individual interpretation domain. The universal concept (Top) ⊺ is interpreted as the entire domain of the interpretation domain $\Delta^{\mathcal{I}}$, while the empty concept ⊥ corresponds to ∅, Table 2-4.

Constructor	Syntax	Semantic
Empty (resp. Universal)	$\top\ (resp. \bot)$	$\emptyset\ (resp.\ \Delta^{\mathcal{I}})$
Atomic Concept	C	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Individual	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Universal Quantifier	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \vert \forall y\ (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
existential Quantifier	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \vert \exists y\ (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
Restriction at least	$\geq nR$	$\{x \in \Delta^{\mathcal{I}} \vert\ \vert\{y \vert (x,y) \in R^{\mathcal{I}}\}\vert \geq n\}$
Restriction at most	$\leq nR$	$\{x \in \Delta^{\mathcal{I}} \vert\ \vert\{y \vert (x,y) \in R^{\mathcal{I}}\}\vert \leq n\}$
Restriction to less qualified	$\geq nR.C$	$\{x \in \Delta^{\mathcal{I}} \vert\ \vert\{y \in C^{\mathcal{I}} \vert (x,y) \in R^{\mathcal{I}}\}\vert \geq n\}$
Restriction most qualified	$\leq nR.C$	$\{x \in \Delta^{\mathcal{I}} \vert\ \vert\{y \in C^{\mathcal{I}} \vert (x,y) \in R^{\mathcal{I}}\}\vert \leq n\}$

Equivalence Constructor	$C_1 \equiv C_2$	$C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$
Subsumption Constructor	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
Concept Assertion	$a: C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Role Assertion	$\langle a \vert b \rangle: R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Table 2-4. Constructors, syntax and Semantic

2.2.2 Description Logic impact on Knowledge base

Description Logics supports the serialization through the human legible forms of the real world scenario with the classification of concepts and individuals. Moreover, it supports the hierarchical structure of concepts in forms of sub-concept/super-concept relationships between the concepts of a given terminology. This hierarchical structure provides efficient inference through the proper relations between the different concepts. The individual-concept relationship could be compared to the instantiation of an object to its class in an object-oriented concept. In this manner, the DLs approach can be related to the classification of objects in a real world scenario. It provides formalization to knowledge representation of real world situations; otherwise, it should provide the logical replies to the queries of real world situations. This is currently the most researched topic in this domain. The results are highly sophisticated reasoning engines which utilize the expressiveness capabilities of DLs to manipulate knowledge. A Knowledge Representation system is a formal representation of a knowledge described through different technologies. When it is described through DLs, it sets up a Knowledge Base (KB) where the contents could be reasoned on, or inferred. A knowledge base could be considered as a complete package of knowledge content. It is however only a subset of a Knowledge Representation system (KR) that contains additional components. In any graphical representation of knowledge, concepts are represented through the nodes. Similarly the roles are binary relationships between concepts, and eventually present the relationships of the individuals of those concepts. They are represented by links in the graphical representation of knowledge.

Baader (Baader et al., 2008) has sketched the architecture of any KR system based on DLs. The central theme of such a system could be seen as a Knowledge Base (KB). The KB is constituted of two main components: the TBox and the ABox where the *TBox* statements are about the *terms* or the *terminologies* that are used within the system domain. In general, they are statements describing the domain through controlled vocabulary. For example, in terms of a social domain, the TBox statements are the set of concepts as Room, *Rail*, *train*, etc. or the set of roles as *hasGeometry*, *has3DSpatialRelation*, *hasCharacteristics* etc. the ABox contains assertions to the TBox statements. In an object oriented concept, the *ABox* statements must comply with the *TBox* one, through instantiating what is equivalent to classes in TBox and relating the roles to those instances, Figure 2-1.

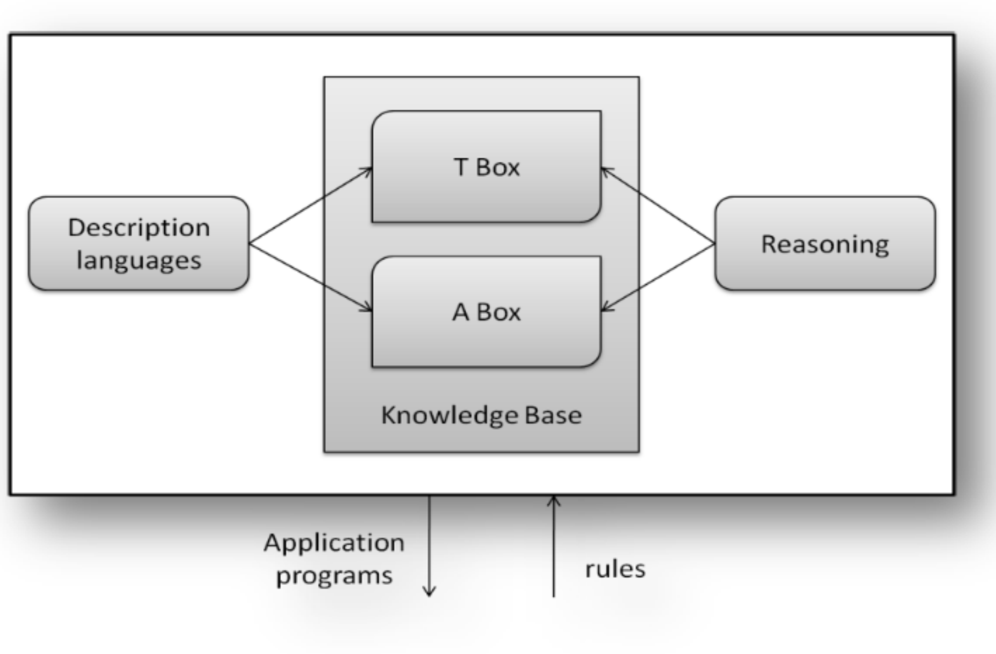


Figure 2-1. The Architecture of a knowledge representation system based on DLs

The definition of a concept is usually done through describing its properties. Two main kinds of axioms describe the structure of the TBox. The first is the axiom of equality, and the second is the axiom inclusion.

$$C \equiv D \text{ or } C \sqsubseteq D \tag{6}$$

Equalities present definitions when the left side is an atomic concept. Definitions are used to introduce symbolic names for complex descriptions.

$$\text{Parent} \equiv \text{Mother} \sqcup \text{Father} \tag{7}$$

An inclusion is a specialization when the left part is an atomic concept.

$$\text{Woman} \sqsubseteq \text{Person} \quad (8)$$

ABox contains a set of assertions about individuals, mainly assertions of belonging and role assertions. Each ABox must be associated with a TBox, because the assertions are expressed in terms of concepts and roles from the TBox. In the following example, a, b and c are individuals; C is a concept and R a role.

$$C\ (a)\ -\ R\ (b,\ c) \quad (9)$$

For example, if John, Paul and Mary are individuals, then Father (John) means that John's father where hasChild (Mary,Paul) means that Paul is a child of Mary.

Description logics adopt the unique name assumption, which means that individuals must have different names. Otherwise, individuals are the same. The knowledge bases based on description logics adopt the semantics of the open world. Indeed, the open world assumption implies that the information may be incomplete. This means that what cannot be proven from the available information is not necessarily false. Unlike the closed world assumption, it implies that the information in the knowledge base is necessarily complete. Otherwise, what cannot be proven from the available information is false. As example, the knowledge base is composed of individuals following Man (John), Man (Paul) and the assertion of the following role: hasChild (John, Paul). The question is: Are all John’s children men? The answer is true in the case of a closed world, such as is adopted in the field of relational databases. On the contrary, the outcome is unknown to the semantics of the open world, because no information is available stating that Paul is the only child of John.

2.3 Reasoning with Description Logic Languages

Traditionally, the reasoning process refers to the inference of new facts from a set of existing ones which have the characteristic of being true. The reasoning process formally verifies a semantic relationship between specific facts in a formal logic, called logical implication relation. In classical logic, the expression of “a proposition P logically implies a proposition Q” means" proposal $\neg P \vee Q$ is true. Formally this is written P => Q.

In general, the logic is equipped with a system of rules for deriving conclusions from a set of hypothesis. Such a system is called evidence computation and generally designed to faithfully capture the semantic relationship of a logical implication. In the standard view, decision problems used to answer questions with ‘yes’ or ‘no’. However, despite the fact that this is considered as a standard reasoning task, checking logical implication is still far from the inference tasks made by applications. Indeed, the problems of finding a solution that involves a question, or rather an injunction in the form of "find an element such as ...", and the answer is to provide such an element are presented nowadays as very common ones. According to the expressiveness of the language for defining the facts of a problem domain, the proofs calculation system can be incalculable. The description logics are less expressive than first order logic, but they form treatable systems of knowledge representation. The current task of reasoning carried on description logics is subsumption checking from one side, and satisfiability verification from another. The first task is to formally prove for a description of concepts that is more specific than another. This can be seen as a specific form of verification of logical implication. The second task is for a given knowledge base to determine whether it is satisfiable, that is to say, it does not contain contradictions. Often, the reasoning tasks such as logical implication in many logics can be expressed in terms of checking satisfiability. The solution adopted for this thesis is based on description logics and especially on the OWL-DL described in the next section. The purpose of this section is the presentation of the standard reasoning tasks and inference using logic programming.

2.3.1 Inference in Description Logic

The inference is performed in description logic through the terminological level TBox, or even the assertional level, taking into account the individuals of the knowledge base like the ABox. In this field, four principal inference concepts are presented on the terminological level (Baader, 2006)

- **Satisfiability**: A concept C of a terminology $\mathcal{T}$ is satisfiable if there exists a model $\mathcal{I}$ of $\mathcal{T}$ where $C^{\mathcal{I}} \neq \emptyset$.
- **Subsumption**: A concept C is subsumed by a concept D referring to a terminology $\mathcal{T}$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for any model $\mathcal{I}$ of $\mathcal{T}$.
- **Equivalence***:* A concept C is equivalent to a concept D referring to a terminology $\mathcal{T}$ and only if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every model $\mathcal{I}$ *of* $\mathcal{T}$.

- **Disjunction**: A concept C is Disjoint to a concept D referring to a terminology $\mathcal{T}$ and only if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for every model $\mathcal{I}$ *of* $\mathcal{T}$

From the other side, the factual level includes four main inference issues**:**

- **Consistency**: An ABox A is consistent with respect to a TBox $\boldsymbol{\mathcal{T}}$ if there exists a model $\boldsymbol{\mathcal{I}}$ of $\mathbf{A}$ and $\boldsymbol{\mathcal{T}}$**.** For example, the set of assertions {Mother (Mary), Father (Mary)} is consistent respecting the empty TBox, since no restrictions on the interpretation of Father and Mother concepts to have common individuals. However, this is no longer true when interpreting the Mother and Father concepts as disjoint.
- **Instance Checking**: Check by inference if an assertion C (a) is true for every model $\boldsymbol{\mathcal{I}}$ of an ABox A and a TBox $\boldsymbol{\mathcal{T}}$.
- **Role Verification**: Check by inference if an assertion R (a, b) is true for any model $\boldsymbol{\mathcal{I}}$ of a ABox A and a TBox $\boldsymbol{\mathcal{T}}$.
- **Retrieval problem**: For an ABox A, a concept C and a terminology $\boldsymbol{\mathcal{T}}$, inferring individuals $\boldsymbol{a_1^{\mathcal{I}}} \ldots \boldsymbol{a_n^{\mathcal{I}}} \in \boldsymbol{C^{\mathcal{I}}}$ for every model $\boldsymbol{\mathcal{I}}$ of $\boldsymbol{\mathcal{T}}$**.**

2.3.2 Logic programming and inference

Logic programming has reached a wide variety of application fields such as the design of expert systems in order to simulate human expertise, the design of RDBMS (Malecha et al., 2010), natural language processing, the e-learning domain, etc. Such a programming language has three major advantages: simplicity, power and non-directional procedures. The declarative aspect of logic programming provides a simple way for solving problems. The programmer's task is also reduced to the description of knowledge and problem solving. Logic programming is based on the idea that predicate logic restricted to Horn clauses can give a procedural interpretation. Programming languages including Prolog logic programming (Swift & Warren, 2012) are non-deterministic primitive operations, where the concept of unification is a central notion of logic predicate and other logic systems. This notion characterizes the Prolog programming languages. To apply the rule with two clauses, it is necessary to know if two or more atomic formulas can be unified, (Baader & Morawska, 2009). Two terms A and B can be unified if there exists a substitution σ where σ (A) = σ (B) (σ is a unifier of A and B). The unification is the process by which any logic language matches a fact with an atom, where the head of rule is used to check the suggested purpose. The algorithm used to implement this process is called the unification algorithm.

$$unify\big(siblings(John, Z), siblings(X, Y)\big) = [X = John, Z = Y] \tag{10}$$
$$unify\big(same(X, X), same(\text{Mary}, Y)\big) = [X = \text{Mary}, Y = \text{Mary}]$$
$$unify\big(cons(X, null), cons(X, Y)\big) = [Y = null]$$
$$unify\big(cons(X, null), cons(X, a)\big) = < Fail >$$

Horn logic programming is a declarative programming paradigm which is based on a subset of first order logic. A logic program consists of simple rules, as follows: "if ... then ...“. These rules are a simple way to represent knowledge. A rule consists of a head and a body.

$$H{:}-B_1, \dots, B_n \tag{11}$$

The head is composed of a literal (H in the above example), while the body of a rule is composed of a set of literals (B_1, ..., B_n in the above example). There are two different notations, which are semantically identical. H:-B_1, ..., B_n is sometimes written H ← B_1 ∧ ... ∧ B_n. In all cases, these rules are read as follows, if the "body" then "head". Literals are atoms that can be either positive p (x) or negative ¬ p (x). A rule without a body is called a fact p (a_1, ..., a_n). A rule without a head is called a query query:?-B_1, ..., B_n, Table 2-5.

Atom	Example
Rule	$hasDaughterl(x, y){:} -haschild(x, y), Woman(y)$ $hasGrandFather(x, z){:} -hasParent(x, y), hasFather(y, z)$
Fact	$Person(John\ )$ $Woman(Mary)$
Query	$query{:}\,?-hasGrandFather(John\ , x)$ $query{:}\,?-hasDaughterl(John\ , Mary)$

Table 2-5. Example of Horn Clause

2.3.2.1 *Prolog*

Prolog (Programming in Logic) (Demoen & DE LA BANDA, 2012) is a logic programming language based on predicate calculus of first order, which was originally limited to only Horn clauses. However, this language has been extended to take into account negation by failure. As described for logic programming, a Prolog program consists of rules and facts. These facts and rules are operated by a theorem prover or inference engine to respond to a question or request. The implementation of the prolog program is based on the resolution principle with specific strategies of restriction. This program is initiated by a query such as: *?-hasChild (x, y)* that results in the enumeration of all possible answers. A Prolog program is declarative where the order of rules is important for the program evaluation, Table 2-6.

Program	Associated questions and answers
Woman(Alexandra)	?- isChild(Alexandra, John).
Woman (Mary)	false
Man(John)	
Man (Peter)	?- isParent(John, Peter).
	true
isMother (Alexandra, Mary).	
isMother(Alexandra, John).	?- isParent(Alexandra,X).
isFather (John, Peter).	X=Mary;
isParent(X,Y) :- isMother (X,Y).	X= John;
isParent(X,Y) :- isFather (X,Y).	
isChild(Y,X) :- isParent (X,Y).	?- isFather(John,X), Woman(X).
	False

Table 2-6. Example in Prolog

2.3.2.2 *Datalog*

Datalog (Gottlob & Schwentick, 2011) is in many ways a simplified version of the general logic programming. In fact, a logic program consists of facts and rules where facts are assertions of a relevant part of the world as "John Henry is the father." A rule is a sentence that allows us to deduce facts from other ones. If X is the parent of Y, and if Y is the parent of Z, then X is a large relative to Z. In the formalism of Datalog, facts and rules are also represented in the form of Horn clause. It should be noted that for a given Datalog program, particular symbols not defining variables are constant or predicate symbols. In addition, all literals of the same predicate symbol have the same priority and number of arguments. Any literal fact, rule or clauses without variables are called ("ground"). Any Datalog program *P* must satisfy the following safety conditions:

- All the facts of P present identificators
- Any variable that appears in the head of the rule P must appear in the body of the same rule.

These requirements help to ensure that any set of facts derived from the Datalog program are closed. In the context of logic programming, it is assumed that all knowledge (facts and rules) within a particular field of application is included in the logic program. However, the Datalog programs were developed for use on a very large number of facts and are stored in a relational database. Therefore, we can consider two sets of clauses, a set of fact identifiers called extensional database, physically stored in a relational database, and a Datalog program P called intentional database. Today, a surprising commercial re-emergence is recognized in the field of Datalog.

2.4 The Ontology Web Language and its related Technology

Currently, the convergence of formal foundations for extensible, semantically understood structure within the description logic and the Web languages has led to efforts such as Ontology Interface Language (OIL) (Fensel et al., 2001). It presents 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 was the first time that the concept within ontology is explicitly used within a Web based environment. However, it did not completely leave out the primitives of frame base languages with the formal semantics and reasoning capabilities by including them within the language. As the Semantic Web technologies matured, the need of incorporating the concepts behind the description logic within the ontology languages was realized. It took few generations for the ontology languages defined within the Web environment to implement the description language completely. The Web Ontology Language (OWL) (Antoniou & Harmelen, 2009) is intended to be used when the information contained in documents needs to be processed by applications and not by humans (McGuinness & Van Harmelen, 2004). The OWL language has direct influence from research on the Description Logics particularly on the formalization of the semantics. In addition, the OWL language has its correspondence to the description logics with its sublanguages as OWL DL and OWL Lite (McGuinness & Van Harmelen, 2004). The Semantic Web includes a number of interrelated technologies organized in different complementary layers where each upper layer presents additional technology and language, making the previous one richer Figure 2-2. In the next part, a quick survey on the different layers will be highlighted.

- URI (Uniform Resource Identifier). The *Uniform Resource Identifier* layers present the basic layer for the semantic web technology, and mainly the Ontology Web Language. It allows characterizing each resource by a unique identifier thus enabling its identification on the network.
- XML (Extensible Markup Language). The second main layer in the semantic web stack is presented through the XML language (Hunter et al., 2011). Indeed, the extensible markup language (XML) is a simplification of the Standard Generalized Markup Language SGML (Kahn, 1999). Its initial goal is to ensure the interoperability between the different heterogynous environments, to facilitate the automatic content exchange between various information systems and particularly to get adapted for sending documents over the Web.
- RDF (Resource Description Framework) is a data model that adopts a syntax allowing tags to represent objects or resources and the relationships between these objects as triplets, (Decker et al., 2000). It is intended to describe the Web Resources and their metadata in a formal way. This layer is structured in several RDF triplets. Every RDF triple is an association of subject, predicate; object where the subject represents the resource describing the type of predicate representing a property applicable to the resource and the object represented given one or another resource.
- RDF-Schema or RDFS vocabulary, (Allemang & Hendler, 2008). As an extension for the RDF layer, the RDFS one is used to describe classes and properties for RDF resources. This extensible language for knowledge representation belongs to the family of semantic Web languages and provides basic elements for the vocabulary definition of ontologies that was intended to structure RDF resources.
- Ontologies and logic. These layers present the main contribution within the semantic web technology compared to the RDF ones. (Antoniou & Harmelen, 2009). It is related to the definition of ontology languages such as OWL (Ontology Web Language). OWL is advocated as a standard by the W3C consortium for modelling ontologies. Added to all the capacities inherited from

the above discussed layers, OWL can model the information domain through new constraints and DLs rules characterising the domain.

- Evidence. As seen in the last section, logical languages enable the implementation of reasoning tools. Reasoning tools are available for languages such as OWL, and allow for example to test the consistency of information, to classify it, etc... It also includes inference engines that allow inferring information from the described one, in order to unify these different approaches. The specification language of the Semantic Web Rule Language (SWRL) and others presents part of the Emerging work, (Horrocks et al., 2004).
- The Confidence layer is located at the top of the pyramid. It addresses the issues of trust that the Semantic Web technology can support. It concerns the use of digital signatures and other types of knowledge ability to guarantee the origin of information.

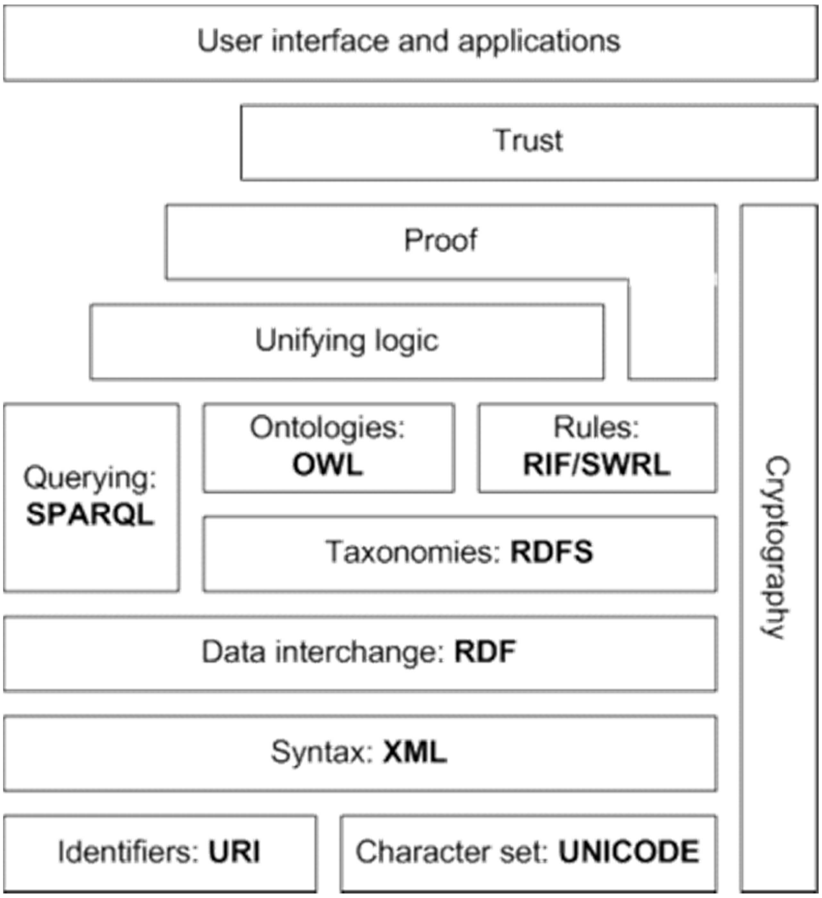


Figure 2-2. Semantic Web Layers

2.4.1 The Ontology Web Language (OWL)

The association of knowledge with the Semantic Web has provided a scope for information management through the knowledge management. Since both technologies use ontology to conceptualize the scenarios, Semantic Web technology could provide a platform for developments of knowledge management systems (Uren et al., 2006). The term Ontology has been used for centuries 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, ontologies are seen as 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, 2008). It can be considered as formalization of knowledge representation where the Description Logics (DLs) provide logical formalization to the Ontologies (Horrocks et al., 2007). Ontology defines the basic terms and relationships comprising the vocabulary of a topic area as well as the rules for combining terms and relationships to define extensions to the vocabulary. According to this definition, ontology includes not only the terms that are explicitly defined in it, but also terms that can be inferred from it. Gruber, (Gruber, 2008) initially define ontology as "*an explicit formal specifications of the terms in the domain and relations among them since it defines a common vocabulary for researchers who need to share information in a domain"*. In other words, ontology is a formal explicit description of concepts in a domain of discourse; the properties of each concept describing various features and attributes of the concept, and restrictions on slots. Ontology, together with a set of individual instances of classes, constitutes a knowledge base. This definition became the most quoted in literature, as well as by the OWL community. Borst (Borst et al., 1997) has slightly modified Gruber's definition as follows: *Ontologies are defined as formal specification of a shared conceptualization*. It refers to abstract conceptualization model of some phenomenon in the world where concepts are identified within its phenomenon. That means explicitly clustering the different kinds of used concept. Guarino and Giaretta (Guarino, 2009) provide the following definition: *ontology is a set of logical axioms designed to account for the intended meaning of a vocabulary*.

The OWL language has direct influence from the researches in Description Logics and insight from Description Logics, particularly on the formalization of semantics. OWL takes the basic fact-stating ability of RDF (Allemang & Hendler, 2008) and the class- and property-structuring capabilities of the RDF Schema and extends them in important ways. OWL can declare classes, and organise these classes in a subsumption ("subclass") hierarchy, as can RDF Schema. OWL classes can be specified as logical combinations (intersections, unions, or complements) of other classes, or as enumerations of specified objects, going beyond the capabilities of RDFS. OWL can also declare properties, organize these properties into a "subproperty" hierarchy, and provide domains and ranges for these properties, again as in RDFS. The domains of OWL properties are OWL classes, and ranges can be either OWL classes or externally-defined datatypes such as string or integer. OWL can state that a property is transitive, symmetric, functional, or is the inverse of another property, extending here again the RDFS. Added to that, OWL can express which objects (also called "individuals") belong to which classes, and what the property values are such a specific individuals. Equivalence statements can be made on classes and on properties, disjointness statements can be made on classes, and equality and inequality can be asserted between individuals. However, the major extension over RDFS is the ability in OWL to provide restrictions on how properties behave that are local to a class. OWL can define classes where a particular property is restricted, so that all the values for the property in instances of the class must belong to a certain class (or datatype); at least one value must come from a certain class (or datatype); there must be at least certain specific values; and there must be at most a certain number of distinct values. The semantics of this language is definable using its translation into description logic. It's not a coincidence, because this correspondence allows OWL to exploit the results of field description logic with respect to the decidability and complexity of key inference problems. Moreover, this correspondence allows applications using OWL to use inference engines. The next two tables show in detail the constructors and the axioms of the OWL language via the DLs language.

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap C_2$	$Human \sqcap Person$
unionOf	$C_1 \sqcup C_2$	$Doctor \sqcup Person$
complementOf	$\neg C$	$\neg Human$
one of	$\{x_1 \dots x_n\}$	$\{John, Mary\}$
allValueFrom	$\forall\, r.C$	$\forall\, hasChild.Doctor$
someValueFrom	$\exists\, r.C$	$\exists\, hasChild.Doctor$

hasValue	$\exists\, r.\{x\}$	$\exists\, Live\_in.\{Germany\}$
minCardinality	$(\geq\ n\, r)$	$(\geq\ 2\ hasChild)$
maxCardinality	$(\leq\ n\, r)$	$(\leq\ 1\ hasChild)$
inverseOf	r^{-}	$hasChild^{-}$

Table 2-7. OWL constructors

Axiome	Syntaxe DL	Example
subClassOf	$C_1 \sqsubseteq C_2$	$Human \sqsubseteq Animal \sqcap Biped$
equivalentClass	$C_1 \equiv C_2$	$Man \equiv Human \sqcap Male$
subPropertyOf	$P_1 \sqsubseteq P_2$	$hasDaughter \sqsubseteq hasChild$
equivalentProperty	$P_1 \equiv P_2$	$Cost \equiv Price$
disjointWith	$C_1 \sqsubseteq \neg C_2$	$Male \sqsubseteq \neg Femele$
SameAs	$\{x_1\} \equiv \{x_2\}$	$\{Chancellor$ _Merkel } ≡ {Angela _Merkel }
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	$\{John\} \sqsubseteq \neg\{Alex\}$
TransitiveProperty	$P\ transitif\ Role$	$is\ Ancetor$
FunctionalProperty	$\top \sqsubseteq (\leq\ 1\ P)$	$\top \sqsubseteq (\leq\ 1\ isMotherOf)$
InverseFunctionalProperty	$\top \sqsubseteq (\leq\ 1\ P^{-})$	$\top \sqsubseteq (\leq\ 1\ hasMother^{-})$
SymmetricProperty	$P \equiv P^{-}$	$isMarried \equiv isMarried$

Table 2-8. OWL Axioms

2.4.1.1 *OWL ontology types and creation strategies*

The semantic web is a vision pioneered by Sir Tim Berners-Lee, in which information is expressed in a language understood by computers. In short, it is a layer that describes concepts and relationships, following strict rules of logic. The purpose of the semantic web is to enable computers to "understand" semantics the way humans do. Equipped with this "understanding," computers will be theoretically able to solve problems that are not possible today. Ontologies present one of the most famous technologies for knowledge modelling and semantic web creation, where the basic idea was to present information using mathematical graphs and logical structure to make computers able to understand and process easily and automatically. As seen in the last section, ontology is composed by Classes, Instances, Relations, Functions and Axioms. While designing an OWL ontology knowledge base, designers should not think that ontologies are just made for machines, but for humans also. To do so, ontologies must respect some criteria in the design steps like:

- Clarity: Objective, formal and complete definitions.
- Coherence: Inferred knowledge consistent with definitions.
- Extendibility: Easy and fast extension and specialization.
- Minimal encoding bias: Conceptualization at knowledge-level (not implementation level).
- Minimal ontological commitment: Support for the intended knowledge sharing tasks (not to represent the entire world).

Van Heijst (Van Heijst et al., 1997) and McGuinness (Lassila, 2007) have proposed two types of classification of ontologies according to different criteria. The first classification is based on the type and abundance of structures used in the ontology. Otherwise, according to the ontology expressiveness, the main categories and their meanings are:

- The terminological ontologies that are used to specify the terms of the vocabulary of a field of knowledge.
- Ontologies that specify the information structure / diagram of a database to allow storage of information.
- Ontologies that model knowledge offering internal structures that are richer and more defined according to their uses such as information sharing.

They also propose a classification of ontologies, based on the consideration of "objectives" of modelling. They identify four categories of ontologies according to this criterion:

- The application ontologies that specify the necessary information on one or more specific application areas.
- The domain ontologies that express the conceptualization of knowledge of a particular area.
- The generic ontologies that model knowledge of the transverse to different areas. Typically, generic ontologies define concepts such as ideas of state, of event, action, etc..
- The representation of ontologies that serve to explain the design utilisations underlying formalisms of representation of knowledge. They represent the real world entities.

In the same context, Lassila (Lassila, 2007), and as a mixture of the both discussed criteria, has differentiated between the domain ontologies and the upper-level ontologies. A domain-specific ontology models a specific domain (medical, pharmaceutical, engineering, law, enterprise, automobile, etc..). It provides vocabulary about concepts within a domain and their relationships, taking up into account the activities of that domain, and the theories and elementary principles governing it. The upper-level ontology (Also known as foundation ontology or top-level ontology) is used to describe very generic common concept across the domains and the general notions under it. There is a clean boundary between domain and upper-level ontologies. The concepts in domain ontologies are usually specializations of concepts already defined in top-level ontologies, and might occur with the relationship. In some cases, top-level ontologies are used to build domain ontologies.

2.4.1.2 *OWL ontology sub-Language*

The Ontology Web Language has provided three main sublanguages with incremental expressiveness designed for different communities and users.

- *OWL Lite:* It presents the main necessary capacity of expressiveness supporting users' primary needs like classes' hierarchy classification and very simple constraint implementation. As an example, and while it supports cardinality constraints, such a language can only support cardinality values of 0 or 1. OWL Lite, as the simplest language, has a very lower formal complexity compared to OWL DL.
- *OWL DL:* It presents the Ontology Web Knowledge based on Description Logic languages. It is designed for engineers who need extreme expressiveness, while producing a knowledge base with completeness and decidability. It is named as OWL DL due to its correspondence with description logics presenting the logics that form the formal foundation of OWL.
- *OWL Full:* Designed for users who need maximum expressiveness capabilities and mainly full syntactic freedom from RDF basis. In this owl field, a class can be treated in the meantime as a collection of individuals and as an individual in his own right. In fact, OWL Full allows to increase the Semantic of the RDF-OWL vocabulary. Unlikely, any reasoning software can support complete reasoning for all features of such a variation.

Actually, each of these sublanguages is an extension of its simpler predecessor respectively, where every OWL Lite ontology is OWL DL ontology. Likewise, each OWL DL ontology is also an OWL Full ontology. Each valid OWL Lite conclusion is a valid OWL DL conclusion. Finally, every valid OWL DL conclusion is a valid OWL Full conclusion. Figure 2-3 presents the language of description logic, beginning from the simplest attributive language and arriving at the most complex language, materialized via the description logic applied to the Semantic Web. It presents the different semantic web language and the associated DLs for each of them.

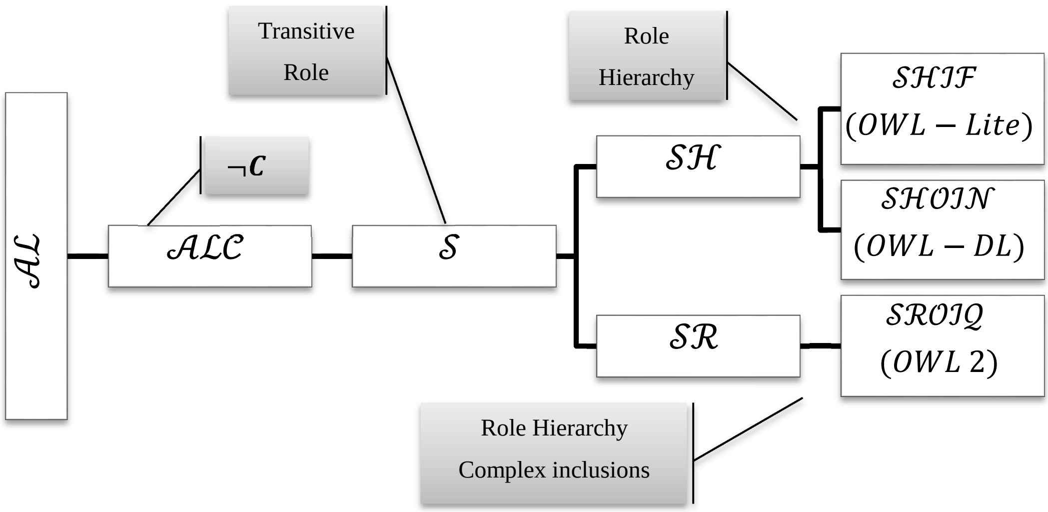


Figure 2-3. Description logic families and Semantic Web languages

2.4.1.3 *OWL 2*

Since the inception of the Semantic Web, the development of languages for modelling ontologies has been seen as a key task. The initial proposals focused on RDF and RDF Schema; however, these languages were soon found to be too limited in expressive power. OWL Web Ontology Language became a W3C recommendation in February 2004. As seen in the last section, OWL is actually a family of three language variants of increasing expressive power: OWL Lite, OWL DL, and OWL Full. The standardization of OWL has sparked off the development and/or adaption of a number of reasoners, including FacT++, (Tsarkov & Horrocks, 2006) Pellet, (Nguyen & Nguyen, 2010), RACER, (Haarslev & Muller, 2001) and (Shearer et al., 2008), and ontology editors, including Protégé (Protege, 2012) and Swoop, (Kalyanpur et al., 2006). Up to a few months ago, practical experience with OWL showed that OWL DL presents the most expressive and decidable language of the OWL family. Moreover, it lacks several constructs that are often necessary for modelling complex domains. OWL 2 is a new version of OWL ontology language which considerably improves the datatype (Motik et al., 2009). Apart from addressing acute problems with expressivity, the goal of OWL 2 was to provide a robust platform for future development. OWL 2 extends the W3C OWL Web Ontology Language with a small, but useful, set of features and effective reasoning algorithms. The new features include an extra syntactic layer, additional property and qualified cardinality constructors, extended datatype support, simple meta-modelling, and extended annotations. In parallel, considerable progress has been achieved in the development of tools supporting OWL 2. The new syntax is currently supported by the new version of the OWL API. The widely used Protégé system has recently been extended with support for the additional constructs provided by OWL 2. Support for OWL 2 has also been included into the FaCT++ and the Pellet systems.

2.4.2 Semantic Web Rule Language (SWRL)

Within the computer science domain, the process of inference aims to apply logic to come to a conclusion from observations and assumptions where the inference engines are applications that derive answers from a knowledge base using a logical program and rules in our case. These rules are used to improve the knowledge contained in an ontology structure. The logic of Horn forms a platform to establish rules particularly based on the syntax of RuleML language (Horrocks et al., 2004). In this context, several languages have emerged during the last decade. One of these languages which evolved quickly is the Semantic Web Rule language known as SWRL (Horrocks et al., 2004). Semantic Web Rule Language (Valiente-Rocha & Lozano-Tello, 2010) is a rule language based on the combination of the OWL-DL with Unary/Binary Datalog RuleML (Boley et al., 2001) which is a sublanguage of the Rule Markup Language. SWRL rules include a high-level abstract syntax for Horn-like rules. The SWRL has the *antecedent→consequent*, form where both antecedent and consequent are conjunctions of atoms written a_1 ... a_n. To detail this, 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 URI ref, x and y are either individual-valued variables or OWL individuals, and z, z1, … zn are either data-valued variables or OWL data literals. Within the swrl rules, variables are indicated by using the standard convention of prefixing them with a question mark (e.g., ?x). The same as OWL language, the URI references (URI refs) are used to identify ontology elements such as classes, individual-valued properties and data-valued ones. For instance, the following famous rule example, asserts that ones parents' brothers are ones uncles where parent, brother and uncle are all individual-valued properties.

$$\text{Parent}(?x, ?p) \wedge \text{Brother}(?p, ?u) \rightarrow \text{Uncle}(?x, ?u) \tag{12}$$

The set of built-ins for SWRL language is motivated by a modular approach to extend the language for future versions. In addition, this approach is based on the reuse of the built-ins defined in the XQuery (Chamberlin, 2002) and XPath language (Boag et al., 2007), which are themselves based on the XML language and its data types. This built-in system should allow the interoperation of SWRL rules with other formalisms by providing an extension, modular infrastructure built-ins for languages of the Semantic Web, Web services and Web applications. These built-ins are keys for any external integration. They help in the interoperation of SWRL with other formalism and provide an extensible infrastructure knowledge based applications. Currently, Comparison Built-Ins, Math Built-Ins and Built-Ins for Strings are already implemented within lots of platforms for ontology management like protégé.

Comparison Built-Ins

The built-ins in this category are operators used to compare two values. W3C has listed the built-ins entrants in this category, (Oconnor et al., 2005).

Built-ins	Description
swrlb:equal	Satisfactory if and only if the first argument is the same as the second argument.
swrlb:notEqual	Satisfactory if and only if the first argument is not the same as the second argument.
swrlb:lessThan	Satisfactory if and only if the first argument is smaller than the second argument.
swrlb:lessThanOrEqual	Satisfactory if and only if the first argument is equal to the second argument or small.
swrlb:greaterThan	Satisfactory if and only if the first argument is greater in the

	second argument.
swrlb:greaterThanOrEqual	Satisfactory if and only if the first argument is greater than or equal to the second argument.

Table 2-9. Comparisons Built-Ins

Mathematic Built-Ins

These built-ins are defined for digital data. These are mathematical operations implemented for SWRL. W3C has listed the built-ins entrants in this category, (Oconnor et al., 2005).

Built-ins	Description
swrlb:add	Satisfactory if and only if the first argument is the arithmetic sum of the second and the third argument.
swrlb:subtract	Satisfactory if and only if the first argument is the arithmetic subtraction of the second and the third argument.
swrlb:multiply	Satisfied if the first argument is equal to the arithmetic product of the second argument through the last argument.
swrlb:divide	Satisfactory if and only if the first argument is the arithmetic division of the second and the third argument.
swrlb:integerDivide	Satisfactory if and only if the first argument is the arithmetic division of the second and the third argument.
swrlb:mod	Satisfactory if and only if the first argument is the modulus of the second by the third argument.
swrlb:pow	Satisfactory if and only if the first argument is equal to the second argument to the power of the third argument.
swrlb:unaryPlus	Satisfactory if and only if the first argument is equal to the second argument with the sign unchanged.
swrlb:unaryMinus	Satisfactory if and only if the first argument is equal to the second argument with the opposite sign.
swrlb:abs	Satisfactory if and only if the first argument is equal to the absolute value of the second argument.
etc.	…

Table 2-10. Mathematical Built-Ins

String Built-Ins

These built-ins are specially designed to manipulate strings. They cannot be used for non-literal types. The W3C, (Oconnor et al., 2005) has listed the built-ins entrants in this category.

Built-ins	Description
swrlb:stringEqualIgnoreCase	Satisfactory if and only if the first argument is equal to the second argument. The check box is ignored.
swrlb:stringConcat	Satisfactory if and only if the first argument is equal to the concatenation of the second argument and the last argument.
swrlb:substring	Satisfactory if and only if the first argument is equal to the sub string of characters in the second argument, the optional length is given in the fourth argument, and the offset is given as the third argument.
swrlb:stringLength	Satisfactory if and only if the first argument is the size of the string corresponding to the second argument.
etc.	…

Table 2-11. String Built-Ins

Several other built-ins are not referenced in this section, which are mainly date, time and duration built-Ins.

2.4.3 Query language (SPARQL and SQWRL)

SPARQL (Sirin & Parsia, 2007) presents the acronym for Simple Protocol and RDF Query Language. It defines a standard query language and data access protocol with the Resource Description Framework (RDF) data model. It works for any data source that can be mapped to RDF. The specification is under development by the RDF Data Access Working Group (DAWG). Essentially, SPARQL is a graph-matching query language. Given a data source D, a query consists of a pattern which is matched against D, and the values obtained from this matching are processed to give the answer. The data source D to be queried can be composed of multiple sources. Finally, the output of a SPARQL query can be of different types: yes/no queries, selections of values of the variables which match the patterns, construction of new triples from these values, and descriptions about resource queries.

As a main limitation, and while using SPARQL for OWL forms, a particular serialization of OWL into RDF has to be first handled where all the OWL- specific semantics from the query form will get lost. As an alternative, SQWRL, (OConnor & Das, 2009) gives access to the OWL semantics and is based on the notion of DL-safe rules. Actually, SQWRL (Semantic Query-Enhanced Web Rule Language) is a SWRL-based language for querying OWL ontologies. It provides SQL-like operations to retrieve knowledge from OWL knowledge base. Like SQL, as a basic set operator, it allows Counting operator, Disjunction, Complex Counting and Aggregation like sqwrl:isEmpty, sqwrl:union, sqwrl:difference, mathematical and logical predicates like: sqwrl:max, sqwrl:min, sqwrl:sum, sqwrl:orderBy. Added to that, SQWRL can act as a DL query language. With such a language, there is no need to invent a new semantic where standard presentation syntax is adopted. Likewise, it can use existing reasoning infrastructure and editors where mainly queries can interoperate with rules. Finally, the same set of built-ins already discussed for SWRL rules and SPARQL queries still valid for the SQWRL rules. The next example presents a SQWRL rule including the mathematical built-ins “swrlb:greaterThan”. Once executed, the different 3D geometries respecting certain characteristics will be relatively selected.

_3D_Geometry(?x) ∧ hasHeight(?x, ?h) ∧ swrlb:greaterThan(?h, 5) → sqwrl:select(?x) (13)

2.5 Conclusion and discussion

This chapter has introduced the different principles related to the Description Logic concepts, the knowledge engineering concept, and mainly the semantic Web and Ontology Web Language. Semantic Web technology is slowly modernizing the application of knowledge technologies and though they existed before the Semantic Web, the implementation in their fullness is just being realized. Our current research is linked to the above mentioned concept and technologies. In fact, this research benefits from the existing OWL languages, the existent inference engines through the inference rules and reasoning engines to reason the knowledge. However, the actual research works moves beyond semantic reasoning and semantic rule processing and attempts to explore the 3D spatial domain from a semantic point of view.

In the next chapters, we take a step forward, based on the semantic web technology via the use of the main inference capacity of the knowledge engineering areas. It lays its foundation on the Semantic Web Rules Language and the DLs capacities to infer on existing knowledge base adjusted with 3D Spatial Operators. In this context, the semantic web technology will be used to model knowledge and to infer knowledge on an existing one. These bases formally define the semantic handled by the 3D spatial domain. This research thesis aims to integrate the correspondent spatial domain within the Semantic web technology in order to handle it in a qualitative manner. Such integration will pen new perspective since it will enable calculation optimisation and mainly the processing simplification, since the user will react in his natural meaning with the machine in this case. Otherwise, it aims to implement new 3D spatial rule inference managing the 3D domain, mainly the geometry and the 3D spatial relation in a qualitative manner.

It can be seen from the above detailed literature, how Ontology structure within SWRL and Built-Ins can offer much more flexibility through defining the 3D Spatial relations as Built-Ins. Besides the definition of the Spatial concept in the ontology and the different Built-Ins, this research studies the best suitable way to present geometries, qualifying 3D Spatial topology, computing relations, and mainly linking the low level physical model to the high level semantic one. The Ontology web language knowledge base will handle the semantic of what we have to model, otherwise, the 3D geometries, its characteristics and relations. While the inference capacities of the SWRL rules and the DLs logic will be used to process the spatial domain in a qualitative manner, to do, new “Spatial” Built-ins linking 3D geometry to their qualitative representation will be created and mapped.

References

Allemang, D. & Hendler, J.A., 2008. Semantic web for the working ontologist: modeling in RDF, RDFS and OWL. Morgan Kaufmann.

Antoniou, G. & Harmelen, F., 2009. Web ontology language: OWL. Handbook on ontologies, pp.91--110.

Baader, F., 2006. Nonstandard inferences in description logics: The story so far. Mathematical Problems from Applied Logic I, pp.1--75.

Baader, F., 2009. Description logics. Reasoning Web. Semantic Technologies for Information Systems, pp.1--39.

Baader, F., Horrocks, I., Sattler, U., 2008. Description logics. Foundations of Artificial Intelligence, 3, pp.135--179.

Baader, F. & Morawska, B., 2009. Unification in the Description Logic EL., 2009.

Baader, F. & Sattler, U., 2001. An overview of tableau algorithms for description logics. Studia Logica, 69, pp.5--40.

Berners-Lee, T., 1998. Semantic web road map. [Online] Available at: http://www.w3.org/DesignIssues/Semantic.html [Accessed 24 September 2012].

Berglund, A., Boag, S., Chamberlin, D., Fernandez, M.F., Kay, M., Robie, J., Simeon, J., 2007. XML Path Language (XPath) 2.0. W3C recommendation, V 23.

Boley, H., Tabet, S., Wagner, G., 2001. Design rationale of RuleML: A markup language for semantic web rules., 2001.

Borst, W., Akkermans, J., Top, J., 1997. Engineering Ontologies. International Journal of Human-Computer Studies, pp.365- 406.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R., 2005. DL-Lite: Tractable description logics for ontologies. In Proceedings of the National Conference on Artificial Intelligence. Ch. 2. p.602.

Chamberlin, D., 2002. XQuery: An XML query language. IBM Syst. J., 41, pp.597--615.

Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I., 2000. The semantic web: The roles of XML and RDF. Internet Computing, IEEE, 4, pp.63--73.

Demoen, B. & DE LA BANDA, M.G., 2012. Introduction to the special issue on Prolog systems. Theory and Practice of Logic Programming, 12, pp.1--4.

Ertel, W., 2011. First-order Predicate Logic. Introduction to Artificial Intelligence, pp.31--55.

Fensel, D., Van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., 2001. OIL: An ontology infrastructure for the semantic web. Intelligent Systems, IEEE, 16, pp.38--45.

Gottlob, G. & Schwentick, T., 2011. Rewriting ontological queries into small nonrecursive Datalog programs. Proc. of DL, 745.

Gruber, T., 2008. What is an Ontology. Encyclopedia of Database Systems, 1.

Guarino, N., 2009. The ontological level: Revisiting 30 years of knowledge representation. Conceptual Modeling: Foundations and Applications, pp.52--67.

Haarslev, V. & Muller, R., 2001. RACER system description. Automated Reasoning, pp.701--705.

Horrocks, I. & Bechhofer, S., 2008. Semantic Web. Web Accessibility, pp.315--330.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B, Dean, M.2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission, 21, p.79.

Horrocks, I., Patel-schneider, P., Mcguinness, L., Welty. C., 2007. OWL: a Description Logic Based Ontology Language for the Semantic Web. In Proc of the 21st Int Conf on Automated Deduction. Ch. 39. pp.249--276.

Hunter, D., Rafter, J., Fawcett, J. Van Der Vlist, E., Ayers, D. Duckett, J., Watt, A. McKinnon, L.,2011. beginning XML. Wrox.

Huynh, D., Mazzocchi, S., Karger, D., 2007. Piggy bank: Experience the semantic web inside your web browser. Web Semantics: Science, Services and Agents on the World Wide Web, 5, pp.16--27.

Kahn, C.E., 1999. Standard Generalized Markup Language for self-defining structured reports. International journal of medical informatics, 53, pp.203--211.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C., Hendler, J., 2006. Swoop: A web ontology editing browser. Web Semantics: Science, Services and Agents on the World Wide Web, 4, pp.144--153.

Lassila, O., 2007. Programming Semantic Web Applications: A Synthesis of Knowledge Representation and Semi-Structured Data. phdthesis.

Lee, T.B., Hendler, J. & Lassila, O., 2001. The semantic web. Scientific American, 284, pp.34--43.

Malecha, G., Morrisett, G., Shinnar, A., Wisnesky, R., 2010. Toward a verified relational database management system., 2010.

McGuinness, D.L. & Patel-Schneider, P.F., 2003. From description logic provers to knowledge representation systems. In The description logic handbook. Cambridge University Press. pp.265--281.

McGuinness, D.L. & Van Harmelen, F., 2004. OWL web ontology language overview. W3C recommendation, 10, pp.2004--03.

Motik, B., Patel-Schneider, P.F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., 2009. Owl 2 web ontology language: Structural specification and functional-style syntax. W3C recommendation, 27, p.17.

Nguyen, D.T. & Nguyen, T.P., 2010. Pellet reasoner based evaluation of OVL (Ontology for Vietnamese Language).

OConnor, M. & Das, A., 2009. SQWRL: a Query Language for OWL. In 6th International Workshop on OWL: Experiences and Directions. Chantilly, VA, United States, 2009.

Oconnor, M., Knublauch, H., Tu, S., Musen, M., 2005. Writing rules for the semantic web using SWRL and Jess. Protege With Rules WS, Madrid.

Protege., 2012. Protege. [Online] Available at: http://protege.stanford.edu/ [Accessed 26 September 2012].

Schmidt-Schauß, M. & Smolka, G., 1991. Attributive concept descriptions with complements. Artificial intelligence, 48, pp.1--26.

Shearer, R., Motik, B. & Horrocks, I., 2008. HermiT: A highly-efficient OWL reasoner., 2008.

Sirin, E. & Parsia, B., 2007. Sparql-dl: Sparql query for owl-dl., 2007.

Sowa, J.F., 2006. Semantic networks. Encyclopedia of Cognitive Science.

Swift, T. & Warren, D.S., 2012. Xsb: Extending prolog with tabled logic programming. Theory and Practice of Logic Programming, 12, pp.157--187.

Tsarkov, D. & Horrocks, I., 2006. FaCT++ description logic reasoner: System description. Automated Reasoning, pp.292--297.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F., 2006. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web, 4, pp.14--28.

Valiente-Rocha, P. & Lozano-Tello, A., 2010. Ontology and SWRL-Based Learning Model for Home Automation Controlling., 2010.

Van Heijst, G., van der Spek, R., Kruizinga, E., 1997. Corporate memories as a tool for knowledge management. Expert Systems with Applications, 13, pp.41--54.

Vaughan-Nichols, S.J., 2010. Will HTML 5 restandardize the web? Computer, 43, pp.13--15.

Chapter 3

From Quantitative Spatial Operators to Qualitative Relationships

3.1 Introduction

The field of qualitative spatial relationships providing a spatial semantic language for analysing building information models is closely related to the concepts and the technologies developed in the area of Geographic Information Systems (GIS) (Bhattacharyya, 2009). Such systems preserve geographical data, such as the position and the shape, etc. and provide functionalities for the spatial analysis of this data. Nowadays, several Geographic Information Systems maintain 2D spatial relations between 2D objects. As a first attempt, Egenhofer presented a 2D query language based on the SQL one applied for the GIS context (Egenhofer, 1994). Later on, several other non-commercial and commercial languages saw the light, such as PSQ (Vijayvargiya, 2005), Spatial SQL (Bocher et al., 2008), GEOQL (Bhattacharyya et al., 2009), KGIS (Yilin et al., 2008) and TIGRIS (Goodrich et al., 2010) for the non-commercial, and PostGIS (Ramsey, 2005), Oracle Spatial (Kothuri, 2007) and Informix Geodetic Datablade for the commercial ones. Most of the above mentioned language and data bases (mainly GIS) comply with the normalization proposed by the OpenGIS as a shared interface to manage 3D spatial data (Consortium, 2012). In this context, Ozel (Ozel, 2005) emphasised the capability of geographic information systems to analyse and process building infrastructure. In this field, he concludes that although there is some robustness of the CAD system to model building in 2D/3D, it still suffers from disability of handling 3D spatial processing in a human understandable way. Based on such observations, the author opted to store the building component within a GIS data base, opening the door for further spatial processing. In fact, it is commonly agreed that the 3D spatial processing domain within any Building Information Model can take a step forward and produce more sophisticated analysis for building more complete BIM models. Until today, such 3D spatial processes are still not available within the formal GIS query languages or the building processing ones, where most of them still rely on 2D spatial processing. From a 3D point of view, and although the 3D CAD model enables storing and presenting simple 3D geometries (Bosche & Haas, 2008), the real processing capabilities of the spatial relation could be found within the research domain. Methods for modelling quantitative spatial relationships have been compiled in several surveys such as (Galton, 2009), (Randell et al., 1992). Actually, current models to define spatial relationships belong to two main categories – connection based (Randell et al., 1992), and intersection based

(Egenhofer & Herring, 1990). Both models fall to the same topological relationships for the simple 2D regions.
From 3D point of view, standard nomination to the basic topological relations is defined by the Open Geospatial Consortium (OGC) (Consortium, 2012). In fact, spatial operators available for spatial query language consist of 3D Topological operators (Meet, Overlap, Cover, Equal…) (Borrmann & Rank, 2008), 3D Metric operators (isFarFrom, isNearTo...) (Borrmann et al., 2009), 3D Directional operators (EastOf, isOn…) (Borrmann & Rank, 2009) and 3D Boolean operators (Difference, intersection, etc.) (Borrmann et al., 2006). As a main role, spatial operators are used to query the spatial relationship between two spatial entities. In this field, Zlatanova (Zlatanova et al., 2002) presented a review related to the target spatial relation. From the $\mathbb{R}^3$ space implementation point of view (Borrmann et al., 2009), the octree-based implementation (Meagher, 1982) and the Boundary Representation (Lee & Lee, 2001) approaches are used to define the spatial operators of a query language.

Through the current contribution, and dissimilar to the literature where main contributions relay on the Boundary Representation model (Haimes & Dannenhoffer, 2010), two main geometric data structures will be presented and compared for the spatial qualification task: the Constructive Solid Geometry (CSG) (Lohmüller, 2009) and the Nef polyhedra structure (Granados et al., 2003) and then linked with the OWL Semantic platform (Antoniou & Harmelen, 2009) via our 3D Spatial Qualification engine. Supported by such structures, this chapter and the next one aims at defining 3D spatial relations and mainly topological ones based on the 9 Intersection Model in $\mathbb{R}^3$ (Ellul & Haklay, 2009), and compute them with the Boolean operators defined through the studied geometric representation structures. This chapter addresses the definition of the quantitative operators and the associated data models. Actually, the 9-IM model is widely used to represent topologic relations in $\mathbb{R}^2$. These relations exist also in $\mathbb{R}^3$ with much more variation and complexity. Semantic considerations on these qualitative relationships are the main subject of the next chapter. 3D spatial relations will be qualified using description logic axioms (DLs) (Baader, 2009). This chapter provides two data models to compute qualitative relationships from quantitative operators.

This chapter is divided into 4 sections. Section 2 introduces the technical background on different spatial relations categories and especially topologic ones. Section 3 deals with the important elements of the quantitative relation implementation and the adopted geometric structure. Finally, section 4 discusses the current issues and concludes the chapter.

3.2 3D Spatial relationship background

In the field of quantitative approaches, the developed theory of the spatial relationship quantification is expected to provide answers to questions like how the spatial relationships can be formalized. What are the main geometric properties affecting the relations, and how can spatial relations be formally defined in terms of fundamental geometric properties? Concerning the 3D topological relationships, several relationship families have been proposed and discussed like the Simple Features family, the Region Connection Calculus (RCC8) and Egenhofer relation family, (Stocker & Sirin, 2009), trying to bring a formal definition of spatial relation between the different predefined geometric primitives. Likewise, a specific bridge between geometric properties and the spatial relation definition was recently built via the 4-Intersection Model, and the 9-Intersection one later on (Egenhofer et al., 1993). In the next part, a survey highlighting mainly the last discussed concepts will take place, added to an overview of the related works in the field of the qualitative spatial relationship. Metric relationships and directional have received less attention, mainly caused by the simplicity of their implementation since they rely more on object coordinates. Although they will be included in this chapter, the main focus will be on topological relationships.

3.2.1 Qualitative 3D topologic relationship overview

Spatial reasoning is a process that uses spatial theory and artificial intelligence to model and to analyse spatial relationships between objects. Concerning the 3D topological relationships, the standard models are composed by the Simple Feature Relations, The Egenhofer Relationships and the RCC8 Relationships (Stocker & Sirin, 2009). The Simple Features Relationships is based on the defined standard of OGC (Consortium, 2012) and are composed of the following relationships: Equals, Disjoint, Intersects, Touches, Within, Contains, Overlaps, and Crosses (Perry & Herring, 2010). The Egenhofer Relationships are composed of the following relationships: Equals, Disjoint, Meet, Overlap, Covers, Covered by, Inside and Contains (Egenhofer, 2010). Finally, the RCC8 Relations are presented by the following relationships: Equals, Disconnected, Externally connected, partially overlapping, Tangential proper part inverse, Tangential proper part, Non-tangential proper part, Non-tangential proper part inverse (Stocker &

Sirin, 2009). Table 3-1 summarizes the equivalences between Simple Feature family relations, the Egenhofer relation family and the RCC8 for closed and non-empty regions (Perry & Herring, 2010).

Simple Features	RCC8	Egenhofer
Equals	EQ	equal
Disjoint	DC	disjoint
intersects	¬ DC	¬ disjoint
Touches	EC	meet
Within	NTPP + TPP	inside + coveredBy
Contains	NTPPi + TPPi	contains + covers
Overlaps	PO	overlap

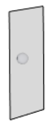
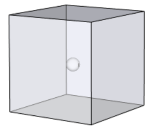
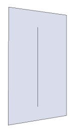
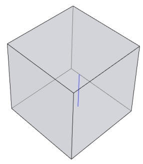
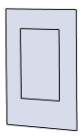
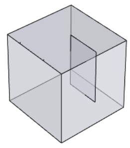
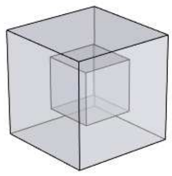
Table 3-1. The standard models of Qualitative Topologic Relationships

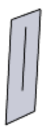
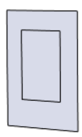
As concluded from Table 3-1, and although there are different expressions used between the presented families, a formal mapping between the different relations is totally possible, where the relation "*Touches*" within the Simple feature family defined by OGC is totally equivalent to the relation "Meet" inherited from the Egenhofer family. As a conclusion, a definition of the 'any relations' families can easily be propagated to the other ones. In the next, we tried to bring a definition to each one of the suggested relations where a more formal definition based on the Intersection Models will be presented later on in Table 3-2.

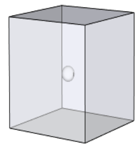
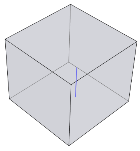
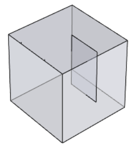
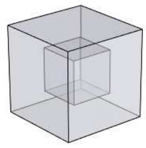
- **Cover**: Every point of the second geometry is a point of the first one.
- **Inside:** Every point of the second geometry is a point of the first one, and the interiors of the two geometries have at least one point in common.
- **Overlap:** The two geometries have at least one interior point in common.
- **Disjoint:** The two geometries have no point in common.
- **Meet:** The geometries have at least one point in common, but their interiors do not intersect.
- **Equals:** The two geometries have every point in common**.**

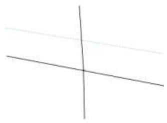
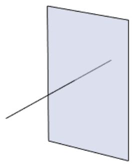
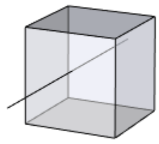
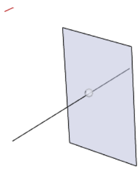
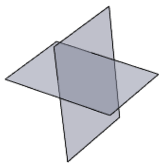
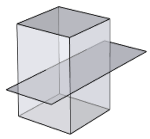
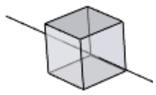
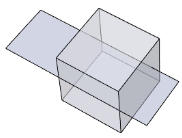
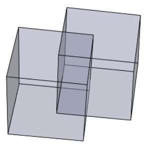
3.2.1.1 *3D Topological Relationships between primitives*

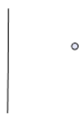
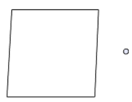
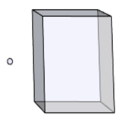
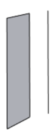
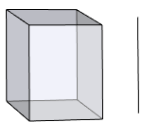
Always in the context of the explanation, each one of the above definitions of a topological relationship will be visualized between the eventual possible geometry which are point, line, surface and Body. The Egenhofer spatial relation in 3D will be adopted during the rest of the different thesis chapters.

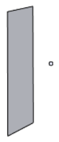
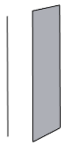
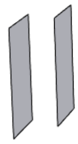
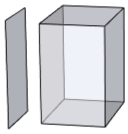
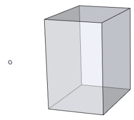
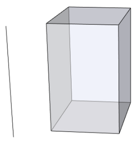
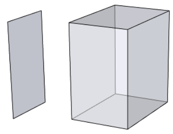
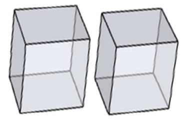
	Point	**Line**	**Surface**	**Body**	
Cover					**Point**
					**Line**
					**Surface**
					**Body**

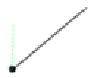
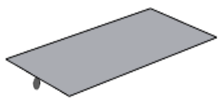
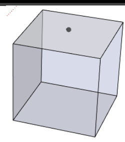
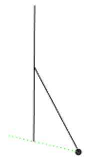
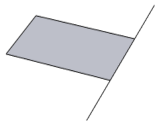
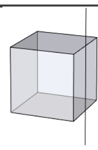
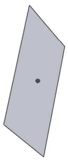
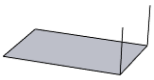
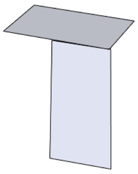
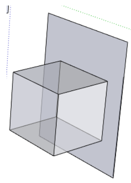
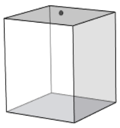
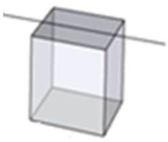
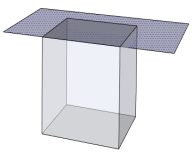
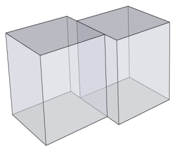
	Point	**Line**	**Surface**	**Body**	
Inside					**Point**
					Line
					**Surface**

					**Body**

	Point	**Line**	**Surface**	**Body**	
Overlaps					**Point**
					**Line**
					**Surface**
					**Body**

	Point	**Line**	**Surface**	**Body**	
Disjoint					**Point**
					**Line**

					**Surface**
					**Body**

	Point	**Line**	**Surface**	**Body**	
					**Point**
Meet					**Line**
					**Surface**
					**Body**

	Point	**Line**	**Surface**	**Body**	
					**Point**
					**Line**

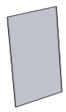
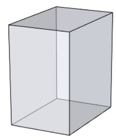
Equals		**Surface**
		**Body**

Table 3-2. Topological relationships between standards geometric primitives

In this field, the concepts of 'interior', 'boundary' and 'exterior' play a fundamental role on the forthcoming discussion on topological relationship qualification between geometries. Indeed, the different presented subspaces will, based on a predefined intersection model, help to identify if the target predicate relation between the candidate geometries is verified or not.

3.2.1.2 *The Qualification of 3D Topological Relationships*

The first step toward the qualification of topological relationships between geometries in 3D space was the development of the 4-Intersection Model based on 4-uplets recording whether the intersection between the interior and the boundary of geometries operand are empty or not. Later on, and mainly caused by the increasing of the geometries' dimensions and complexities, a new model providing more details than the 4-Intersection Model become mandatory. In fact, the need for more extensive models, which are able to compare different geometries from heterogeneous environment dimensions, becomes necessary. To do so, an extension of the 4-IM including the intersection with the geometry exterior was proposed. It allows the identification of more detailed relations, particularly when the candidate geometries are embedded in higher dimensions. Such a new model was called the 9-Intersection Model, where both models will be discussed in the next section.

4IM

Initially, Binary topological relationships between two objects, A and B, are defined in terms of the 4-Intersections Model (Egenhofer et al., 1993) of A's boundary (δA) and interior (A°) with the boundary (δB) and interior (B°) of B. By considering the value of empty (∅) and non-empty (¬∅) for the four intersection models, we can distinguish 2^4

topological relations, where just eight of these sixteen relationships can take place in the in $\mathbb{R}^2$, Table 3-3.

$$R_{(A,B)} = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \delta B \\ \delta A \cap B^\circ & \delta A \cap \delta B \end{pmatrix}$$

Table 3-3. The 4-IM matrix

Concerning the spatial relationships, they are usually formally defined through their intersection model filters. Table 3-4 presents an overview on the different masks related to each one of the Egenhofer Spatial Relations based on the predefined mask of Table 3-3, where the symbol (Ø) means an empty intersection between the correspondent geometries subspaces while ¬Ø means a non-empty one.

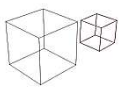
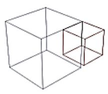
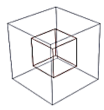
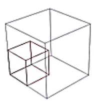
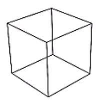
A disjoint B	A meets B	A contains B
		
$\begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \emptyset \end{pmatrix}$	$\begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset \end{pmatrix}$
A coveredByB	**A equals B**	**A overlaps B**
		
$\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$

Table 3-4. The 6 topological relationships between objects based on the 4-IM

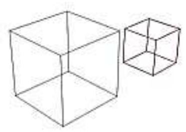
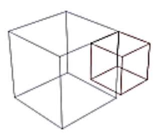
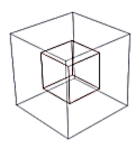
9IM

The 9 intersections model describes topological relationships which are represented by a 3x3 matrix (Zlatanova et al., 2004). The binary relationship R(A,B) between the two bodies is then identified by composing all the possible intersections of the six topological primitives, i.e. $A^\circ \cap B^\circ$, $\delta A \cap B$, $A^- \cap B^\circ$, $A^\circ \cap \delta B$, $\delta A \cap \delta B$, $A^- \cap \delta B$, $A^\circ \cap B^-$, $\delta A \cap B^-$, $A^- \cap B^-$, and qualifying empty ($\varnothing$) or non-empty ($\neg\varnothing$) intersections.

For example, if two objects have a common boundary, the intersection between the boundaries is non-empty, i.e. $\delta A \cap \delta B = \neg\emptyset$. if they have intersecting exteriors, then the intersection $A^{-} \cap B^{-}$ is not empty, i.e. $A^{-} \cap B^{-} = \neg\emptyset$. Table 3-5 shows the 9-IM matrices of the eight topological predicates defined by Egenhofer. One drawback of the 9-IM is that some topological configurations that are intuitively different result in the same 9-IM matrix while others that are intuitively identical are treated as being different. In addition, the number of detectable relationships between two objects thus increases to 2^9=512. The criticism is mostly about the fact that not all the relations are possible in reality, the intersections are not further investigated, and many object intersections are topologically equivalent (Zlatanova et al., 2002). The first problem is partially solved by the Dimensionally Extended 9-Intersection Model (DE-9IM) which also records the dimensionality of the intersection set. The DE-9IM forms the basis for the formal definitions of topological relationships in the OGC standard. However, in this contribution, no kind of optimization and wildcards were applied for the computation of a relation.

$$R_{(A,B)} = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \delta B & A^\circ \cap B^- \\ \delta A \cap B^\circ & \delta A \cap \delta B & \delta A \cap B^- \\ A^- \cap B^\circ & A^- \cap \delta B & A^- \cap B^- \end{pmatrix}$$

Table 3-5. The 9-IM matrix

A disjoint B	A meets B	A contains B
		
$\begin{pmatrix} \emptyset & \emptyset & * \\ \emptyset & \emptyset & * \\ * & * & * \end{pmatrix}$	$\begin{pmatrix} \emptyset & \neg\emptyset & * \\ * & * & * \\ * & * & * \end{pmatrix}\begin{pmatrix} \emptyset & * & * \\ \neg\emptyset & * & * \\ * & * & * \end{pmatrix}$ $\begin{pmatrix} \emptyset & * & * \\ * & \neg\emptyset & * \\ * & * & * \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & * & * \\ * & * & * \\ \emptyset & \emptyset & * \end{pmatrix}$

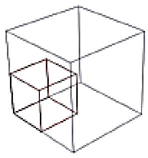
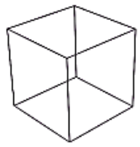
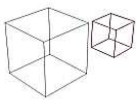
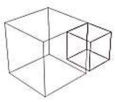
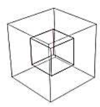
A coveredByB	A equals B	A overlaps B
		
$\begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ * & \neg\emptyset & \emptyset \\ \neg\emptyset & * & * \end{pmatrix}$	$\begin{pmatrix} * & \emptyset & \emptyset \\ \emptyset & * & \emptyset \\ \emptyset & \emptyset & * \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & * & \neg\emptyset \\ * & * & * \\ \neg\emptyset & * & * \end{pmatrix}$

Table 3-6. The 6 topological relationships between object based on the 9-IM

Referred to Egenhofer, eight topological relationships between simple bodies can be deduced. Table 3-6 represents the topology in $\mathbb{R}^2$ and $\mathbb{R}^3$ with the 9-IM matrixes. A basic body object in 3D space is a convex polyhedron that constructed by n (n>2) connected regions (r_1, r_2, …,r_n) where the interior must connect and does not contain holes. Each 9-intersection of the 3D spatial topological relationship can be represented with a code. For instance, if two spatial objects have no intersection points, then the binary code is [000 001 111] and the decimal number is 015. Consequently, the topological relationship “disjoint”, labelled “R015” in this case, represents the topological relations between these two objects. To do so, the following relation sequence is used: R_n=[$A^\circ \cap B^\circ, A^\circ \cap \delta B, A^\circ \cap B^-, \delta A \cap B^\circ, \delta A \cap \delta B, \delta A \cap B^-, A^- \cap B^\circ, A^- \cap \delta B, A^- \cap B^-$].

R015	R255	R505
		
$\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$
A disjoint B	A meets B	A contains B

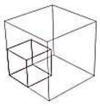
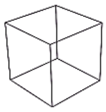
R311	R273	R511
		
$\begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
A coveredByB	A equals B	A overlaps B

Table 3-7. Decimal topological relationships between the two bodies with 9-IM in $\mathbb{R}^3$

A formal presentation is shown in more detail here. The corresponding 9-IM to each topological predicates for the Simple Features family, Table 3-8, and the RCC8 one, Table 3-9, is highlighted where F (false) is used in the matrices to denote an empty set ($\emptyset$), T (true) to denote an non-empty set ($\neg\emptyset$), and the wildcard (*) may be used at certain places in the matrix that are not relevant for the particular predicate, thereby solving the second of the aforementioned problems.

Relation Name	9IM Pattern
Equals	TFFFTFFFT
Disjoint	FF*FF****
Intersects	T******** *T******* ***T***** ****T****
Touches	FT******* F**T***** F***T****
Within	T*F**F***
Contains	T*****FF*
Overlaps	T*T***T**
Crosses	T*T******

Table 3-8. Simple Features family Topological Relations

Relation Name	9IM Pattern
Equals	TFFFTFFFT
Disconnected	FFTFFTTTT
externally connected	FFTFTTTTT

partially overlapping	TTTTTTTTT
tangential proper part inverse	TTTFTTFFT
tangential proper part	TFFTTFTTT
non-tangential proper part	TFFTFFTTT
non-tangential proper part inverse	TTTFFTFFT

Table 3-9. RCC8 Topological Relations

As seen from the previous tables, various collections of different families in term of spatial relationship exist nowadays (Simple family, Egenhofer family, RCC8 family) with informal notation of spatial relationships well explained in the human natural language. With the above detailed tables, we succeed to map the different definitions to the formal foundation of the 9-IM matrix. In order to compute a specific spatial relationship, nine Boolean operations are required. To reach this goal, the geometric model used to compute these Boolean operations should take into account the interior, the exterior and the boundaries in a highly precise and efficient manner, especially in three dimensions and with complex objects. Before going through the different issues and the adopted geometric structure in detail, a brief survey on the quantitative metric and directional approaches, deduced to qualify spatial metric and directional relationships respectively in 3D environment, will be highlighted.

3.2.2 Qualitative 3D Metric relationships overview

Actually, quantitative and qualitative representation describes the same domain. Only the symbols used are different where representative symbols are used in qualitative case while more numerical ones are used in the case of quantitative calculations. Moreover, it should be possible to transfer between both presentations. In this field, symbolic qualitative values should correspond to a range of quantitative ones. Such a context can be discussed well through the metric relationship modelling, where metric analysis refers to the distance from one object to another. In order to perform this analysis, the geometry coordinates are used. Figure 3-1 presents a detailed example on how the distance measurement is made. One of the suggested solutions for the three-dimensional Euclidean space consists of using the Pythagoras principal to conclude that the distance from P1 (x1,y1,z1) to P2 (x2,y2,z2) is $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$, (Deza, 2009).

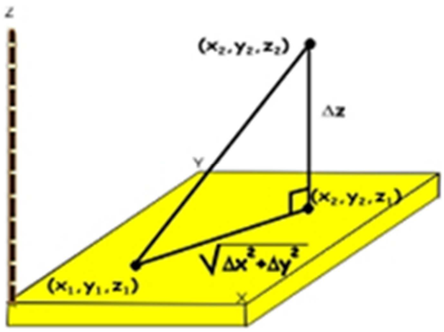


Figure 3-1. The metric Analysis

Actually, the mapping between quantitative and qualitative models is interval-based in this case. Such an interval base can be applied to the spatial metric relationship context where "*Very Far*" can be interpreted as a distance interval over 10 km in architectural context for example, Figure 3-2.

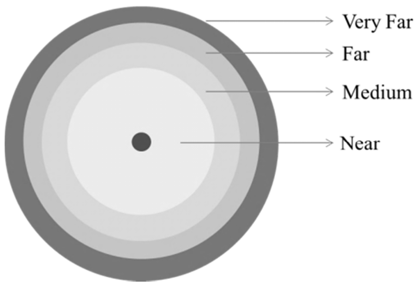


Figure 3-2. The qualitative metric distances

3.2.3 Qualitative 3D Directional relationships overview

As a definition, directional analysis refers to the position of an element with respect to another. We use operators that reflect the directional relationships between 3D spatial objects, such as north of, south of, east of, west of, above and below. Figure 3-3 shows the principal of the directional analysis.

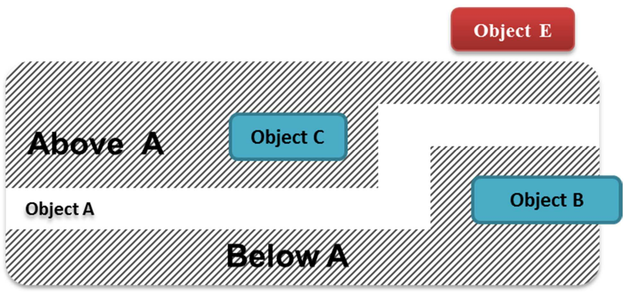


Figure 3-3. Directional relationships

Let A and B be two 3D spatial objects, ie. A, B ∈ $\mathbb{R}^3$ Where the 3D point a = (a_x,a_y,a_z) ∈ A and b = (b_x,b_y,b_z) ∈ B. For any point a of A and b of B verifying that $a_x \geq b_x$ while $a_y = b_y$ and $a_z = b_z$, the directional predicate "east_of" will be returned in this case allowing to stimulate that the geometry A is located on the east_of B (Borrmann et al., 2006). Based on this principle, the different used rules for the qualification of the directional relations will be deduced, Table 3-10.

Directional Relation	Used rules
A east_of B	$a_y = b_y \wedge a_z = b_z : a_x \geq b_x$
A west_of B	$a_y = b_y \wedge a_z = b_z : a_x \leq b_x$
A north_of B	$a_x = b_x \wedge a_z = b_z : a_y \geq b_y$
A south_of B	$a_x = b_x \wedge a_z = b_z : a_y \leq b_y$
A above B	$a_x = b_x \wedge a_y = b_y : a_z \geq b_z$
A beneath B	$a_x = b_x \wedge a_y = b_y : a_z \leq b_z$

Table 3-10. The directional relation and correspondent rules

3.2.4 Discussion

Regarding this section, directional and metric operators are relatively easy to implement. However, the issue is more complex when it arise the time to compute the topological relationships in 3D. In fact, not only the points have to be considered, but also the facets and its interiors, exteriors and its boundaries. The next section deals more specifically with this issue where we tried, through new data structure and semantic qualification, to optimize the calculation process to work with logical predicates without more recourse to the numerical model.

3.3 Formal specification of quantitative operators

The previous section focused on the modelling of 3D spatial relationships. To be able to produce qualitative relationships, the 3D processing require 3D representations of geometries. Nowadays, such a representation is divided into two main models, the boundary representation model (BenkHo, 2001) and the solid representation one (Lohmüller, 2009). While the boundary representation or B-rep is based on the two-dimensional surfaces of the model and supports shading and rendering, solid modelling manages the model element as real volumes and normally allows Boolean operations to be performed. Some solid modelling kernels make use of boundary representation to calculate and maintain solid and volume information as well, so there is a certain amount of overlap between them. In fact, each of the mentioned intersection models in the section 2 is based on the accepted definitions of the boundaries, interiors and exteriors for the basic geometry types which are taken into consideration. Therefore, the first step is the definition of the interior, the boundary and the exterior of the involved geometry types. The domain of geometric objects considered is those that are topologically closed. Up to now, mathematicians, CAD implementers and others have made various attempts in finding a definition for handling geometry that is computable, robust and mathematically correct at the same time. This led to constructs such as regularization and regularized operators, where all geometries are closed and each Boolean operation is followed by a closure operation as well.

3.3.1 The Octree based implementation

For a better understanding of the most suitable structure and methodology able to satisfy the above mentioned condition in 3D Space for the surveyed 9-IM purpose, Octree structure (Meagher, 1982) is used nowadays. Borrmann (Borrmann & Rank, 2009) suggests a technique based on the Octree representation of the spatial objects involved in the topological query based on 3D spatial operators. Each object is materialized via an individual Octree extracted from the object's boundary representation. Actually, the Octree is a "*space-dividing hierarchical tree data structure for the discretized representation of 3D volumetric geometry*" (Meagher, 1982). Every node in the above mentioned tree symbolizes a cubic cell with a specific black, white or grey colour. It signifies whether the octant stretches out inside, outside or on the boundary of the object respectively. In the tree structure, black and white octants present branch nodes where no more sub-division is required and therefore have no children. Grey octants present the uncertain information in the interior side and will therefore be divided into eight children where the re-combination of all the child octant cells must be equal to the volume of the parent cell. To cover dimensionally reduced entities DEM-9IM with the Octree algorithm, the author introduces the fourth colour: black/white. It presents areas where the object´s boundary is missed. Within the presented approach, the Octree creation is not achieved in advance, but coupled with the recursive algorithm and the root octants as parameters, the predicate to be tested, and an empty 9-IM matrix which will be successively filled during the algorithm execution. The presented algorithm is mainly based on the execution of different rules. These rules aim to fill the 9-IM matrix based on the octant pair colour combination. A white octant is part of the exterior of an operand, and a black octant is part of its interior. If a white octant of operand A occurs in the same place as a black octant of operand B, it means that the intersection between the exterior of A and the interior of B is non-empty. The author has fixed 12 positive and 9 negative rules where he supposes that a positive rule is applied when a certain colour combination takes place, and vice-versa for the negative rule. Positive rules lead to empty set entries in the matrix while negative rules to non-empty set entries. Supported by the presented algorithm, the 9-IM matrix is consecutively filled by applying the above mentioned rules for all octant pairs. Whenever a new entry in the 9IM is made, the matrix is matched up with predefined matrices for each topologic relation. When no complete matching with one of these matrices occurs, the recursion will continue with a further refinement i.e. octant pairs of the next level are created. Although the presented approach is considered as a reference for our thesis, we tried to overcome the limitations discussed by the author.

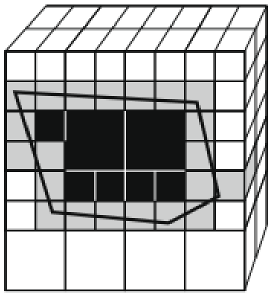


Figure 3-4. Cross-section through an octree (Borrmann & Rank, 2009)

As a first limitation, if the process achieves the maximum level defined by the user with no decision, the highest non-disproved predicate is returned in this case. Such an assumption may lead to an “incorrect” topological relation qualification especially in the case of complex geometries. In other scenario, the author suggests a way of viewing topology in a fuzzy way where the user can fix the resolution of relationships based on relative experience. Such an assumption can led to a non-precise topological predicate qualification with an error that will be propagated for further process.

In the next two sub-sections, we outline a solid representation structure (CSG) and a mixed one between solid and B-rep representation attempting to enrich a correct and efficient spatial relation qualification between 3D objects.

3.3.2 The Constructive Solid Geometry based implementation

As seen during this chapter, the technical implementation of the above formalized spatial and especially topologic model has to relay on the geometric representation of the spatial objects that can be involved in the topological model. In fact, B-rep and surface modelling strategy suffer from some weakness like the ambiguous and incomplete geometric description, the lack of topological information, tedious modelling process or awkward user interface. On the other hand, the solid representation materialized via the Constructive Solid Geometry (CSG), (Lohmüller, 2009) is defined by Friedrich A. Lohmüller (Corporation, 2006) and presents a technique used in solid modelling. Solid modelling consists of geometric data materialized by shape, size, location and topology data presented by the connectivity and the associativity of geometric elements. Within the constructive solid geometry, objects are represented as a combination of simpler solid objects, known as primitives where the different primitives are cube, cylinder, cone, torus, sphere etc. Once instances of these primitive shapes are created and positioned, a complete solid model is constructed by combining these "instances", using set specific logic operations where each primitive solid is assumed to be a set of points. A Boolean operation is performed on point sets and the result is a solid model.

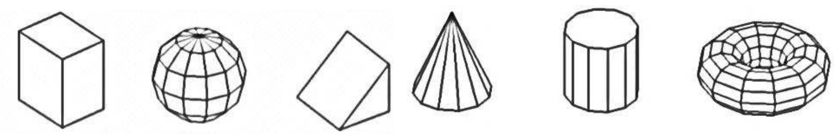


Figure 3-5. The CSG Primitives

Solid modelling is based on complete, valid and unambiguous geometric representation of physical objects since points in space can be classified as inside and outside where vertices, edges and faces are connected properly. As a result, there can only be one interpretation of the created object. A CSG object can be represented by a tree, where each leaf represents a primitive and each node a Boolean operation, Figure 3-6. Such a structure enables the modeller to create a complex surface or object by using Boolean operators such as union, intersection or difference to combine objects.

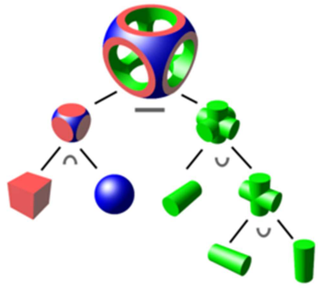


Figure 3-6. A CSG tree example (Wiki source)

There are only five CSG standard defined operations, which are materialized by the union, intersection, difference, inverse and Clipped_by. These methods return the solid result of the operation. The union operation aims to combine two or more objects to a new object. It results in the sum of all points in each of two defined sets and refers to the logical "OR". The *Difference* one subtracts from a basic object all subsequent objects from the other one, and results in the points in the source set minus the points common to a second set. It refers to the logical "NOT". The *intersection* operation results in an object which has an area that consists of the common one to the objects. Those points common to each of two defined sets can be defined through the logical expression "AND". The *inverse* one generates a new object's area containing everything but the first object's area. Finally, the *Clipped_by* operation looks similar to the intersection, but the shape of this new object is opened at the cutting surfaces. Boolean operations are intuitive to user and easy to use and understand. It provide for the rapid manipulation of large amounts of data. To apply it, the *UnBBoolean* tool (http://unbboolean.sourceforge.net) is used. It presents a 3D tool to model the different CSG primitives' structure and their correspondent Boolean set of operations; their algorithms are presented by (Laidlaw et al., 1986). As an example, the expression "*Solid1.subtract(Solid2)*" will return the new generated solid presenting the difference between Solid 1 and Solid 2 if the candidate sets share a common volume.

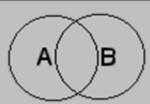
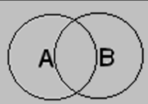
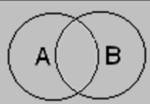
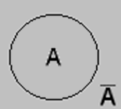
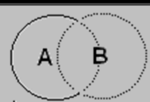
Union	Difference	Intersection	Inverse	Clipped_by
 = A ∪ B	 = A \ B	 = A ∩ B	 $\bar{A}$	 = A clipped_by {B inverse}

Table 3-11. Supported operation by CSG Structure

These Boolean operations return the solid resulting of the operation and are restricted to objects including a closed space. Actually, lines and planes are both objects which do not enclose a volume, as consequence, no possible CSG operations can be applied on them. As a conventional solution, a solid will be created from a line and plane by adding a small noise rate to the above mentioned geometry, always with respect to the fact that the added noise rate is always less than those related to the used instrument during the survey of real objects. In the next part, we will mainly focus on the implication of the CSG topologic operator within the 9-IM model to qualify the 3D spatial topologic relation directly.

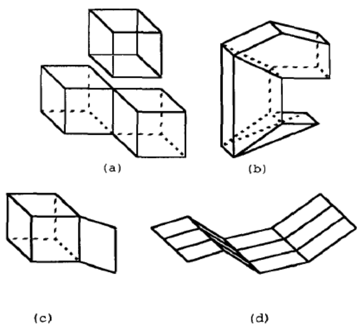


Figure 3-7. Objects (a) and (b) are valid, objects (c) and (d) are not

As an extension of the 4-IM, the 9-IM model is created by considering the location of each interior and boundary with respect to the other object's exterior. Therefore, the binary topological relation between two objects A and B in $\mathbb{R}^2$ is based upon the intersection of A's interior (A°), boundary (δA), and exterior (A-) with B's interior (B°), boundary (δB), and exterior (B-). A spatial region has simply three topologically distinct parts: the interior, boundary, and the exterior, where specifying any part of the first geometry will completely determine the region of the other parts. Based on this observation, it appears reasonable to assume that topological relationships between regions can be characterized by considering the intersections of any pair of parts mainly boundary/exterior or interior/exterior rather than only the boundary/interior intersections. To assess such alternatives, we have to determine whether the 4-Intersection based on the boundary / interior / intersections is equivalent to the one based on boundary / exterior or interior / exterior intersections. If so, the characterization of the topological relations would have to be the same in each case. Based on this assumption, we opt to use the 9-IM principle in a more optimal way, by reducing it to a four intersection model based on the interior / exterior of 3D geometry, Table 3-12, (Ben Hmida et al., 2012).

$$R_{(A,B)} = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap B^- \\ A^- \cap B^\circ & A^- \cap B^- \end{pmatrix}$$

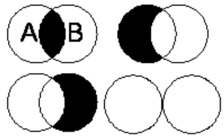


Table 3-12. The optimized 9-IM model (left) and the correspondent graphical representation (right)

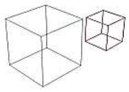
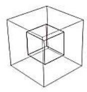
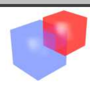
A disjoint B	A contains B	A overlaps B
$\begin{pmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$
		

Table 3-13. The optimized 9-IM matrix

Regarding the optimized 9-IM matrix, Table 3-12, only operators for intersection ($A \cap B$), interior (A° equivalent A), complement (A^- is equivalent to $\bar{A}$) are necessary. Once created, Table 3-13 presents the equivalent qualitative relations for each CSG operator. If one of these equations is false, the relation between the two objects cannot be verified. Finally, Table 3-14 presents the new suggested mask for 3D topological operations based on the interior and the exterior of each solid geometry. In parallel, Table 3-15 presents the relative CSG operation corresponding to each part of the mask.

$$R_{(A,B)} = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap B^- \\ A^- \cap B^\circ & A^- \cap B^- \end{pmatrix} \qquad R_{(A,B)} = \begin{pmatrix} A \cap B & A \backslash B \\ B \backslash A & \bar{A} \cap \bar{B} \end{pmatrix}$$

Table 3-14. The optimized 9-IM model (left) with the Equivalent mask using CSG operators (right)

Spatial relation	CSG operators
Disjoint	$(A \cap B = \emptyset) \wedge (A \backslash B = \neg\emptyset) \wedge (B \backslash A = \neg\emptyset) \wedge (\bar{A} \cap \bar{B} = \neg\emptyset)$
Contain	$(A \cap B = \neg\emptyset) \wedge (A \backslash B = \neg\emptyset) \wedge (B \backslash A = \emptyset) \wedge (\bar{A} \cap \bar{B} = \neg\emptyset)$
Overlaps	$(A \cap B = \neg\emptyset) \wedge (A \backslash B = \neg\emptyset) \wedge (B \backslash A = \neg\emptyset) \wedge (\bar{A} \cap \bar{B} = \neg\emptyset)$
CoveredBy	$(A \cap B = \neg\emptyset) \wedge (A \backslash B = \emptyset) \wedge (B \backslash A = \neg\emptyset) \wedge (\bar{A} \cap \bar{B} = \neg\emptyset)$
Equals	$(A \cap B = \neg\emptyset) \wedge (A \backslash B = \emptyset) \wedge (B \backslash A = \emptyset) \wedge (\bar{A} \cap \bar{B} = \emptyset)$

Table 3-15. Equivalent qualitative relations to qualitative CSG operator

Although the success of the presented solution based on CSG data structure in bringing a solution to the topological qualification problem (Ben Hmida et al., 2012), a refinement process still required to ensure a high performances and optimisation of the model to respond to the standard 9IM requirement and not the optimized one. Let´s first recall that the CSG solid is represented as a set-theoretic Boolean combination of primitive solid objects where the Boolean operations were not evaluated. Likewise, algorithms on such a CSG-tree first evaluate properties on the primitive objects and propagate the results using the tree structure.

Main limitations are issued from complex and non-uniform geometries where CSG is restricted by the selection of the primitive solids. In fact, only Boolean operations are allowed in the modelling process. Likewise, the range of shapes to be modelled is severely restricted, which makes it impossible to construct unusual shapes. To convert it to a Boundary representation, it requires a great deal of computation to derive the information on the boundary faces and edges, which is important for the interactive display and manipulation of solid geometries. In the case of 3D topological operators, some restrictions related to the ability to express some of the presented topological predicates is also noticed, where, and caused by the disability of the CSG structure to specify the 3D geometries' boundaries, a relation like "*touch*" will not be qualified. As a result of this research, a second main robust data structure is suggested for geometry representations avoiding the limitation of the Octree representation and the CSG one. In our implementation of Nef Polyhedra in 3D (Granados et al., 2003), we offer a B-rep data structure that is closed under Boolean operations and with all their generality starting from half space where we can work with union, intersection, difference, complement, interior, exterior, boundary, closure, and regularization operations.

3.3.3 The Selective Nef Complex Structure based implementation

Partitioning a 3D space into cells is a common theme of solid modelling and computational geometry, where defining the partitions of such a space onto different "cells" with its labelling is called a Selective Nef Complex (SNC). When the labels are Boolean, the complex is called Nef polyhedra (Granados et al., 2003). Such a structure was introduced by Nef (Nef, 1978) and is closed with respect to the elementary Boolean set operations as well as all topological operations.

A Nef polyhedron P is a subset of $\mathbb{R}^d$ *generated by applying set intersection and complement operations to a finite number of open halfspaces*. Thus, the class of Nef polyhedra is closed with respect to the Boolean set operations such as Union, Intersection and Difference. While implementing the Nef polyhedra in 3D, they offer a mixture of B-rep and CSG data structure that is closed under Boolean operations and with all their generality. Starting from halfspaces, it is possible to work with union, intersection, difference, complement, interior, exterior, boundary, closure, and all the regularization operators. In fact, these operators work with two data structures. A first one represents the local neighbourhoods of vertices, which already has a complete description. A second data structure is used to connect these neighbourhoods to a global data structure with edges, facets, and volumes. In the following figure, an example of the difference between two Nef Polyhedra is depicted in $\mathbb{R}^2$ about the A\B resulted polyhedron, where the edges and vertices in gray colour are not of part of the new polyhedron, Figure 3-8.

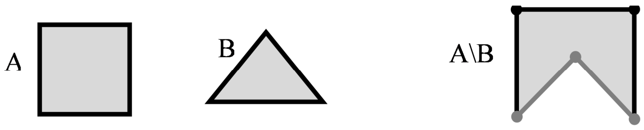


Figure 3-8. An example of the difference between two Nef polyhedra

The theory of Nef polyhedra has been developed for arbitrary dimensions. A Nef-polyhedron in dimension d is a point set $P \subseteq \mathbb{R}^d$ generated from a finite number of open half spaces by set complement and set intersection operations. Unbounded Nef polyhedra are problematic. In order to transform unbounded Nef polyhedra to a bounded one, they are intersected with a bounding cubical volume of size $[-R, R]^3$. R is a symbolical unspecified value, which is finite but larger than all coordinate values that may occur in the bounded part of the polyhedron. This box or frame is called the infimaximal box. The R-sets are topological polyhedral and may be viewed intuitively as curved polyhedra with well-behaved boundaries. The different Boolean set operations R-sets are not algebraically closed, Figure 3-9, but they are closed under the so-called regularized set intersection, union, and difference, denoted $\cap^*$, $\cup^*$, $-^*$, which are modified versions of their conventional counter parts (Boigelot et al., 2012). Boigelot proposes to use regularized set operations (Boigelot et al., 2012) where A set is regular, if it is equal to the closure of its interior. A regularized set operation is defined as the standard set operation followed by a regularization of the result. Regularized sets are closed under regularized set operations. The implementation in (Granados et al., 2003) has provided the regularization operation as a shortcut for the consecutive execution of the interior and the closure operations.

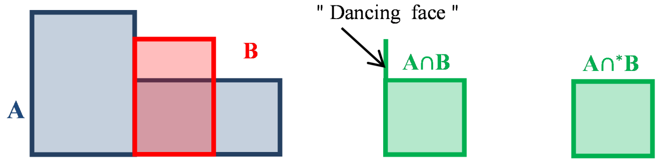


Figure 3-9. The regularized intersection of two r-sets

3D Nef polyhedra are closed under all Boolean set operations (closure, interior, exterior, boundary...) where its implementation in (Granados et al., 2003) provided functions and operators for the most common ones: mainly complement union, difference, intersection and symmetric difference. Such an implementation is materialized through the Computational Geometry Algorithms (CGAL) library (http://www.cgal.org). It presents an Open Source C++ software library where predefined methods can be directly used to apply such a Boolean operation. It provides the topological operations with interior, closure and boundary. The interior operator deselects all boundary items. The boundary operator deselects all volumes, and the closure operator selects all boundary items. Same, such a structure provides more complex operations like the Complement, Union, Difference, Intersection and Symmetric difference. Table 3-16 summarize the different available operation on the 3D Nef Polyhedra, where the syntax of each operator and the correspondent used CGAL function are highlighted.

Operators	Syntax	CGAL Used Methods
Complement	A^C	Nef_polyhedron Res = A.complement();
Union	$A \cup B$	Nef_polyhedron Res = (A + B)
Difference	$A \backslash B$	Nef_polyhedron Res = (A - B)
Intersection	$A \cap B$	Nef_polyhedron Res = (A * B)
Symmetric difference	$A \Delta B$	Nef_polyhedron Res = (A ^ B)
Interior	I(A)	Nef_polyhedron Res = A.interior()
Closure	C(A)	Nef_polyhedron Res = A.closure()
Boundary	B(A)	Nef_polyhedron Res = A.boundary()

Table 3-16. The set of binary and unary operators

Table 3-16 presents an overview of the available SNC Boolean operators. Regarding the Table 3-5 about the 9-IM matrix, only the operators about intersection ($A \cap B$), interior ($A°$ equivalent I(A)), boundary (δA is equivalent to B(A)) and complement (A^- is equivalent to $I(A)^C$ that we will denote E(A)) are necessary. Consequently, the following 9-IM matrix is deduced, Table 3-17, (Ben Hmida et al., 2012).

$$R_{(A,B)} = \begin{pmatrix} \mathrm{I}(\mathrm{A}) \cap \mathrm{I}(B) & I(A) \cap \mathrm{B}(B) & I(A) \cap E(B) \\ B(A) \cap \mathrm{I}(B) & B(A) \cap \mathrm{B}(B) & B(A) \cap E(B) \\ E(A) \cap \mathrm{I}(B) & E(A) \cap B(B) & E(A) \cap E(B) \end{pmatrix}$$

Table 3-17. The updated 9-IM matrix

The algorithm that computes a topological relationship consists for each cell of the updated 9-IM matrix to compute the nine equations and compare the result (false/0 or true/1) with each cell of the corresponding relation in the matrix. If the result of the nine updated equations is conform to the expected results, and then the relation is true. Otherwise the relation is false, Table 3-17.

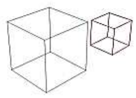
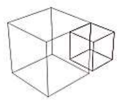
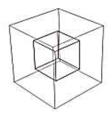
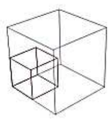
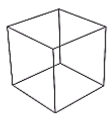
A disjoint B	A meets B	A contains B
		
$\begin{pmatrix} \emptyset & \emptyset & * \\ \emptyset & \emptyset & * \\ * & * & * \end{pmatrix}$	$\begin{pmatrix} \emptyset & \neg\emptyset & * \\ * & * & * \\ * & * & * \end{pmatrix}\begin{pmatrix} \emptyset & * & * \\ \neg\emptyset & * & * \\ * & * & * \end{pmatrix}\begin{pmatrix} \emptyset & * & * \\ * & \neg\emptyset & * \\ * & * & * \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & * & * \\ * & * & * \\ \emptyset & \emptyset & * \end{pmatrix}$
A coveredBy B	**A equals B**	**A overlaps B**
		
$\begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ * & \neg\emptyset & \emptyset \\ \neg\emptyset & * & * \end{pmatrix}$	$\begin{pmatrix} * & \emptyset & \emptyset \\ \emptyset & * & \emptyset \\ \emptyset & \emptyset & * \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & * & \neg\emptyset \\ * & * & * \\ \neg\emptyset & * & * \end{pmatrix}$

Table 3-18. The updated 9-IM matrix

In more details, Table 3-19 is an example for the disjoint relationship created based on the updated 9-IM matrix presented through the SNC Boolean operators, Table 3-17, if one of these equations is false, then the relation between the two objects does not exist.

$$R_{\mathrm{disjoint}(A,B)} = \begin{pmatrix} \mathrm{I}(\mathrm{A}) \cap \mathrm{I}(B) = \emptyset & I(A) \cap \mathrm{B}(B) = \emptyset & I(A) \cap E(B) = \neg\,\emptyset \\ B(A) \cap \mathrm{B}(B) = \emptyset & B(A) \cap \mathrm{B}(B) = \emptyset & \delta A \cap E(B) = \neg\,\emptyset \\ E(A) \cap \mathrm{I}(B) = \neg\,\emptyset & E(A) \cap B(B) = \neg\,\emptyset & E(A) \cap E(B) = \neg\,\emptyset \end{pmatrix}$$

Table 3-19. An example of the disjoint relationship

The main presented solution for the 3D spatial topological relation qualification is based on the discussed 3D Nef Polyhedron structure. The calculation of these relations is based on the definition of 3D Nef Polyhedra and can be generated automatically from standard Polyhedron respecting the validity conditions of the 3D Nef Polyhedra ones. Compared to the literature, the provided solution suggests an optimal data structure able to discriminate with high efficiency the different object regions without further complex processing. Such a process is being developed using the library CGAL (http://www.cgal.org) where the original 9 IM matrix is used and verified. Moreover, main limitations related to open polyhedron or non-convex ones have to be highlighted, where the correction process of such geometry is not always evident.

3.4 Conclusion and Discussion

Currently, 3D spatial reasoning with its three derivations (3D Topologic reasoning, 3D metric reasoning and 3D directional reasoning) is an important part of Artificial Intelligence where existing approaches in this field are numerical based. In this chapter, we have suggested a new method for 3D spatial relation qualification, taking onto account the geometry structure and its ability to specify the internal, the external and the boundaries of each object. In parallel, we have presented, through the adopted geometric structure and the 9-IM a formal logic rules able to satisfy the intersection model in each case to be mapped later on to the qualitative level. To do, we have first introduced an overview about the different 3D spatial component model, where we have put the light on the 3D topologic relation based on the OCG definition. In section three, we recommended the CSG geometric representation and later on the Nef Polyhedra one according to our vision for the 3D spatial concepts, where we tried to prove our choice and defend it as it is more feasible and exact than other models based on the literature synthesis.

In a more generic critical point of view, the founded approach qualifying spatial relationship and ensuring further processing of engineering problems seem to be unsatisfactory for mainly two reasons. First, existing approaches mainly focus on the low level quantification where no work combines the physical quantification of spatial relationships between geometries with the logical qualification of such relations semantically. Second, such a solution must be as generic as possible where the most efficient and precise geometric data structure has to be used for the spatial qualification model. To overcome these limitations, the next chapter will focus on the integration of semantics in the pre-defined 3D spatial relations. It will make an attempt to emphasize the possibility to combine 3D spatial technology and the above mentioned 3D spatial reasoning by the integration of such a technology in the semantic web framework. Otherwise, the suggested solution moves outside from the range of the data interoperability while presenting the concepts and make an effort to utilize others areas of semantic web technology. The basic capacity of knowledge processing provides the semantic Web with the capabilities to process the semantics of the information through close collaboration with the machine. In fact, it makes not only the perceptive of 3D spatial data easier for interoperability among different data sources, but mainly provides helpful knowledge able to enrich the knowledge base with new knowledge. Such a process will primarily help to understand spatial data and relationships in a better way.

References

Antoniou, G. & Harmelen, F., 2009. Web ontology language: OWL. Handbook on ontologies, pp.91--110.

Baader, F., 2009. Description logics. Reasoning Web. Semantic Technologies for Information Systems, pp.1--39.

BenkHo, P., Martin, R.R., Varady, T., 2001. Algorithms for reverse engineering boundary representation models. Computer-Aided Design, 33, pp.839--851.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2012. From 9-IM Topological Operators to Qualitative Spatial Relations using 3D Selective Nef Complexes and Logic Rules for bodies. In KEOD 2012, 4th International Conference on Knowledge Engineering and Ontology Development. Barcelona, Spain, 2012.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2012. From Quantitative Spatial Operator to Qualitative Spatial Relation Using Constructive Solid Geometry, Logic Rules and Optimized 9-IM Model. IEEE International Conference on Computer Science and Automation Engineering (CSAE 2012), 3, pp.453- 458.

Bhattacharyya, D., Chakraborty, D., Bandyopadhyay, S.K., Kim, T., 2009. GIS and Mobile Query. In IEEE Second International Conference on Computer and Electrical Engineering, ICCEE'09. pp.648--651.

Bocher, E., Leduc, T., Moreau, G., Gonzailez Cortas, F., 2008. GDMS: An abstraction layer to enhance Spatial Data Infrastructures usability. 11th AGILE International Conference on Geographic on Information Science - AGILE'2008.

Boigelot, B., Brusten, J., Degbomont, J.F., 2012. Automata-Based Symbolic Representations of Polyhedra. Language and Automata Theory and Applications, pp.3--20.

Borrmann, A. & Rank, E., 2008. Topological operators in a 3D spatial query language for building information models. In Proc. of the 12th Int. Conf. on Computing in Civil and Building Engineering (ICCCBE-XII). Beijing, China.

Borrmann, A. & Rank, E., 2009. Specification and implementation of directional operators in a 3D spatial query language for building information models. Advanced Engineering Informatics, 23, pp.32--44.

Borrmann, A. & Rank, E., 2009. Topological analysis of 3D building models using a spatial query language. Advanced Engineering Informatics, 23, pp.370--385.

Borrmann, A., Schraufstetter, S., Rank, E., 2009. Implementing metric operators of a spatial query language for 3D building models: octree and B-Rep approaches. Journal of Computing in Civil Engineering, 23, p.34.

Borrmann, A., Van Treeck, C., Rank, E., 2006. Towards a 3D spatial query language for building information models. Proc. Joint Int. Conf. of Computing and Decision Making in Civil and Building Engineering.

Bosche, F. & Haas, C., 2008. Automated retrieval of 3D CAD model objects in construction range images. Automation in Construction, 17, pp.499--512.

Consortium, O.G., 2012. OGC Reference Model. [Online] Available at: http://www.opengeospatial.org/standards/orm.

Corporation, L., 2006. Leadwerks Engine. [Online] Available at: http://www.leadwerks.com/files/csg.pdf.

Deza, M.M. & Deza, E., 2009. Encyclopedia of distances. Springer Verlag.

Egenhofer, M.J., 1994. Spatial SQL: A query and presentation language. Knowledge and Data Engineering, IEEE Transactions on, 6, pp.86--95.

Egenhofer, M.J., 2010. Qualitative Spatial-Relation Reasoning for Design. NSF International Workshop on Studying Visual and Spatial Reasoning for Design Creativity, Aixen-Provence, 2010.

Egenhofer, M.J. & Herring, J., 1990. A mathematical framework for the definition of topological relationships. Fourth International Symposium on Spatial Data Handling. Zurich, Switzerland. pp 803--813.

Egenhofer, M.J., Sharma, J. & Mark, D.M., 1993. A critical comparison of the 4-intersection and 9-intersection models for spatial relations: formal analysis. In Autocarto-Conference. ASPRS American Society for Photogrammetry. PP.1--1.

Ellul, C. & Haklay, M.M., 2009. Using a B-rep structure to query 9-intersection topological relationships in 3D GIS--reviewing the approach and improving performance. 3D Geo-information Sciences, pp.127--151.

Ericson, C., 2005. Real-time collision detection. Morgan Kaufmann.

Galton, A., 2009. Spatial and temporal knowledge representation. Earth Science Informatics, 2, pp.169--187.

Goodrich, J.M., Miquelle, D.G., Smirnov, E.N., Kerley, L.L., Quigley, H.B. Hornocker, M.G., 2010. Spatial structure of Amur (Siberian) tigers (Panthera tigris altaica) on Sikhote-Alin Biosphere Zapovednik, Russia. Journal of Mammalogy, 91, pp.737--748.

Granados, M., Hachenberger, P., Hert, S., Kettner, L., Mehlhorn, K., Seel, M., 2003. Boolean operations on 3D selective Nef complexes: Data structure, algorithms, and implementation. Algorithms-ESA 2003, pp.654--666.

Haimes, R. & Dannenhoffer, J., 2010. Control of Boundary Representation Topology in Multidisciplinary Analysis and Design. In 48th Aerospace Sciences Meeting.

Kothuri, R.V., Godfrind, A., Beinat, E., 2007. Pro oracle spatial for oracle database 11g. Apress.

Laidlaw, D.H., Trumbore, W.B., Hughes, J.F., 1986. Constructive solid geometry for polyhedral objects. In Computer Graphics., 1986.

Lee, S.H. & Lee, K., 2001. Partial entity structure: a compact non-manifold boundary representation based on partial topological entities. In Proceedings of the sixth ACM symposium on Solid modeling and applications., 2001. ACM.

Lohmüller, F.A., 2009. CSG - Constructive Solid Geometry. [Online] Available at: http://www.f-lohmueller.de/pov_tut/csg/povcsg1e.htm [Accessed 20.09.2012].

Meagher, D., 1982. Geometric modeling using octree encoding. Computer Graphics and Image Processing, 19, pp.129--147.

Nef, W., 1978. Beitrage zur Theorie der Polyeder: mit Anwendungen in der Computergraphik. Lang.

Ozel, F., 2005. Confluence of Building Information for Design, Construction and Management of Buildings. International Journal of Architectural Computing, 3, pp.373--390.

Perry, M. & Herring, J., 2010. GeoSPARQL--A geographic query language for RDF data. A proposal for an OGC Draft Candidate Standard. Open Geospatial Consortium, 27.

Ramsey, P., 2005. PostGIS manual. Refractions Research Inc.

Randell, D.A., Cui, Z. & Cohn, A.G., 1992. A spatial logic based on regions and connection. Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, 92, pp.165--176.

Stocker, M. & Sirin, E., 2009. Pelletspatial: A hybrid rcc-8 and rdf/owl reasoning and query engine., 2009. Proc. OWL: Experiences and Directions (OWLED), 6th International Workshop.

Chiticariu, L., Tan, W.C., Vijayvargiya, G., 2005. DBNotes: a post-it system for relational databases based on provenance., 2005. ACM Press.

Yilin, Y., Ling, Y., Zhe, W., 2008. Continuous improvement of public project management performance based on project governance. In Networking and Mobile Computing, WiCOM'08. 4th International Conference on Wireless Communications. pp.1--6.

Zlatanova, S., Rahman, A.A., Shi, W., 2002. Topology for 3D spatial objects. In International Symposium and Exhibition on Geoinformation., 2002.

Zlatanova, S., Rahman, A.A., Shi, W., 2004. Topological models and frameworks for 3D spatial objects. Computers & Geosciences, 30, pp.419--428.

Chapter 4

Integration of 3D Spatial Processing with Knowledge Processing

4.1 Introduction

The last chapter focused on the 3D processing of spatial relationships materialized by different spatial operators. As a continuation, this section will deal with the semantic aspects of these relationships. In fact, the semantic definition of objects in an OWL ontology knowledge base (Antoniou & Harmelen, 2009) enables the storage of spatial relationships. In this proposition, the description logic rules (Baader & Sattler, 2001) and the OWL-DL language of the Semantic Web (McGuinness & Van Harmelen, 2004) are used to model the semantics of objects and their spatial relationships. These relationships are computed from quantitative data stored in the ontology. With its help, standard reasoning tools can be applied on the created 3D spatial Qualification (3DSQ) data model and its spatial relationships. The Semantic aspects of the 3DSQ approach will be discussed in this chapter. To do so, two issues had to be resolved.

1. The precise computation of spatial relationships on 3D quantitative data.
2. The definition of a qualitative rule language for the access and the management of spatial information.

Concerning the first point, and as discussed during the last chapter, the precise computation of spatial relationships in our contribution relays mainly on the Nef polyhedra structure (Granados et al., 2003). Supported by such a data representation, the presented work has defined the spatial relation and mainly the topological one, based on the 9-Intersection Model in $\mathbb{R}^3$ (Ellul & Haklay, 2009), and computed them with the Boolean operators defined by the studied geometric representation structures. Concerning the second point in relation with the definition of a rule language, the qualitative spatial operators are implemented using built-ins based the Semantic Web Rules Languages (SWRL) (Horrocks et al., 2004) which enables the definition of logic programs based on Horn-like clauses. This language is designed to perform logical programs on Ontology Web Language (OWL) (Gruber, 2008). Consequently, the results of these 3D spatial operators may enrich the ontology with spatial relations between the different objects.

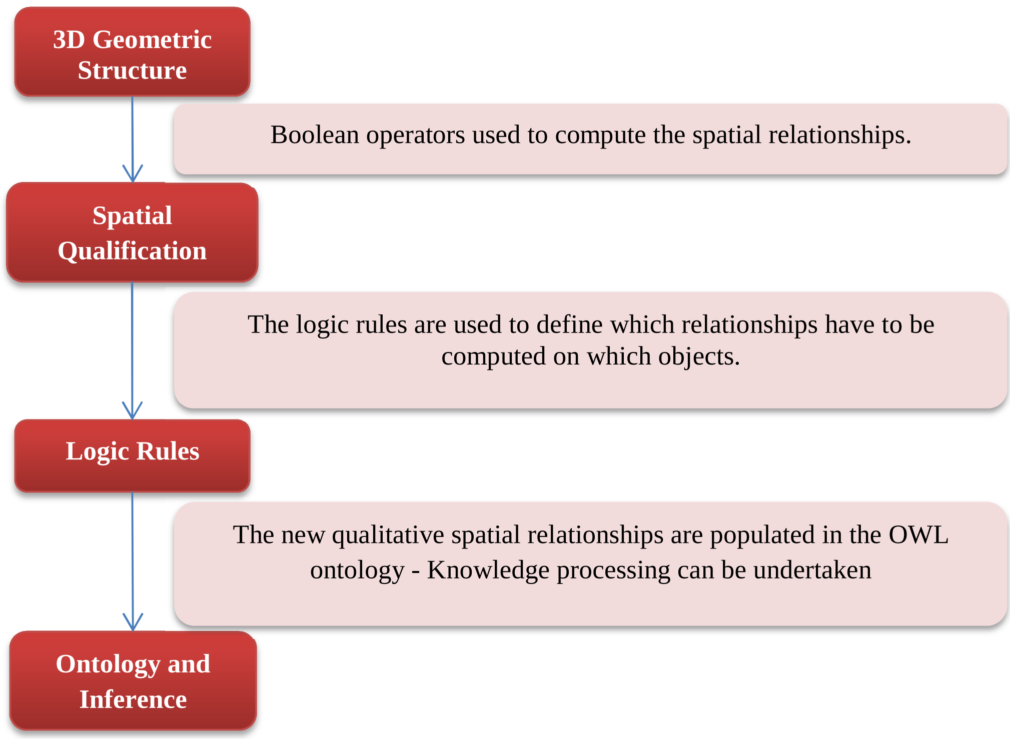


Figure 4-1. From quantitative operators to qualitative relationships

Figure 4-1 depicts the process sequence for the enrichment of an OWL ontology containing 3D objects. This ontology is populated with data from different resources. Then the spatial relationships are computed using semantic rules. These rules make it possible to process queries on the ontology knowledge base. The inference process on these relationships makes a step forward to infer new knowledge out. The logic rules are based on new 3D spatial built-ins defined based on the qualification engine requirement, and computed for each object´s relationships using 3D Nef polyhedron (Granados et al., 2003) and its respective Boolean operators. The following example is a SWRL rule that uses the “swrl_topo:overlaps” built-ins which select all the 3D models of buildings and railways which overlaps.

$$Buiding(?b) \wedge Railway(?r) \wedge swrl\_topo{:}\ overlaps(?b, ?r) \rightarrow RailStation(?b) \qquad (14)$$

This chapter begins with the presentation of the spatial technologies integration onto the Semantic Web Stack. Section 3 introduces the integration of 3D spatial processing with the knowledge processing in an OWL ontology domain from one side and highlights the important elements of the Built-In implementation from another side. Section 4 highlights the semantic translation engine and the rule execution process. Finally, Section 5 concludes the chapter.

4.2 Integration of 3D spatial processing within the semantic web stack

Modelling spatial information on the web is an important field of research, where huge amounts of spatial information already exist in an unstructured way. Such spatial reasoning can take a step forward and support building decisions for a variety of domains, mainly the weather domain, road mapping, biomedical, etc. Nowadays, several approaches investigate the problematic of qualitative spatial knowledge representation on the Semantic Web. In fact, OWL-DL provides some of the expressive powers required for the representation of spatial regions and their relationships. However, a direct representation is far from intuitive.

Recently, several languages representing relations between spatial regions were developed. Among these formalisms for qualitative spatial reasoning, is the Region Connection Calculus (RCC8) (Li & Ying, 2003), which introduces a set of eight basic relationships between spatial regions in 2D, and has received particular attention. In this field, several authors have focused on the RCC qualitative spatial relation instead of the 9IM quantitative one, since it is based on logic theory, while the 9IM is based on elementary geometries. RCC describes regions in an abstract way, where three main regions are taken into consideration: the Closure region, the Boundary region and the Interior region. The Closure region C presents the smallest non-opened space containing the region R and can be expressed as $\mathrm{Closure} \equiv \mathrm{Interior} \sqcup \mathrm{Boundary}$. Grutter et al (Grutter & Bauer-Messmer, 2007) suggested a combination of OWL with RCC for Spatio-Terminological Reasoning on Environmental Data. The author outlines a translation of the RCC-8 calculus into OWL-DL language (Baader et al., 2005) by adapting some of the known results of qualitative spatial formalisms into a logic model. As an example, a disconnection relation (DS) between two regions R1 and R2 can be modelled as $DC(R1, R2) \equiv \neg C(R1, R2)$ where C means the *Connection* relation. To encode the RCC-8, it is necessary to extend the Web Ontology Language with the main ability to define reflexive roles, which require an extension of the DLs language syntax from one side and the integration of its logic onto the existing OWL reasoners. The same discussed DC relation can then be expressed with the OWL DL ontology by the following expression: $DC(R1, R2) \equiv R1 \sqsubseteq \neg R2$. Stocker et al (Stocker & Sirin, 2009) presented "*PelletSpatial*" as a qualitative spatial reasoning engine implemented on top of Pellet (Parsia & Sirin, 2004). Pellet Spatial provides consistency checking and query answering over spatial data represented with the Region Connection Calculus (RCC). It supports all RCC-8 relations as well as standard RDF/OWL semantic relations, both represented in RDF/OWL language.

Qualitative RCC approaches aim at providing calculus enabling machines to define, process and reason on spatial entities without going back to traditional methods like the 9IM for example. The different provided definitions to include the RCC relation in the semantic web framework aims at supporting spatial analysis, reasoning and queries on abstract spatial data. In this era, the different extensions of several OWL languages have made attempts to present data in a spatially correct way. Such contributions have resolved huge issues within the qualitative spatial domain on the Web, where inferences and DLs constraints on the defined classes will therefore be used for reasoning in an abstract-oriented way. It is therefore likely to query for regions already populated within the knowledge base structure spatially. For instance, such a contribution has held selective queries combining spatial and non-spatial set of axioms.

In fact, we argue that providing an encoding of qualitative spatial relations into OWL-DL is one of the keys for integrating spatial and reasoning in OWL-based tools. Moreover, such an encoding mix the different semantic layers from one side and lack efficiency of suggesting real solutions for real engineering problems on the other side, since it was designed to reason on qualitative knowledge without building any bridges with the quantitative one. Currently, we see the Semantic Web as an extension of the existing World Wide Web technology (Lederer et al., 2000), (Blumauer & Pellegrini, 2006) where information is provided with its semantics for a better cooperation between human and machines. Such a challenge can be achieved by increasing the existing layout information with semantics by adding descriptive terms defined in ontologies to web content. In this field, ontologies have a crucial role in conceptualizing a domain and enabling Web-based knowledge processing to be shared and reused between applications.

This section mainly focuses on the extension of the semantic web platform, with the functionalities of the 3D qualitative spatial knowledge integration, including a real analysis of 3D geometries. The integration of such a spatial relationship enables the properties to be used directly in ontology structure to test the existence of asserted binary spatial relationships between 3D objects. As a result, spatial queries and rules engine for spatial relation are created. To do so, a top level ontology with spatial relationships is defined enabling the adjustment of an existing ontology in order to be able to process spatial knowledge through the above mentioned spatial technology. As a result, the Semantic Web layer is adjusted with the newly created layer containing 3D spatial information as shown in Figure 4-2.

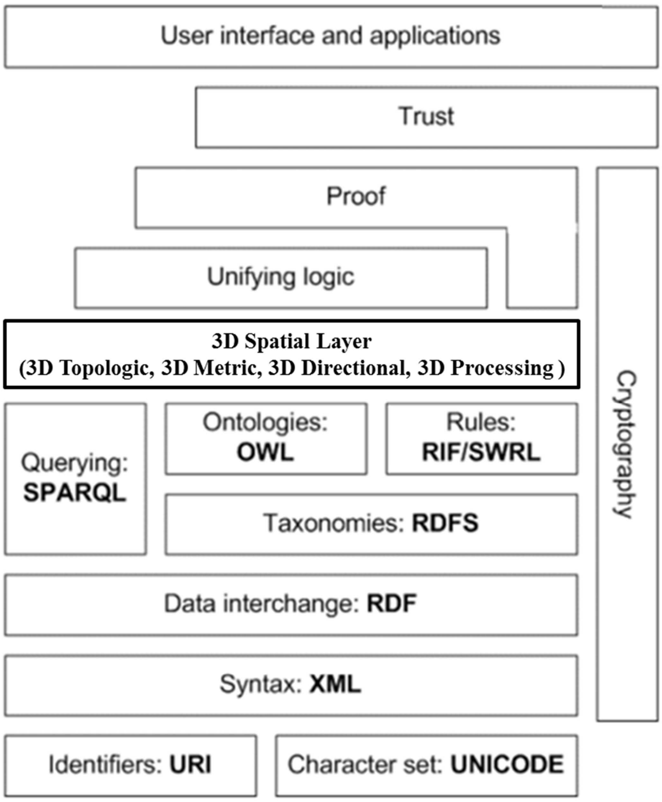


Figure 4-2. The Semantic Web Stack and the added 3D Spatial Layer

In the next paragraph, we show in detail the suggested formal solution for the arrangement of such a layer. This layer, while using the standard syntax of OWL/RDF, can perform spatial knowledge and operations through SWRL Built-Ins or infer rules through standards as SWRL. The integration process of the 3D spatial operations and knowledge onto the Semantic Web stack is controlled via new kinds of Built-ins for semantic rules. In fact, the created built-Ins enable the process of rules with 3D spatial operations related to semantic data.

4.3 The 3D spatial top level ontology and built-ins

The created top level ontology related to the 3D spatial relation qualification tasks serves as a foundation knowledge base where qualified geometries can be instantiated. The top level ontology axioms providing an overview of the domain knowledge and the created application should be discussed to provide a general system overview. In the next section, we will highlight the different axioms of this top level ontology in relation with the spatial knowledge processing.

- Semantic - dc:DomainConcept
- Geometric - geom:Geometry
- Spatial Relationship – sp:hasSpatialRelation
    - Topological Relationship - topo:hasTopologicalRelationships
    - Metric Relationship - met:hasMetricRelations
    - Directional Relationship - direc:hasDirectionalRelations

Within the geometry class axiom, defined as: *geom:Geometry*, several properties and restrictions are defined for describing geometric data and for associating geometries with other features. The class *geom:Geometry* is a main top level class, where *3D_Spatial_Geometry*, as a subclass, contain candidates geometry for the 3D spatial processing and defined by the following XML syntax of a class definition and its related DL expression.

```
<owl:Class
      rdf:about= "http://www.WiDOP.de/DB.owl#3D_Spatial_Geometry">
      <rdfs:subClassOf>
            <owl:Class
                rdf:about="http://www.WiDOP.de/DB.owl#_3D"/>
       </rdfs:subClassOf>
</owl:Class>
```

$$\text{3D\_Spatial\_Geometry} \sqsubseteq \text{\_3D} \tag{15}$$

3D Geometry individuals are defined by their shapes, standardized via the CSG (Choi et al., 2009) or the SNC (Granados et al., 2003) geometric data structure or adopted in the case of lines and planes. The important class axiom *geom:Geometry* stores the local coordinates and the object geometry characteristics, and differentiates from an object to another within the knowledge base. This generalized class is specialized into *`geom:_2D`* and *`geom:_3D`* subclasses to specify the object geometry.

The semantics of objects in the knowledge base is defined through their 3D geometric, characteristics and spatial one. This is managed through the specialized object property *`geom:hasGeometry`*, *`charac:hasCharacteristics`*, `spa:` *`hasSpatialRelations`*. Once both the object and its coordinates are enriched, `spa:` *`hasSpatialRelations`* provides a relationship between geometry individuals. Finally, for geometry qualification purposes, the class axiom *`dc:DomainConcept`* represents the detected and annotated objects. This class axiom is the generalized class of any object within the point cloud scene. This class is further specialized into classes representing the different objects in the scene, as in Figure 4-3. (Ben Hmida et al., 2012).

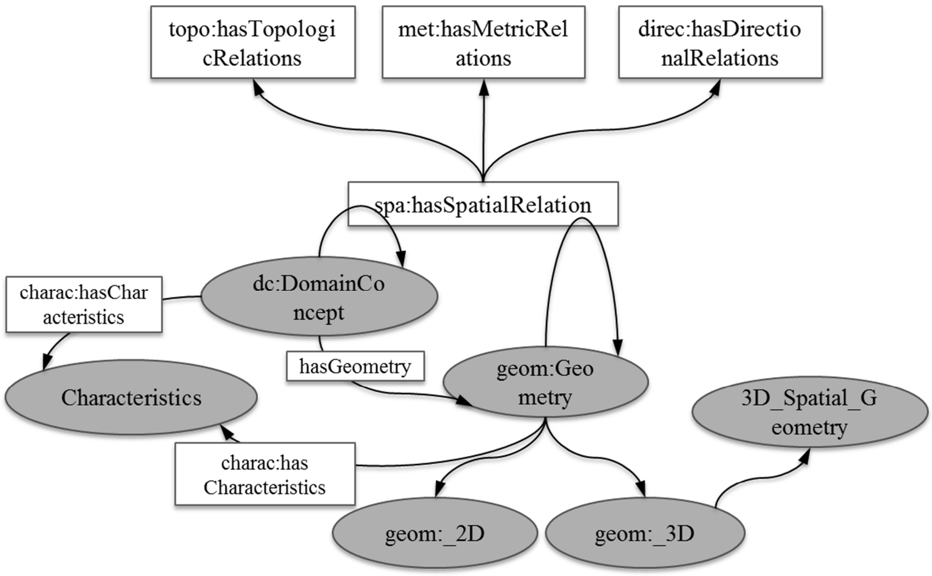


Figure 4-3. An overview of the class and the object properties axiom for the 3D spatial adjustment process

In order to highlight the utilisability of the created 3DSQ tool, we decided to extend the research by taking a step forward from the qualification of the spatial relation semantically to the extension of the semantic rules and query language. Such an improvement will support the inference on 3D spatial knowledge and will finally enable querying spatial knowledge base. Added to its ability to process spatial data in our case, the semantic approach will ensure a common understanding of the spatial domain between humans and machines via the semantic inference and queries using spatial knowledge. The presented solutions for spatial relationship qualification rely on the marriage between the low level quantification and the high level qualification of the spatial relation. Once the top level ontology with logic definition of geometries and 3D spatial relations is adjusted, a new need to ensure an automatic link between both levels is required. To do so, new specific built-ins called spatial are used. In fact, they play a major role in synchronizing both levels; such new created built-ins will be designed within the top level knowledge base as axioms of the defined semantic rules. Once executed, a direct call of correspondent low level spatial function will ensure the validation of the candidate spatial predicate. Once returned, the created translation engine within the 3DSQ platform will control the re-generation of new simple rules to be semantically inferred. The declaration of the spatial built-ins in our cases respects the standard nomination suggested by Egenhofer (Egenhofer, 2010). As a convention, each built-in begins with the prefix "swrlb_" concatenated to the spatial relation type where the first syllable states that it presents complex built-ins while the second one highlights the type of built-ins. Finally, the type of spatial topological predicate, for example "Inside" will validate such a relation, (Ben Hmida et al., 2012).

4.3.1 Implementation of the 3D spatial topological operators

This section mostly details the last one through a survey on the different top level ontology classes and objects properties in relation with the qualitative spatial topologic relation. Likewise, details related to the linking process between the high semantic level, and the low physical one related to the topological relations qualification issues will be discussed. Such a bridge is materialized via description logic and semantic rules concept with the help of new defined Built-Ins, demonstrating the real execution of the spatial relations between objects. Finally, the adjustment of the OWL ontology with new knowledge and mainly the inference process will take place based on the mentioned semantic web technology.

Regarding the ontology, the top level ontology is created to model the topological relationships. This ontology is used to enrich an existing knowledge base, so as to make it possible to define topological relationships between objects. Table 4-1 summarizes, for each topological relation, its name in the ontology using the prefix "topo", its semantic characteristics and the new built-in to automatize the computation of the relations with the help of any semantic rule system.

4.3.1.1 *Semantics of 3D topological relationships in the ontology*

The top level ontology is created to model the topological relationships where ontology is used to make it possible to define topological relationships between objects. The different topological relationships inherit from an upper topologic relation called “topo:hasTopologicalRelationships”.

- topo:hasTopologicalRelationships
  - topo:equals
  - topo:disjoint
  - topo:intersects
  - topo:touches
  - topo:crosses
  - topo:within
  - topo:contains
  - topo:overlaps

To ensure the execution of the Egenhofer relationship in the best conditions, each predefined topological model is linked to a set of spatial objects in the OWL top level. Bellow, the XML syntax of an Object Property definition and its DL expression is presented.

```
<owl:ObjectProperty
     rdf:about="http://www.WiDOP.de/spatial.owl#hasTopologi
     calRelationships">
     <rdfs:domain
          rdf:resource="http://www.WiDOP.de/spatial.owl#3D_
          Spatial_Geometry"/>
     <rdfs:range
          rdf:resource="http://www.WiDOP.de/spatial.owl#3D_
          Spatial_Geometry "/>
</owl:ObjectProperty>
```

$$\mathrm{3D\_Spatial\_Geometry} \sqsubseteq \exists \mathrm{hasTopologiecalRelationship.\,3D\_Spatial\_Geometry} \qquad (16)$$

The next table summarizes, for each topological relation, its name in the ontology using the prefix “*topo*”, its semantic characteristics. In addition, two inverse relations are defined in the top level ontology. The *topo:inside* relation is the inverse relation of *topo:contains*, and the relation *topo:covers* is the inverse relation of `topo:coveredBy`. The topological relationships between objects are created automatically by the rule calculation process in the ontology. Once a relation has already been computed, then there is no need to recalculate it.

Topologic Relation	Name of the property	DL Characteristics
Disjoint	topo:disjoint	Symmetric, irreflexive
Meets	topo:meets	Symmetric, irreflexive
Contains	topo:contains	Transitive, asymmetric, irreflexive
Inside	topo:inside	Transitive, asymmetric, irreflexive
Covers	topo:covers	Asymmetric, irreflexive
CoveredBy	topo:coveredBy	Asymmetric, irreflexive
Equals	topo:equals	Transitive, symmetric, reflexive

Table 4-1. The semantic definition of spatial relationships

$$\text{topo: equals} \sqsubseteq \text{hasTopologiecalRelationships} \tag{17}$$

$$\text{topo: inside} \equiv \text{topo: contains}^{-} \tag{18}$$
$$\text{topo: covers} \equiv \text{topo: coveredBy}^{-}$$

Table 4-1 and the related equations illustrate the different description logic characteristics (DLs) for every topological relationship. They are mainly used to infer new topological relationships without having to make any new calculations. To explain in more detail, a relation R is transitive if whenever an element x is related to an element y, and y is in turn related to an element z, then x is also related to z. A relation R is symmetric if whenever x is related by R to y, then y is related by R to x. On the contrary, a relation R is asymmetric if each time x is related by R to y, then y is not related by R to x. Finally, a relation R is reflexive if x is related by R to itself and a relation R is Irreflexive if x is not related by R to itself.

4.3.1.2 *The 3D topological built-ins*

As seen in Table 4-2, topologic functions demonstrate the topologic relations between objects; therefore, they are directly used to adjust the ontology. These topological relations are adjusted as specialized object properties of the `topo:hasTopologicalRelationships`. The different topological built-ins identify the nature of topological relations and execute them. It then populates the suitable object properties in the knowledge base. The next table presents the corresponding built-ins for each relationship.

Topologic Relation	Name of the property	SWRL built-ins
Disjoint	topo:disjoint	swrlb_Topo:disjoint(?geom1, ?geom2)
Meets	topo:meets	swrlb_Topo:meets(?geom1, ?geom2)
Contains	topo:contains	swrlb_Topo:contains(?geom1, ?geom2)
Inside	topo:inside	swrlb_Topo:inside(?geom1, ?geom2)
Covers	topo:covers	swrlb_Topo:covers(?geom1, ?geom2)
CoveredBy	topo:coveredBy	swrlb_Topo:coveredBy(?geom1, ?geom2)
Equals	topo:equals	swrlb_Topo:equals(?geom1, ?geom2)
Overlaps	topo:overlaps	swrlb_Topo:overlaps(?geom1, ?geom2)

Table 4-2. The Built-ins that compute spatial relationships

During the execution of the topological built-ins, the topological rule engine first calls the functions under the required category with mainly two features. The features "?geom1" and "?geom2" are individuals of the class *geom:Geometries*, extracted from the OWL ontology structure. Once the Built-Ins execution is done, and in order to maintain the qualified relationship in case of a true returned assignment, the relationships between the spatial geometry individuals are populated in the ontology via object properties inherited from the *topo:hasTopologicalRelationship* and added to the top level ontology, Table 4-3. During the built-Ins execution, a real link between the high qualitative spatial operators and the low quantitative one is established. In fact, the correspondent 3D spatial function already explained and discussed in the last chapter (Section 3.3.3) is called and executed, Figure 4-1. Once done, the produced spatial relations results is returned and processed through the semantic level. All this process is ensured through a translation engine discussed in section 4.4.

Qualitative Topological function (SWRL/SPARQL)	Equivalent quantitative Relation
SWRL_Topo:equals (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): Boolean	Boolean Res= equals (geom1, geom2)
SWRL_Topo:disjoint (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): boolean	Boolean Res= disjoint (geom1, geom2)

SWRL_Topo:meet (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): boolean	Boolean Res= meet (geom1, geom2)
SWRL_Topo:overlap (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): boolean	Boolean Res= overlap (geom1, geom2)
SWRL_Topo:covers (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): Boolean	Boolean Res= covers(geom1, geom2)
SWRL_Topo:coveredBy (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): Boolean	Boolean Res= covered by (geom1, geom2)
SWRL_Topo:inside (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): Boolean	Boolean Res= inside (geom1, geom2)
SWRL_Topo:contains (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry): Boolean	Boolean Res= contains(geom1, geom2)

Table 4-3. The topological predicate execution

4.3.2 Implementation of the 3D Spatial Qualitative Metric operators

Metric knowledge's presents important information especially in several engineering domains, where we have lots of requests from industries for metric measurement between varieties of detected elements in specific scenes. As example, the German Railway domain needs such sophisticated knowledge for distance measurement especially that the different installed elements flows very strict rules imposing that the distance between a couple of Electric terminals has to be an average of 50 m for example. Thus, the implementation of metric measurement knowledge and rules is highly recommended. As seen in the last chapter, and to support the metric knowledge, the already created top level ontology will be extended with new axioms in relation with the metric knowledge processing via the class: `met:Distance.`

The class axiom `met:Distance` presents the main class axiom of any metric class. It presents also the generalized class of any Metric object. The next important object of a property axiom is `met:hasdistanceGeom`. This property aims to create the input geometry for the distance function processing. Finally, the qualified distance metric value is stored in the data property axiom `hasDistanceValue`, Figure 4-4. In fact, description logics and OWL language do not allow the definition of ternary relationships. The principle that consists in using a fourth object to link three objects is well known and often used in literature, (Baratis et al., 2009).

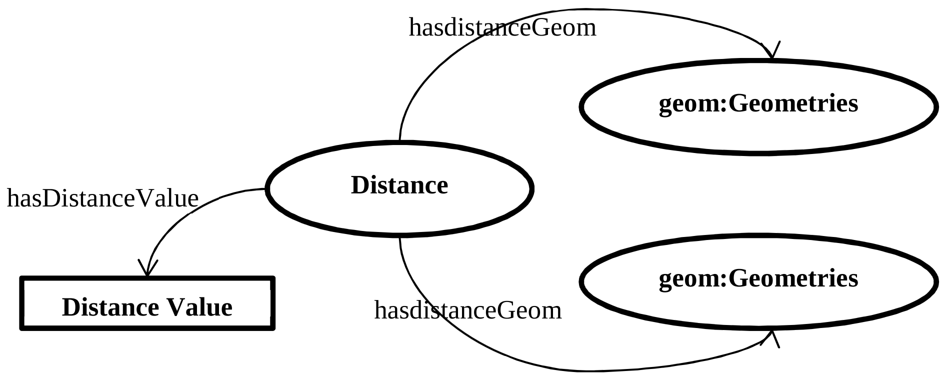


Figure 4-4. The qualitative distance modelling

4.3.2.1 *The Distance relationship*

The metric processing operation returns the real measurement of metric distance between couples of geometries on their execution. It is hence important to have a provision where to store the returned qualified metric value. To do, the data property axiom `met:hasDistanceValue` is created. Likewise, the input geometry for the distance processing function is managed through the object property axiom `met: hasdistanceGeom` object properties with a restriction of getting exactly two inputs.

As seen before, the metric relation processing function returns metric value, and then adjusted in the ontology via `met:distance`, `met:hasdistanceGeom` and `Met:hasdistanceValue`. The initial step consists of the built-Ins parsing to be processed by the translation engine (Section 4.4), where the correspondent directional function will be running with respect to the metric specification defined in the last chapter (Section 3.2.3). In fact, the feature on which the built-Ins is applied are basically Bounding boxes geometries.

Function	Class	ObjectProperty	Data property	Built-Ins
Distance	Met: Distance	Met: hasdistanceGeom	Met: hasdistanceValue	Swrlb: distance(?x,?y,?z)

Table 4-4. The metric predicate execution

4.3.2.2 *The Qualitative Metric relationship*

Once the metric knowledge materialized through the distance class and the processing built-Ins behind takes place, new ambitions and challenges are seeing the light, basically, the qualification of new knowledge like *met:isNearTo* and *met:isFarFrom* that can take place via the creation of new object properties axioms depending on a specific threshold.

Based on such an interval (Section 3.2.2), new qualitative distance relations can be generated through the execution of complex semantic rules including the metric distance built-Ins, Table 4-5, where "?x" and "?y" presents the candidate 3D spatial geometries while "?z" presents the created distance class individuals. Once done, the "?val" value return the real numerical distance related to the "?z" distance individual. As discussed in section 3.2.2, the "Thres_Max" and the "Thres_Min" are relative threshold values that vary from one domain to another.

Metric qualitative Relation	Execution rules	DL Characteristics
isFarFrom	3D_Spatial_Geometry (?x) ^ 3D_Spatial_Geometry (?y) ^ swrlb:distance(?x,?y,?z) ^ hasDistanceValue(?z,?val) ^ swrlb:greaterThan(?val,Thres_Max) → isFarFrom(?x,?y).	symmetric, irreflexive
isNearTo	3D_Spatial_Geometry (?x) ^ 3D_Spatial_Geometry (?y) ^ swrlb:distance(?x,?y,?z) ^ hasDistanceValue(?z,?val) ^ swrlb:lessThan(?val,Thres_Min) → isNearto(?x,?y)	Symmetric, Irreflexive
hasMediumDistance	3D_Spatial_Geometry (?x) ^ 3D_Spatial_Geometry(?y) ^ swrlb:distance(?x,?y,?z) ^ hasDistanceValue(?z,?val) ^ swrlb:greaterThan(?val, Thres_Max) ^ swrlb:lessThan(?val, Thres_Min) → hasMeduimDistance(?x,?y)	Symmetric, Irreflexive

Table 4-5. The qualitative metric relationships

4.3.3 Implementation of the 3D Spatial Directional operators

The directional relationship of 3D spatial objects presents important spatial information in any 3D system or GIS one involved in spatial query, and especially spatial analysis. The directional relations are most commonly represented with qualitative and quantitative representation models. The quantitative models represent direction relationships by angle in most cases, while the qualitative models use ordered classes. Traditional models of directional relations define a certain class of direction relationships between spatial objects within any data base. This section, as a continuation of the 3D spatial knowledge modelling and processing, deals with a new directional relation model based on semantic knowledge processing, where semantic directional knowledge can be well propagated. This model expresses the semantic directional relation through specific class axioms, object properties axioms, data properties axioms and new created directional built-Ins qualifying the basic directional relations between 3D geometries. In this field, semantic web technology provides a new more reasonable and rich model to analyse and propagate the directional knowledge. Furthermore, it is a human machine-understandable model qualifying directional relation concept. From this point of view, we see directional relationships as an extension of the predefined topologic knowledge with taking the orientation knowledge into account. To ensure efficient homogeneous directional functions, all 3D geometric objects must be presented in the same space and with the same global reference system. In this field, the concept of orientation and topological information is integrated into the single model.

One of the most required directional relations in concordance with the architectural scene is the "*isOn*" and "*isAbove*" relation. While the first one reflects the fact that the second geometry is situated on the first one, the second one reflects the fact that the first geometry is situated above the second one respectively. Such knowledge is very useful in many use cases especially with architectural scenes where a use case example can be to select and visualize vertical geometries "*isOn*" the ground which can eventually be qualified as Walls. The example qualifying *isOn* relation between two geometries includes lots of implicit topologic relation knowledge since it first means that the first geometry "*Meet*" the second one. From another side, it also includes the fact that the lower Z points value of the first geometry are equal to the higher Z point value of the second one. The directional relation in question is integrated with the knowledge base model via the use of directional Built-Ins and object properties axioms *direct:isOn, direct:isAbove…* As seen in the last chapter, these functions demonstrate the created directional relations; hence, it is very straightforward when adjusting the ontology. It can be directly adjusted through the object property *direct:East_of*, *direct:West_of* within the top level ontology. Table 4-6 illustrates the different steps for the above mentioned directional qualitative relations.

Function	Built-Ins	Object Property	Characteristics
Above	Swrlb:IsAbove (?x,?y)	isAbove	Transitive Inverse of “IsBelow”
Below	Swrlb: IsBelow (?x,?y)	IsBelow	Transitive Inverse of “isAbove”
WestOf	Swrlb: On _WestOf (?x,?y)	On_westOf	Transitive Inverse of “On_EastOf”
NorthOf	Swrlb: On _NorthOf (?x,?y)	On_NorthOf	Transitive Inverse of “On_SouthOf”
SouthOf	Swrlb: On _SouthOf (?x,?y)	On_SouthOf	Transitive Inverse of “On_NorthOf”
EastOf	Swrlb: On _EastOf (?x,?y)	On_EastOf	Transitive Inverse of “On_NorthOf”

Table 4-6. The directional predicate execution

The spatial relationship with its three variations presents a great complete process. However, each one of the spatial variants can play a major role as a complement supporting the qualification of the other ones semantically. In this field, the semantic inference systems on spatial quantitative relations have added through this thesis a new dimension to the above mentioned domain. In fact, such a system has become possible through different rules and DLs expression to infer new spatial relations based on the primary qualified ones via Built-ins. As an example, the next rule illustrates the adjustment of the semantic rule language with new 3D knowledge to infer the directional relation “*isOn*” from a combination of “*Meet*” and “*isAbove*”.

$$\text{Meet}(?x,?y) \wedge \textit{isAbove}(?x,?y) \rightarrow \text{isOn}(?x,?y) \qquad (19)$$

Likewise, the following rule demonstrates the generation of metric knowledge based on a combination of topological and directional ones.

$$\text{Overlap}(?x,?y) \rightarrow \text{isNearTo}(?x,?y) \qquad (20)$$

4.3.4 Implementation of the 3D spatial processing functions

As a convention, and far from the spatial predicates returning Boolean values, the spatial processing operations (Boolean) return solid geometries to their executions. It is hence important to have provision to store these returned geometries in the ontology within the *Geom:Geometry* class. An upper level class, the *proc:SpatialProcessing* one is recently introduced in the top level ontology where every spatial processing function is then adjusted as one of its subclass. The class hierarchy of *proc:SpatialProcessing* reveals that the subclasses within it are the classes which need to calculate and return geometries in some form of combinations.

The discussed spatial functions within this section are Union, Intersection and Difference. These functions compute new 3D solid geometries stored in the geom:Geometry class in order to get qualified later on. To do, we have defined new three classes called *proc:Union*, *proc:Intersection* and *proc:Difference* which are of specialized classes of *proc:SpatialProcessing* one. The classes are instantiated once the *SpatialProcessing* operation are executed through new specific built-Ins The result of execution is stored within the instantiated individual in the geom:Goemetry class. The spatial processing functions under this category need to take solid geometries as input to execute them. The feature presents geometries within class *geom:Geometry*. In order to maintain a relationship between the processing operations under *proc:SpatialProcessing* and geometry under *geom:Geometry* in the ontology, a top level object property *proc:hasInputGeom* and *proc:hasOutputGeom* are created and added to the top level ontology.

For example for every instance in class proc:Union (subclass of proc:SpatialOperation) has a property proc:hasUnioninput (specialized object property of proc:hasInput) which relates the proc:Union class to the classes specializing Geom:Geometry. There are also three defined object properties corresponding to each topologic functions (proc:Union, proc:Intersect, proc:Difference), Table 4-7, Figure 4-5. The correspondent low level calculation of the processing spatial functions was not introduced in the last chapter since it will be directly ensured by predefined functions related to the 3D Nef Polyhedra libraries.

$$\text{proc: SpatialProcessing} \sqsubseteq \exists\text{proc: hasIntputGeom. geom: \_3D} \wedge \exists\text{proc: hasOutputGeom. geom: \_3D} \quad (21)$$

$$\begin{gathered}\text{proc: hasUnionInput} \sqsubseteq \text{ proc: hasIntputGeom} \\ \text{proc: hasIntersectionInput} \sqsubseteq \text{ proc: hasIntputGeom} \\ \text{proc: hasDifferenceInput} \sqsubseteq \text{ proc: hasIntputGeom}\end{gathered} \quad (22)$$

$$\begin{gathered}\text{proc: Union} \sqsubseteq \text{pro: SpatialProcessing} \wedge \text{proc: hasUnionInput} \\ \text{proc: Intersection} \sqsubseteq \text{pro: SpatialProcessing} \wedge \text{proc: hasIntersectionInput} \\ \text{proc: Difference} \sqsubseteq \text{pro: SpatialProcessing} \wedge \text{ proc: hasDifferenceInput}\end{gathered} \quad (23)$$

Function	Concepts	SWRL/SPQRQL	Equivalent relation
Union	`geom: Union`	`proc:Union(geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry ): 3D_Spatial_Geometry`	`3D Geom = Union (geom1, geom2)`
Difference	`geom: Difference`	`proc:Difference (geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry ): 3D_Spatial_Geometry`	`3D Geom = Difference (geom1, geom2)`
Intersect	`geom: Intersect`	`proc:Intersect(geom1: 3D_Spatial_Geometry, geom2: 3D_Spatial_Geometry ): 3D_Spatial_Geometry`	`3D Geom = Intersect (geom1, geom2)`

Table 4-7. The Processing functions execution

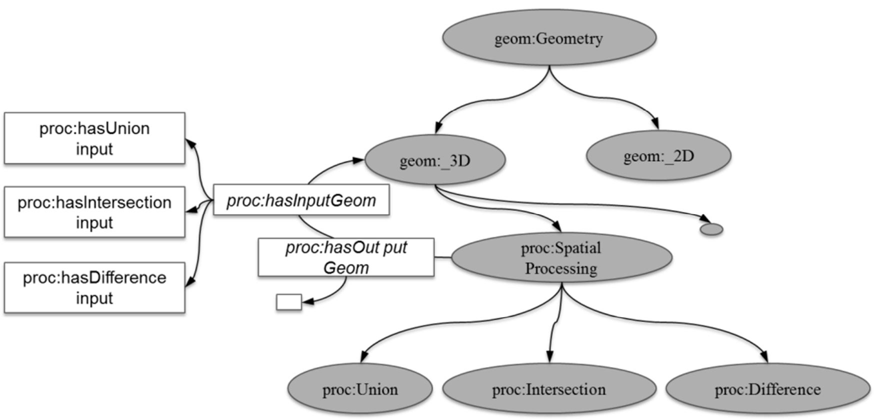


Figure 4-5. An overview of the classes and the properties axiom for the 3D spatial processing

4.4 Translation engine

Once all the functions related to the quantitative operators and qualitative spatial relationships are implemented, they will be used through the semantic Web rule Language rules (SWRL) (Horrocks et al., 2004) and the Semantic Web Query Language rule (SWQRL) (OConnor & Das, 2009) from one side and the Query Language for RDF (SPARQL) (Sirin & Parsia, 2007) from another side. The execution of the built-ins looks first for the type of the parameters, and then gets all the individuals of these parameters. Once done, the related spatial functions are executed, taking as a parameter the individuals of the first and second types. After its execution, the built-in is replaced by the name of the related property in the expression of the rule so that the inference can be done. Once achieved, the qualified relation is populated into the ontology, (Ben Hmida et al., 2012).

The above detailed process is ensured through the created translation engine. It enables the computation of spatial semantic rules (SWRL, SWQRL) and queries (SPAQRL) and interprets the statements to parse the spatial components. Once done, they are computed through relevant spatial processing functions and operations by the translation engine, through the operations provided in the above mentioned CSG (Corporation, 2006) or Nef_Polyhedron geometry level (Granados et al., 2003). Once done, the results are populated in the knowledge base, thus making it spatially rich. After that, the spatial statements are translated to standard ones 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 eventually stored in the knowledge base, Figure 4-6.

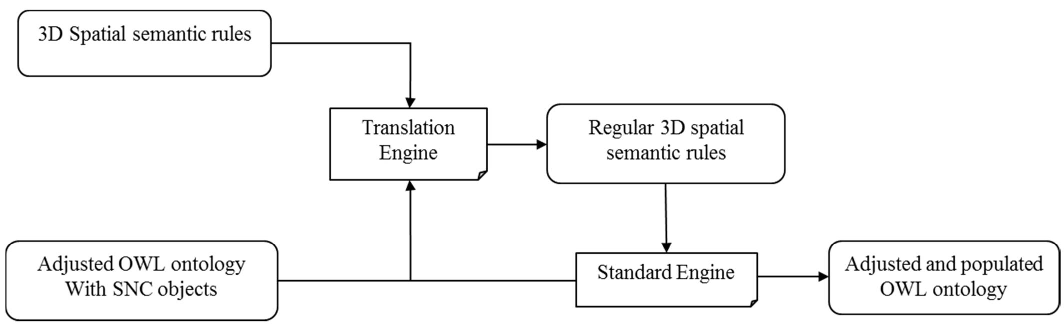


Figure 4-6. On overview of the translation engine principle

4.5 Conclusion

In the current chapter, the properties of the target 3D Spatial Qualification platform (3DSQ) are presented and discussed. In fact, the 3DSQ platform has to take onto account the geometry structure and its ability to specify the internal, external and the boundaries of each one of the geometries. Once done, it has to suggest formal logic expressions able to satisfy the intersection model in each case, to be mapped later on to the semantic level. Likewise, by the actual contribution, the semantic qualification will be linked to the quantitative one, where no further complex modification on the Standards $\mathcal{SHOIQ}$ language (Horrocks et al., 2003) neither in any reasoners will be achieved, thus avoiding complex computation while qualifying spatial relation based on DLs language. Finally, it is highly recommended that such a solution separates the low level quantification from the high level qualification, while always ensuring a communication bridge between both of them.

The research work presented here makes an attempt to emphasize the possibility to combine 3D spatial technology and the above-mentioned 3D spatial reasoning by the integration of such a technology in the semantic web framework. This chapter discusses the 3D spatial operators and their integration within a quantitative manner for an OWL knowledge base. The semantics of the spatial predicate and relations is formally implemented and defined in an OWL knowledge base from one side and linked to a quantitative layer from the other side. The Selective Nef Complex based implementation technique of spatial operators, which was highlighted, and considered as the most efficient one compared with the CSG and Octree implementation, supports a big variation of non-uniform geometries with high discrimination of the Interior, Boundary and Exterior of each of them was adapted for this work.

In addition, this chapter moves outside of the range of data interoperability while presenting the concepts and makes an effort to utilize other areas of semantic Web technologies. The basic capacity of knowledge processing provides to the semantic web the capabilities to process the semantics of the information through close collaboration with the machine. In fact, it makes not only the perceptive of 3D spatial data easier for interoperability among different data sources, but mainly provides helpful knowledge enabling to enrich the knowledge base. Such processes will primary help to understand spatial data and relations in a better way. From our point of view, it is extremely important to have standard terms for every created spatial relationship and built-in to process the 3D spatial knowledge. To do so, we tried to rely on the standards presented by the W3C and OCG.

As perspectives, 3D spatial reasoning with its three derivations (3D Topologic reasoning, 3D metric reasoning and 3D directional reasoning) monopolize an important area of Artificial Intelligence, where existing approaches in this field are mainly numerically based. Geometric properties and spatial relationships between building elements play a major role in the design processes of the Architecture, Engineering and Construction domain. So far, spatial relations are not supported by existing building information models. To close this technological gap, a bridge between a qualitative spatial relationship and the quantitative one has been developed. In the next section, and based on the approved spatial relation qualification, we will look at the developed 3DSQ platform and discuss it through a real applied area related to indoor architectural scene extracted from CAD geometries. In this field, the problematic related to geometries qualification in 3D data will be dealt with where a formal solution based on semantic technology and on the 3DSQ platform will be presented.

References

Antoniou, G. & Harmelen, F., 2009. Web ontology language: OWL. Handbook on ontologies, pp.91--110.

Baader, F., Horrocks, I., Sattler, U., 2005. Description logics as ontology languages for the semantic web. Mechanizing Mathematical Reasoning, pp.228--248.

Baader, F. & Sattler, U., 2001. An overview of tableau algorithms for description logics. Studia Logica, 69, pp.5--40.

Baratis, E., Petrakis, E., Batsakis, S., Maris, N., Papadakis, N. 2009. TOQL: temporal ontology querying language. Advances in Spatial and Temporal Databases, pp.338--354.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F, 2012. From 9-IM Topological Operators to Qualitative Spatial Relations using 3D Selective Nef Complexes and Logic Rules for bodies. In KEOD 2012, 4th International Conference on Knowledge Engineering and Ontology Development. Barcelona, Spain, 2012.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2012. From Quantitative Spatial Operator to Qualitative Spatial Relation Using Constructive Solid Geometry, Logic Rules and Optimized 9-IM Model. IEEE International Conference on Computer Science and Automation Engineering (CSAE 2012), 3, pp.453- 458.

Blumauer, A. & Pellegrini, T., 2006. Semantic Web und semantische Technologien: Zentrale Begriffe und Unterscheidungen. Semantic Web, pp.9--25.

Choi, G.H., Mun, D., Han, S., 2009. Exchange of CAD section models based on the macro-parametric approach. International Journal of CAD/CAM, 2, pp.13--21.

Corporation, L., 2006. Leadwerks Engine. [Online] Available at: http://www.leadwerks.com/files/csg.pdf.

Egenhofer, M.J., 2010. Qualitative Spatial-Relation Reasoning for Design. In Studying Visual and Spatial Reasoning for Design Creativity. Springer.

Ellul, C. & Haklay, M.M., 2009. Using a B-rep structure to query 9-intersection topological relationships in 3D GIS--reviewing the approach and improving performance. 3D Geo-information Sciences, pp.127--151.

Granados, M., Hachenberger, P., Hert, S., Kettner, L., Mehlhorn, K., Seel, M., 2003. Boolean operations on 3D selective Nef complexes: Data structure, algorithms, and implementation. Algorithms-ESA 2003, pp.654--666.

Gruber, T., 2008. What is an Ontology. Encyclopedia of Database Systems, 1.

Grutter, R. & Bauer-Messmer, B., 2007. Combining OWL with RCC for Spatioterminological Reasoning on Environmental Data. Proc. of 3rd OWL: Experiences and Directions (OWLED2007).

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission, 21, p.79.

Horrocks, I., Patel-Schneider, P.F., Van Harmelen, F., 2003. From SHIQ and RDF to OWL: The making of a web ontology language. Web semantics: science, services and agents on the World Wide Web, 1, pp.7--26.

Lederer, A.L., Maupin, D.J., Sena, M.P., Zhuang, Y., 2000. The technology acceptance model and the World Wide Web. Decision support systems, 29, pp.269--282.

Li, S. & Ying, M., 2003. Region Connection Calculus: Its models and composition table. Artificial Intelligence, 145, pp.121--146.

McGuinness, D.L. & Van Harmelen, F., 2004. OWL web ontology language overview. W3C recommendation, 10, pp. 2004--03.

OConnor, M. & Das, A., 2009. SQWRL: a Query Language for OWL. In 6th International Workshop on OWL: Experiences and Directions (OWLED 2009). Chantilly, VA, United States, 2009. Vrije Universiteit Amsterdam.

Parsia, B. & Sirin, E., 2004. Pellet: An owl dl reasoner. In 3rd International Semantic Web Conference (ISWC2004)., 2004.

Sirin, E. & Parsia, B., 2007. Sparql-dl: Sparql query for owl-dl. In In 3rd OWL Experiences and Directions Workshop (OWLED-2007., 2007.

Stocker, M. & Sirin, E., 2009. Pelletspatial: A hybrid rcc-8 and rdf/owl reasoning and query engine., 2009. Proc. OWL: Experiences and Directions (OWLED), 6th International Workshop.

Chapter 5

The 3DSQ platform

Demonstration using 3D CAD/IFC geometries

5.1 Introduction

The present chapter aims to validate, through real use cases, the principles exposed in the previous ones. The major context behind the current chapter is to present and discuss the developed generic concept for spatial relation and geometric object qualification from one side and to make a step to manage engineering problem based on the 3DSQ platform. In this chapter, the context of the study is the CAD geometric data where a description of the developed **3D S**patial **Q**ualification platform and its visualization will be presented for a given geometric data. More precisely, we will consider CAD (Armstrong et al., 2002) and IFC (Hallberg & Tarandi, 2011) geometric structure enrichment with semantic knowledge.

The created 3DSQ Platform make an attempt to ensure interaction between heterogeneous environments. Currently, such a semantic platform connects an adjusted OWL ontology structure, a 3D quantification engine, a visualization engine and a set of geometry via a knowledge processing layer materialized via SWRL (Horrocks et al., 2004) and SQWRL rules (OConnor & Das, 2009) within its extended Built-Ins. It first enables, via the developed interface, the loading of an OWL ontology structure adjusted with spatial knowledge from one side and to populate its content with a selected set of geometries from another side. The created spatial built-ins are connected to the presented quantification engine discussed in chapters 3 and 4, and furthermore enable qualifying semantic spatial relations on them. This will mainly require us not just to apply semantic queries selecting geometry based on such a qualified relation, but also to benefit from the richness of the knowledge based schema, the related geometric characteristics of such an object added to the spatial one to ensure the semantic qualification process. It mainly consists of affecting a semantic identity to a geometry, yielding in the end a rich ontology structure processing in a much more flexible way the content of a CAD/IFC file, and serving as an intelligent knowledge base for any Building Information Model (Eastman et al., 2008). To give more details, the relative architecture of the developed 3DSQ platform will be provided. In a second section, example queries and rules will be presented and proved through a real use case, showing the functionality in practice and the final results with a brief discussion of the achievements reached.

The **3D S**patial **Q**ualification tool (3DSQ) was built to compute data stored in OWL-DL ontology. By using the adjustment principle of an existing ontology, it is then possible to add 3D data to existing objects and compute their spatial relationships. After demonstrating the impact of such a tool and the related knowledge modelling and processing in the filed in 3D spatial relation with very generic knowledge, an applied use case related to an architectural scene will be used for validation. A demonstration of the impact of spatial knowledge on the geometry qualification in the case of an architecture scene with existent geometries will be realized. Thereby, it will be achieved via CAD objects related to the Frankfurt airport (Fraport) (Fraport, 2012). The airport scene is an indoor architectural scene. It contains regular walls, floor, chairs, advertisement panels, signs etc. The whole scene was stored as 3D CAD data resulting in large geometries representing object boundaries. Based on this data and the defined knowledge, we will try to give a clear understanding of benefits from a knowledge base processing and mainly the 3DSQ tools, Figure 5-1.

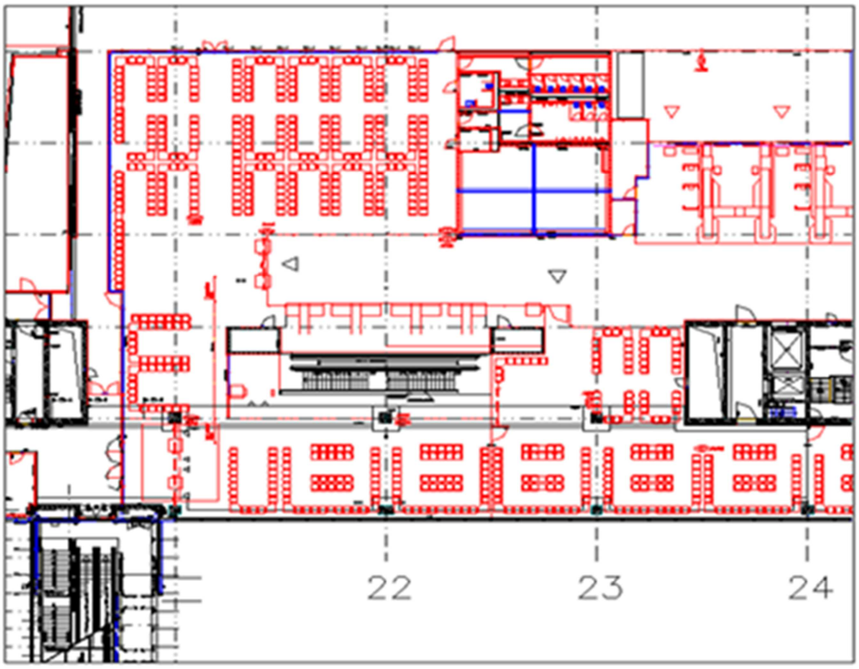


Figure 5-1. The Fraport Gate 2 CAD file

This chapter will continue with the 3DSQ architecture overview, establishing a base for ontology adjustment, rules processing and result visualization. Section 3 presents the first extension of the 3DSQ platform in its principles for the object qualification purpose. The chapter concludes with the conclusion in section 4.

5.2 3DSQ architecture overview

Initially, the **3D S**patial **Q**ualification tool (Ben Hmida et al., 2012) was built to compute spatial data stored in OWL-DL ontology (Baader et al., 2005). By using the adjustment principle of an existing ontology, it is then possible to add 3D data to existing objects available in ontology and to compute their spatial relationships. The created 3DSQ platform, Figure 5-2, enables via the developed interface, to load an OWL ontology structure adjusted with spatial knowledge from one side and to populate its content with a selected set of geometries from another side. The created spatial built-ins are connected to the presented quantification engine discussed in the last chapter and enable further qualifying semantic spatial relation on them.

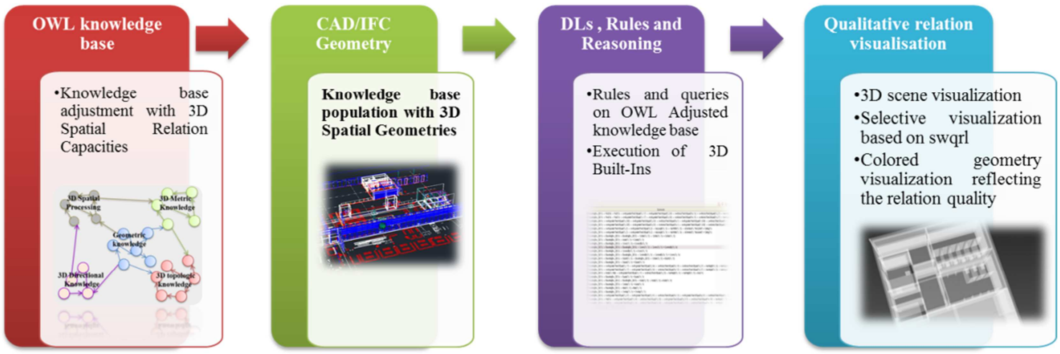


Figure 5-2. The 3DSQ Prototype component

The developed prototype requires connection to an OWL knowledge base. The ontology is used as a bridge to manage all the data obtained from the qualification of geometries. This includes their name and the directory, width, height, length, orientation and position. Likewise, the platform manages semantic rules that enable the program to qualify spatial relationships and characteristics from one side and to recognize and assign qualities to the already populated geometries. The ontology used is based on Web Ontology Language where characteristics and restrictions of each of the classes and properties will be automatically propagated to the individuals that will populate this ontology.

5.2.1 The Ontology adjustment process

To adjust the used owl ontology, a various number of geometries were created from the IFC (Vanlande et al., 2008) selected scene and populated in the OWL ontology adjusted with 3D spatial operators. During such a first step, a pre-processing phase takes place on which IFC and/or CAD data are transformed to independent Object Format File (OFF) files through specific local plug-Ins and software, where the directory of such a files is mainly kept within our knowledge base. It has to be noted that a variety of constructors can be used where polyhedrons can be created otherwise than with the OFF files. Let's remember that the Object Format File presents a data structure for storing 2D and 3D objects created from several polygons. It characterizes geometries by a set of vertex, sides and edges, where each one is presented by a set of XYZ coordinates in case of 3D. The geometries can be much more complex but have to be closed. The presented 3DSQ platform will run according to all the specifications needed, and will output reliable results independently. Added to this, it enables the export of scene elements as VRML files (W3C, 1995) for visualisation purposes where the 3D scene and the spatial qualification results are presented in different colours depending on the semantic of the relation. Once the 3DSQ engine is ready, the platform enables the execution of semantic processing rules with complex 3D spatial built-ins mainly based on two different languages: SWRL and SQWRL, Figure 5-3, where "*BldElem_xxx*" presents the name of the 3D geometries individuals already populated in the knowledge base that may contain any valid 3D geometry like "*Stairs*", Figure 5-6.

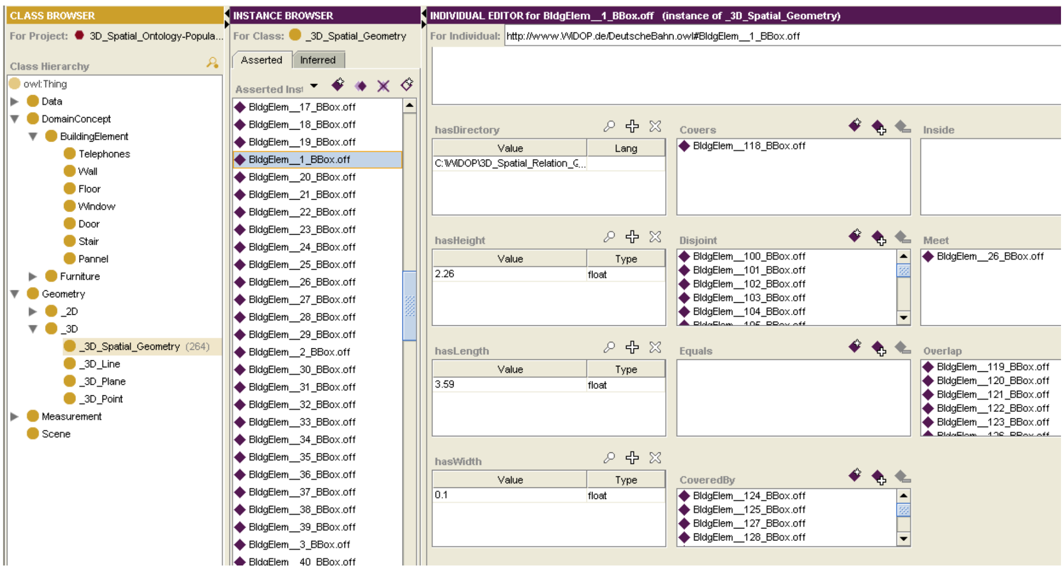


Figure 5-3. The 3DSQ OWL adjusted ontology overview

While running any 3D spatial built-ins, the 3DSQ engine first proceeds with the conversion of the different OFF files to Polyhedra structure through constructor standards defined by the CGAL library (http://www.cgal.org/). Once done, converted geometries are verified and validated as Nef Polyhedra structure. Valid 3D geometry has to be close, without redundant vertices presenting its structure. In such a case, the translation engine will proceed with the rule execution. It will first interpret the statements in order to parse the spatial components. Once done, the specific spatial relation is computed through relevant spatial functions and operations depending on the relation identity. During its execution, the rule engine first calls the functions under the required category with mainly two features. Once the built-ins execution is achieved, correspondent relationships between the individuals are populated in the ontology, thus making it spatially rich, Figure 5-4.

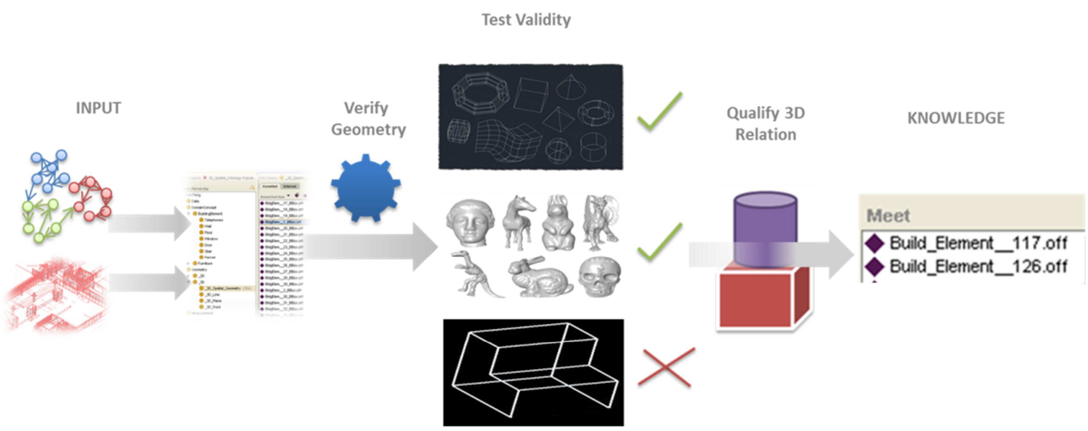


Figure 5-4. The 3DSQ Spatial relationship qualification process

5.2.2 3D Spatial Queries and Inference Rules

To highlight the capabilities of the 3DSQ Platform within its quantitative operators and qualitative 3D spatial relationships, a first subsection focuses on the query rule language SQWRL (OConnor & Das, 2009), and a second one shows reflecting the rules acting with the semantic web which is the SWRL language (Horrocks et al., 2004).

5.2.2.1 *Spatial Relationships and query language*

Actually, SQWRL (Semantic Query-Enhanced Web Rule Language) is a SWRL-based language for querying OWL ontologies. It provides SQL-like operations to retrieve knowledge from the OWL knowledge base. Like SQL, it enables counting operator, disjunction, complex counting and aggregation like sqwrl:isEmpty, sqwrl:union, sqwrl:difference, mathematical and logical predicate like: sqwrl:max, sqwrl:min, sqwrl:sum, sqwrl:orderBy. Added to that, SQWRL can act as a DL query language. With such a language, there is no need to invent a new semantic where standard presentation syntax is adopted. Likewise, it can use existing reasoning infrastructure and editors where mainly queries can interoperate with rules. The next rule is an example of a query that selects all distinct overlapping 3D_Spatial_Geometry in the current knowledge base.

$$\_3D\_Spatial\_Geometry(?x) \wedge \_3D\_Spatial\_Geometry\ (?y) \wedge swrl\_topo{:}overlaps(?x, ?y) \rightarrow sqwrl{:}selectDistinct(?x,?y) \quad (24)$$

In contrast with the literature, (Borrmann et al., 2006), (Borrmann & Rank, 2009) we can conclude that the same query capacity supported by SQL query language can be maintained via SQWRL rule language with a more light and portable knowledge base. In addition, the queries can interoperate with rules where no need to install huge SQL servers and extra data base software. Likewise, through the flexibility of the SQWRL rule language and its built-Ins, 3D Spatial Relationships can be qualified and computed just for the specific case and between specific candidate geometries as seen in the next rule.

$$\_3D\_Spatial\_Geometry\ (?x) \wedge hasSurface(?x,?s) \wedge swrlb{:}Greaterthan\ (?s,50) \wedge hasOrientation\ (?x, Horizontal) \wedge \_3D\_Spatial\_Geometry\ (?y) \wedge hasOrientation\ (?y,Vertical) \wedge hasSurface(?y,?s1) \wedge swrlb{:}Greaterthan\ (?s1,20) \wedge swrl\_topo{:}Meet(?x, ?y) \rightarrow sqwrl{:}select(?x,?y) \quad (25)$$

In the next section, we will take a step forward with this thesis via the use of the main inference capacity of the knowledge engineering areas. It lays its foundation on the Semantic Web Rules Language and the DLs capacities to infer on existing knowledge base adjusted with 3D Spatial Operators.

5.2.2.2 *Inference based on Semantic Web Rules Language*

As we have already demonstrated how the created platform support a main Semantic Query-Enhanced Web Rule Language, more prove of the robustness of the developed concept will be enhanced with examples reflecting the power of the OWL DL knowledge base and its related technology, especially the SWRL language. Known as SWRL, Semantic Web Rules Languages with the extended built-Ins to support the 3D spatial processing is performed. As seen in the next SWRL rule, the previous section is composed of classes like "_3D_Spatial_Geometry" and properties, but also built-ins for 3D spatial processing that will later on be converted to simple object properties in this case. In the consequent section, and once the spatial assertion is verified, "Meet" in this case, the (?y) elements will be denoted through the "Meet" object property as range of the elements (?x). As a first scenario, new spatial knowledge can be deduced from the geometric one and the 3D spatial engine via the SWRL rule:

$$\_3D\_Spatial\_Geometry\ (?x) \wedge \_3D\_Spatial\_Geometry\ (?y) \wedge swrl\_topo{:}Meet(?x, ?y) \rightarrow Meet(?x, ?y) \quad (26)$$

Likewise Inference engine can manage spatial semantic qualification without recourse to the developed spatial engine, for instance; the next simple SWRL rules deduce the relationship as being "disjoint" between two geometries. If A meets B and A contains C then A and C are disjoint, (Ben Hmida et al., 2012).

$$meet\ (?a, ?b) \wedge contains(?a, ?c) \rightarrow disjoint\ (?a, ?c) \quad (27)$$

The examples presented aim to raise the issues for the spatial relation integration within relatively new semantic technologies through the 3DSQ platform. Initially addressed to deal with heterogeneity in the web technology, the knowledge engineering technology is more and more deduced for the collaborative approach between humans and machine where human intelligence and reasoning are injected inside the actual 3DSQ solution. In fact, the created OWL knowledge base enables to store the full set of information available in CAD/IFC, including attributes and relationships, which will make it possible to employ such information added to the Spatial one for further process, especially like the geometry qualification in our case, yielding in the end a rich ontology structure processing in a much more flexible way the content of any building model.

5.2.3 The visualisation process

Figure 5-5 shows the prototype interface that the user will be interacting with. First, the prototype enables users to directly connect to an OWL knowledge base, after selecting the desired ontology; the prototype is ready to start analysing data in a qualitative manner depending on the desired purpose, inferring or querying on spatial objects. In fact, semantic querying and inferring technologies, which are completely embedded in the knowledge base structure, provide a high portability and efficiency, where no need for extra servers and huge data bases is required. In addition, it provides an interactive semantic based solution via the visual results. In such a case, the scene in question and the spatial qualification results are presented in different colours depending on the nature of the returned relation, Figure 5-6 which corresponds to the populated and adjusted OWL ontology of Figure 5-3.

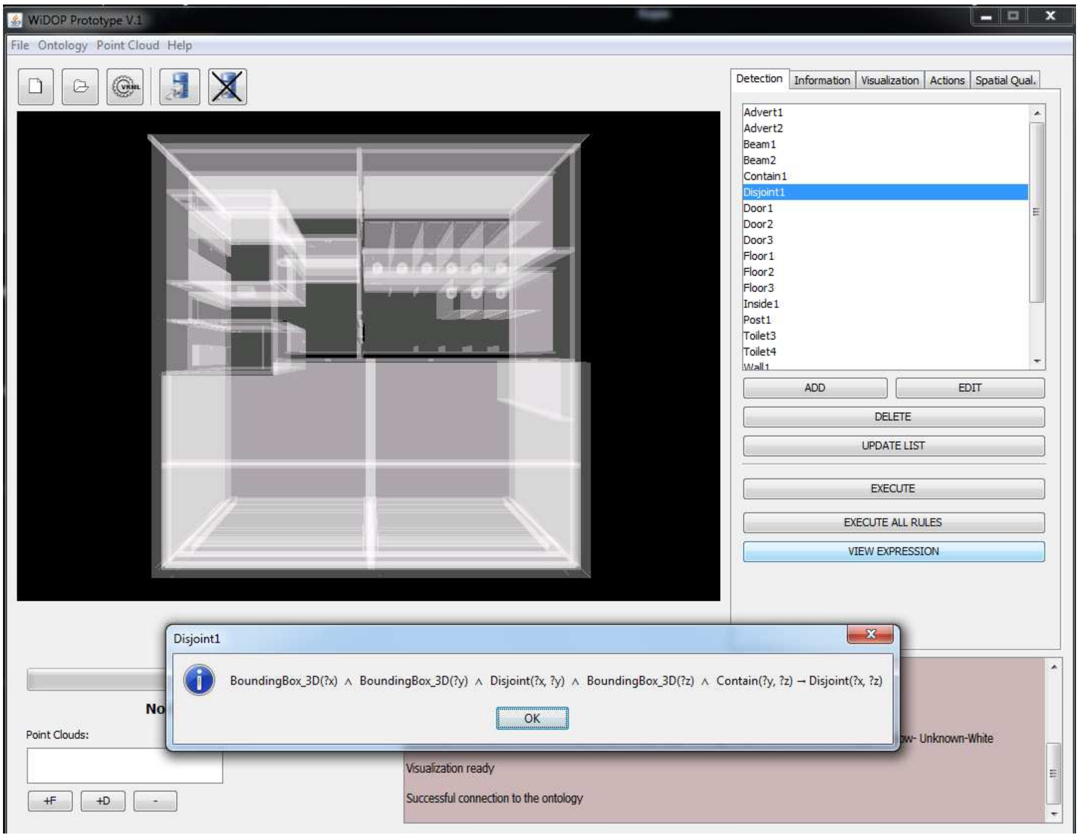


Figure 5-5. The 3DSQ platform interface

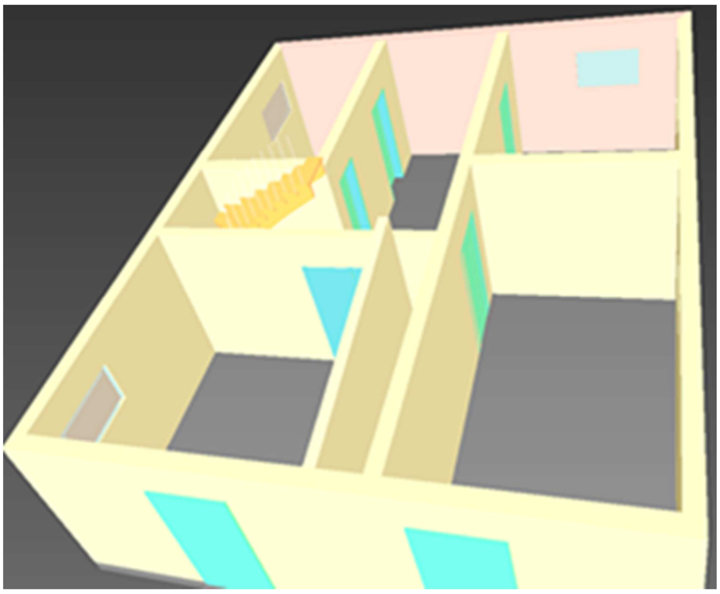


Figure 5-6. The visualisation of 3DSQ output

5.3 3DSQ first extension and 3D geometry qualification process

In fact, the 3DSQ platform enables, via the developed interface, to load an OWL ontology structure adjusted with spatial knowledge from one side and to populate its content with a selected set of geometries from another side.

The created spatial built-ins are connected to the presented quantification Engine and allow further qualifying semantic spatial relationships on them. By the present applied area, we will try through the created 3DSQ Platform to make an attempt to enclose the lifecycle of a BIM model through the qualification of the different geometric elements. It mainly consist of affecting a semantic identity to a geometry yielding in the end to a rich ontology structure processing in much more flexible way the content of an IFC files. It serves as Intelligent knowledge base for any Building information model where geometries at the end are represented by a full set of information. It includes not just attribute and relationship but also its identity.

In fact, the example presented will concern Gate 2 of Frankfurt International Airport, Figure 5-7. For this example, it was given an IFC file, Figure 5-9 related to the above mentioned CAD scene, Figure 5-8. The whole Gate 2 of the Frankfurt airport is a product of more than 4000 elements; the scene initially in DWG AutoCAD format (Armstrong et al., 2002) was exported into the IFC format (Vanlande et al., 2008) and independent Object Format files (OFF files) containing just geometric elements. The DWG file does not contain object definitions, which implies that all objects of the building are converted to an IfcBuildingProxyElement. Consequently, the semantics of the 3D scene are lost. The example purpose within this chapter is to prove how the 3DQS platform is able to recover the semantics of the IfcBuildingProxyElement. The IFC file format developed by the International Alliance for Interoperability is an object format which somehow contains the semantic of the scene (Vanlande et al., 2008). It is composed of objects that belong to a class of the schema within the different relationships. For Instance, the “IFCRelConnection” is not defined as a symmetric relation within the IFC file. However, in such a scenario where IFC files are generated from AutoCAD ones, the semantic of the classes and the relationships are not defined. Based on these different data sources, the different observations, about the scene, geometry, and spatial relation can be expressed as far as possible in our knowledge base. In an ideal case, we would therefore know about the semantic of objects (Walls, floors, ceiling,...), the geometry (Position, extension, orientation,...), additional features (roughness, colour, other surface characteristics) and spatial relations (Wall A isOn a Floor B), that would give a good base for a 3D geometric qualification process.

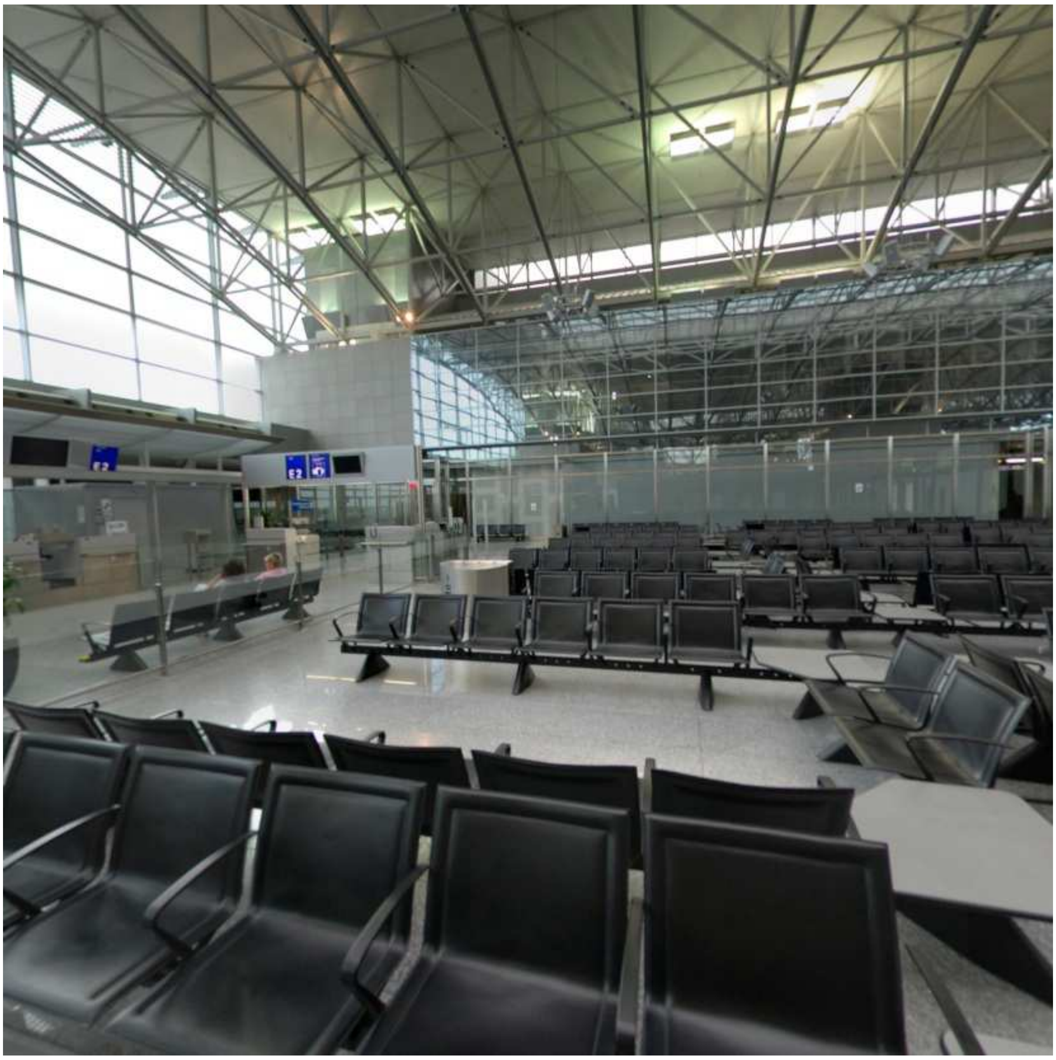


Figure 5-7. Fraport Scene example

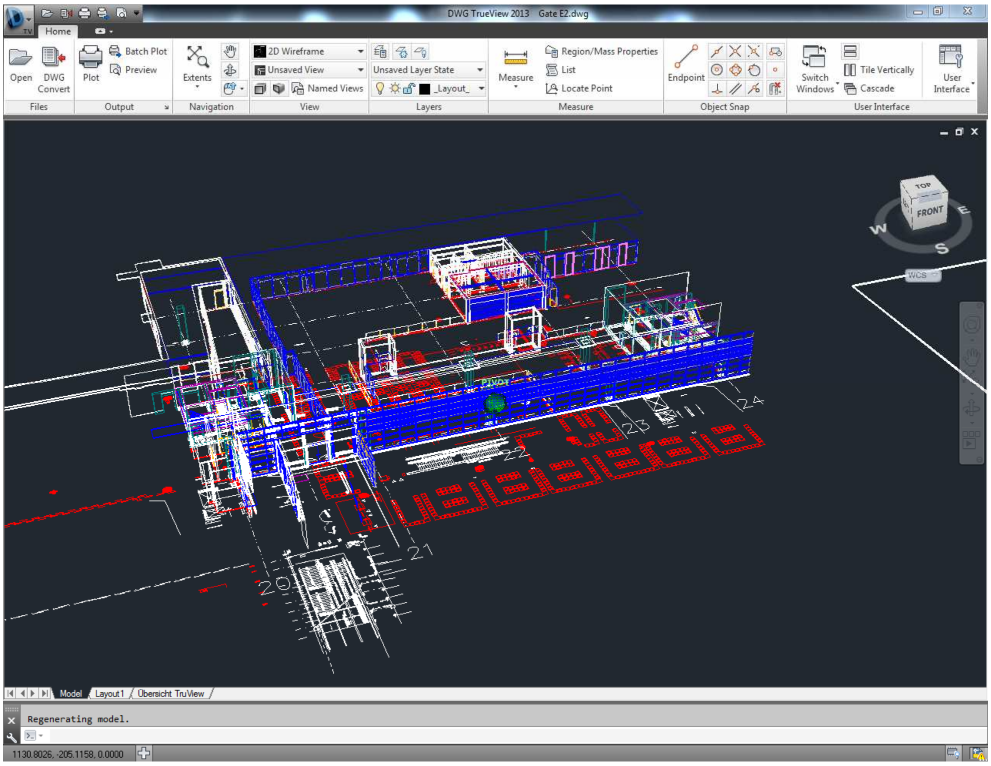


Figure 5-8. The whole gate 2 scene in AutoCAD format in 3D

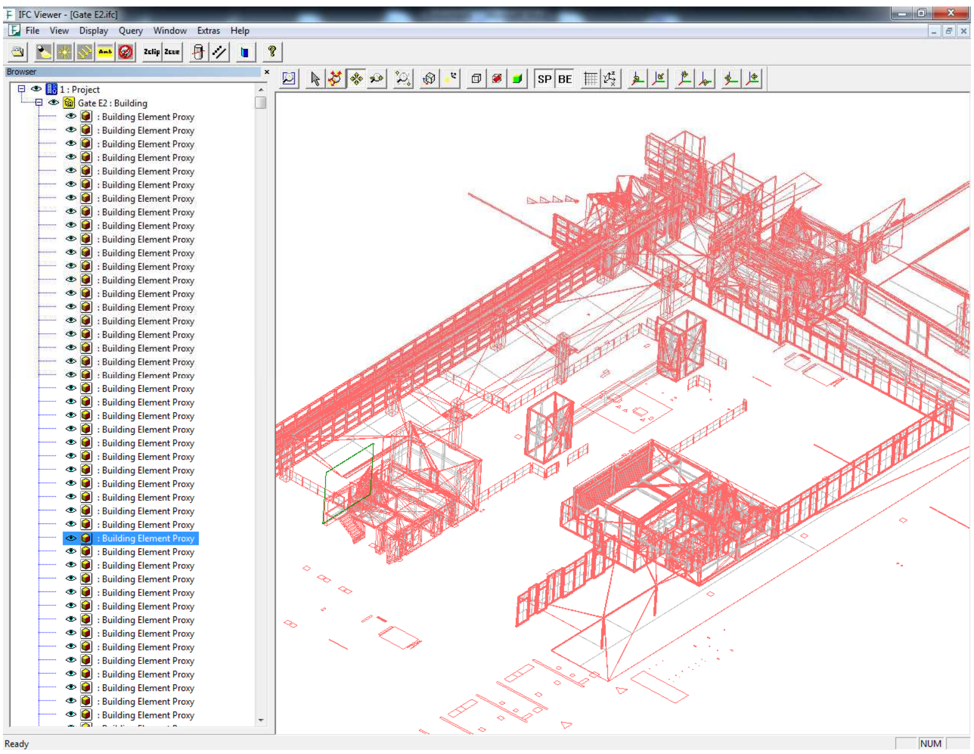


Figure 5-9. The whole gate 2 scene in IFC format

In order to qualify populated geometries in the 3DSQ platform (e.g. IfcBuildingProxyElement), the ontology and its associated geometrical objects have to be adjusted. Actually, the main objective of this process is to affect a semantic quality to the different geometries. The created OWL ontology structure is re-adjusted with new 3D domain concepts to make its enrichment possible with the correspondent 3D geometries, Table 5-1. Regarding the rules, a set of new semantic rules are defined for the 3D geometry qualification process. In fact, geometrical and spatial knowledge play a major role in the characterisation of the different elements inside an architectural scene, where geometries have specific forms, dimensions and relations with the other ones. Such an assumption will encourage the presented 3DSQ mainly to qualify and identify the different geometries, Figure 5-10.

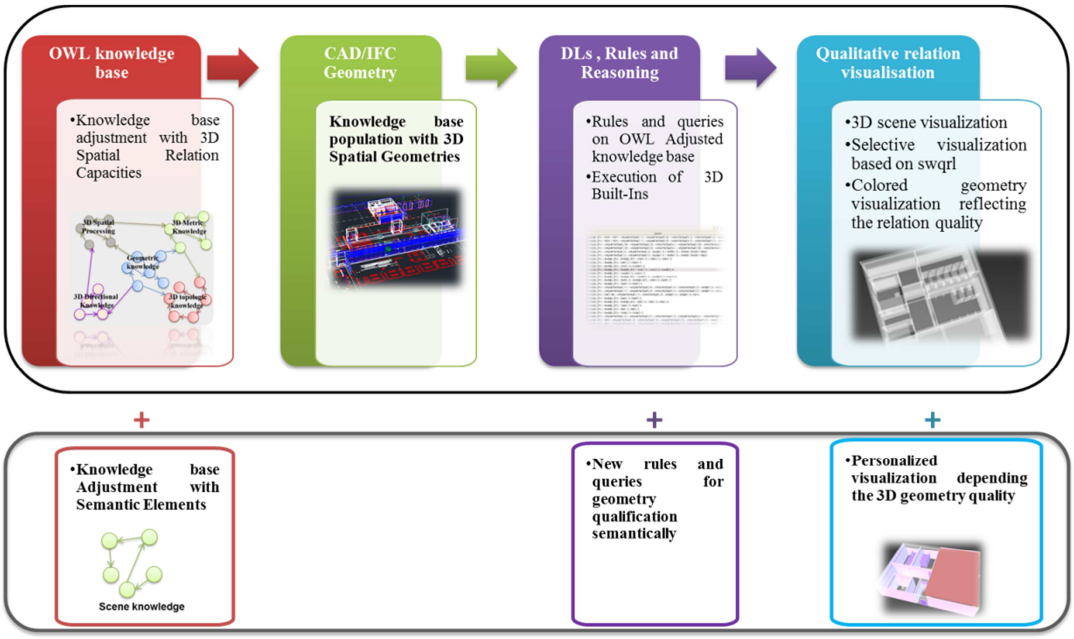


Figure 5-10. The 3DSQ Prototype extension for the geometry qualification task

5.3.1 Knowledge base extension

The extended knowledge should contain all relevant information about objects and elements that could be found within an indoor architectural scene. This could make up a list such as: {Door, Window, Wall, Ceil, Barrier, Post, beams, etc.}. The created knowledge base related to the Fraport scene was inspired following our discussion with the domain expert and our study on its technical drawings. An overview of the targeted elements, the most useful and discriminant characteristics to qualify geometries is presented.

Object	Geometry	Spatial Characteristics	3D spatial relationships	Correspondent image
Floor	Rectangle	**Width**: Larger than 2m **Length**: Larger than 2m **Orientation:** Horizontal	-Perpendicular to Wall -Chair on the floor -Table on the floor -Walls on the Floor	
Ceiling	Rectangle	**Width**: Larger than 2m **Length**: Larger than 2m **Orientation:** Horizontal	-Perpendicular to Wall -Connected to Post	
Door	Rectangle	**Height**: Between 1.7 and 2.7m **Length**: Between 0.4 and 1.5m **Orientation:** Vertical	-Inside a Wall On the Ground -Near to Door	
Window	Rectangle	**Height**: Between 0.4 and 1 m **Orientation:** Vertical	-Inside a Wall -Inside the Ceiling -Rectangle sides lays far from the wall boundary	

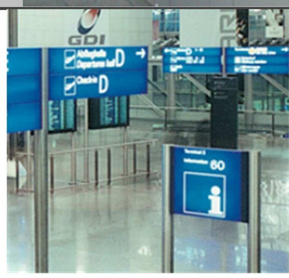
Post	Cylinder Rectangle	**Height**: Larger than 2,5 m Width: Between 0.1 and 1.3m **Length**: Between 0.1 and 1.3m **Orientation:** Vertical	-Connected to Wall -Connected to Ground	
Beam	Rectangle	**Height**: Between 0.05 and 1m **Length**: Larger than 0.5 m **Orientation:** Horizontal	Near the Ceiling	
Toilet	Rectangle	**Height**: Between 0.13 and 0.8m **Width**: Between 0.4 and 0.6m **Orientation:** Vertical	-Overlaps Wall -Near Door	
Advertise ment	Rectangles	**Height**: Between 1 and 3m **Length**: Between 0.5 and 1.5m **Orientation:** Vertical	-Perpendicular to the Floor -Isolated -Boundaries are near to the floor	
Wall	Rectangle	**Height**: Larger than 2 **Length**: Larger than 1.4 **Orientation:** Vertical	-Contain Door -Contain Window -On Floor	

Table 5-1. The Frankfurt airport architectural scene observations

To validate the 3DSQ Platform mainly within its quantitative 3D spatial operators, a various number of geometries were created from the IFC selected scene and populated in the adjusted OWL ontology with 3D spatial operators, Figure 5-11.

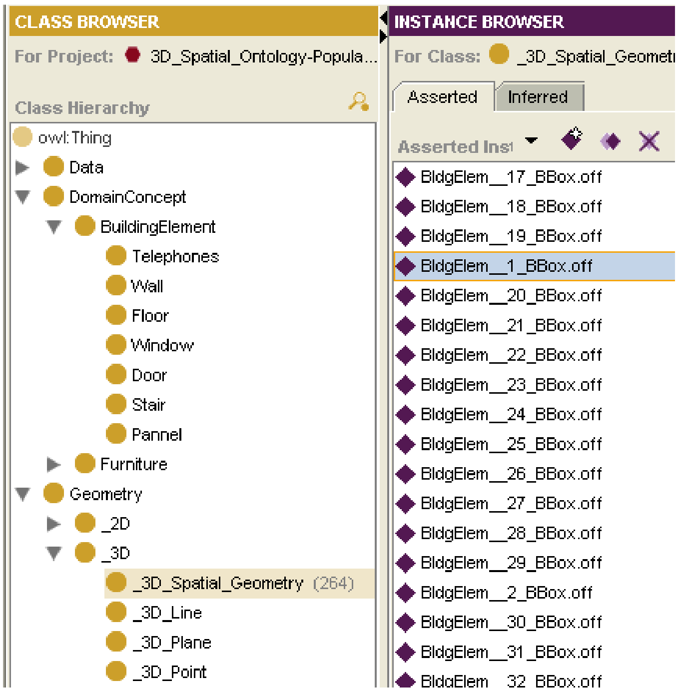


Figure 5-11. The Fraport populated knowledge base overview

As discussed, the basic strength of formal ontology is their ability to reason in a logical way based on Descriptive Logic language DL (Baader et al., 2008). The last one presents a form of logic to reason on objects. Lots of reasoners exist nowadays like Pellet (Sirin et al., 2007), and KAON (U. Hustadt, 2010). Actually, despite the richness of the OWL set of relational properties, the axioms does not cover the full range of expressive possibilities for object relationships that we might find, since it is useful to declare the relationship in term of conditions or even rules. These rules are used through different rule languages to enhance the knowledge possessed in ontology. Within the 3DSQ platform, the domain ontologies are used to define the concepts, and the necessary and sufficient conditions that describe the concepts. These conditions are of value, because they are used to populate new concepts. For instance, the concept "_3D_Spatial_Geometry" can be specialized into "Wall" if it fit the above designed observation. Consequently, the concept "Wall" will be populated with all "_3D_Spatial_Geometry" if they are linked to a "Vertical Orientation" with certain parameters and characteristics. In addition, the rules are used to compute more complex results such as the spatial relationships between objects. For instance, the relations between two objects are used to get new efficient knowledge about the object. The ontology is then enriched with this new relationship. Once the above steps are achieved, the main process related to extracting spatial characteristics and relations from one side, and qualifying the populated geometry from another side, will be accomplished. The presented 3DSQ platform will run according to all the specifications seen before and will output reliable results independently. Once the 3DSQ engine is ready, the platform enables the execution of semantic processing rules with complex 3D Spatial built-ins mainly based on two different languages: SWRL and SQWRL.

5.3.2 Definitions of rules for the geometry qualification process

In the case of an architectural scene, geometric and spatial knowledge play a major role in the characterisation of the different elements inside it, where each geometry has a specific form, dimension and relation with the other ones. Such an assumption will motivate us profiting from the presented 3DSQ mainly to qualify the different geometries. It should be understood that the next examples provide a proof about the platform ability and to which degree we can profit from the qualitative spatial relations. In the next subsections, and to make it more structured, inference on existing geometries and spatial relation can be presented in several ways. Three different scenario examples aim to highlight how the SWRL language and DLs constraints can interact with the OWL ontology adjusted with 3D spatial knowledge to enhance new knowledge and identify the different geometries. To do so, we have developed different types of rules; these rules are divided into topologic, geometric and semantic. Topologic rules perform a test on the spatial relationship of geometries and checks for symmetry, transitiveness and uniqueness. Figure 5-13. Geometric and semantic rules are totally correlated, here we perform several tests to let the knowledge base recognize individuals and assign an appropriate class, Figure 5-12.

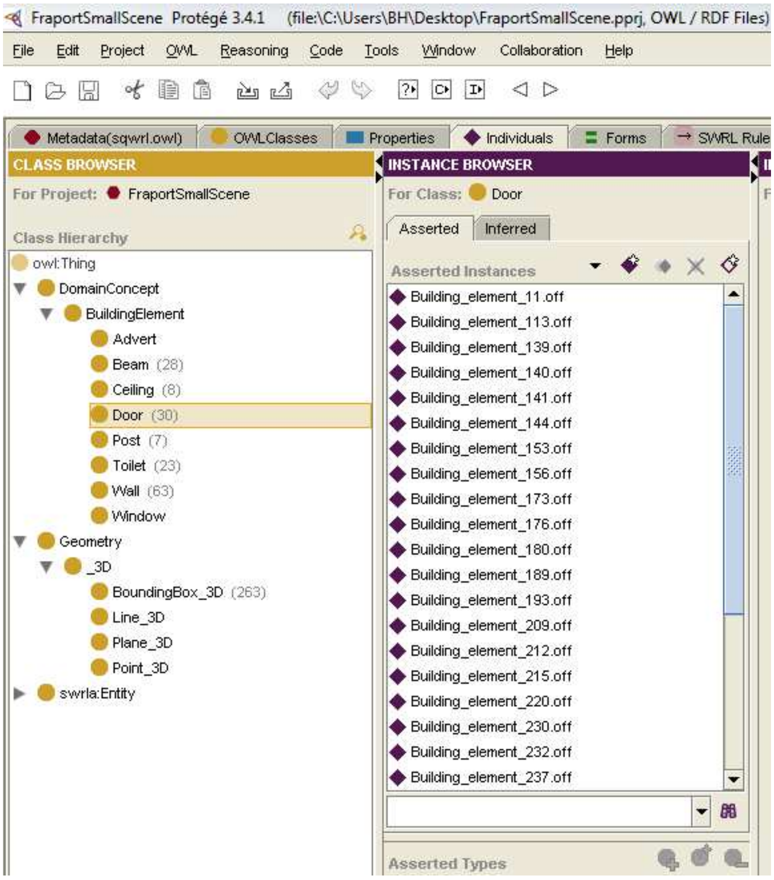


Figure 5-12. The Fraport qualified individual overview

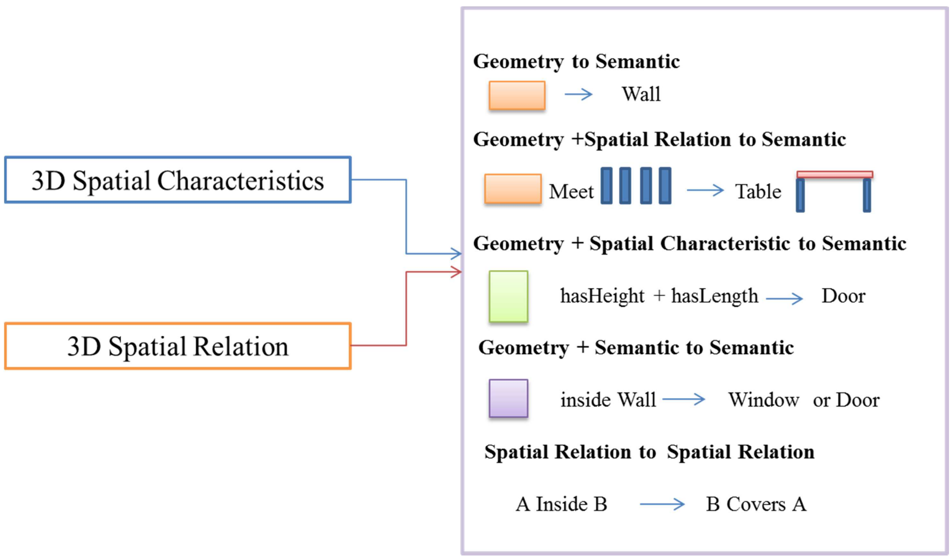


Figure 5-13. Different rules execution

5.3.2.1 *From 3D qualitative geometric characteristics and qualitative spatial relation to Semantic elements*

As a first main kind of rule, the semantic qualification of geometries can be done in the simplest case just with references to the spatial characteristics of elements, basically its height, length, orientation… The next rules qualify horizontal geometries with certain characteristics as Ground.

_3D_Spatial_Geometry (?x) ∧ swrl_Charac_hasSurface(?x,?s) ∧ swrlb:Greaterthan (?s,100) ∧ hasOrientation (?x, Horizontal) → Ground (?x) (28)

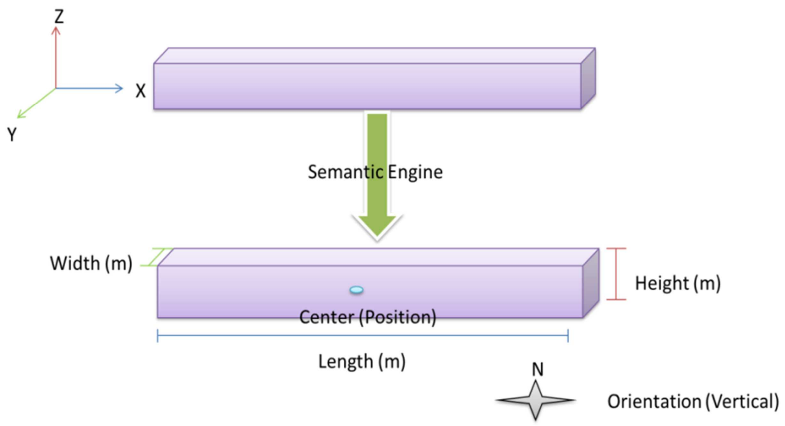


Figure 5-14. New built-ins to extract 3D geometries physical characteristics

To do so, new spatial built-ins (swrl_Charac_hasSurface) able to extract the physical characteristics of the objects are created, Figure 5-14. The characteristics are necessary in order to run semantic rules and for a better understanding of the geometry. While executing the created built-ins, the prototype enables the extraction of the built-ins on the target geometry (?x) and the population of the returned characteristics in the knowledge base.

5.3.2.2 *From geometric characteristics and spatial relationships to semantics*

In several cases, spatial knowledge has a main impact and improvement on the qualification tasks. Giving reference to the above rules, the qualification process of the geometries based on spatial characteristics has lots of drawback since it is too risky to decide about the elements nature just based on its spatial geometric characteristics. In close scenarios, verifying the object context would make the qualification mode robust.

_3D_Spatial_Geometry (?x) ∧ hasSurface(?x,?s) ∧ swrlb:Greaterthan (?s,50) ∧

hasOrientation (?x, Horizontal) ∧ _3D_Spatial_Geometry (?y) ∧ hasOrientation (?y,Vertical) ∧ hasSurface(?y,?s1) ∧ swrlb:Greaterthan (?s1,20) ∧ swrl_topo:Meet(?x, ?y) → Ground (?x) (29)

5.3.2.3 *From geometric characteristics, spatial relationships and semantics to semantics*

To make it more robust and to profit from the already qualified geometries, the next kinds of rules rely on different knowledge to qualify the target elements. The next example infers that height elements inside a Wall could be logically inferred as a Door:

Wall(?x) ∧ _3D_Spatial_Geometry (?y) ∧ swrl_topo:inside(?x,?y) ∧ hasheight(?y,?h) ∧ swrlb:greaterThan(?h,3)→ door(?y) (30)

Figure 5-15 and Figure 5-16 shows the final created SWRL rules, where its executions will results in the qualification of the scene geometries.

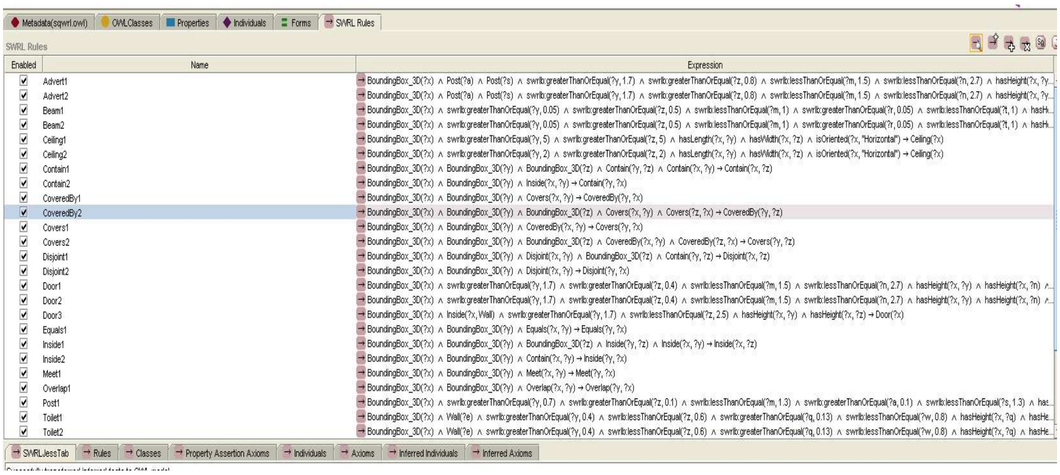


Figure 5-15. Semantic Rules in Ontology

The examples presented in this section aim to raise the issues for the Spatial Relation integration within a relatively new semantic technology. Initially addressed to treat heterogeneity in the web technology, the knowledge engineering technology is more and more deduced for a collaborative approach between humans and machines where human intelligence and reasoning are injected inside actual solutions.

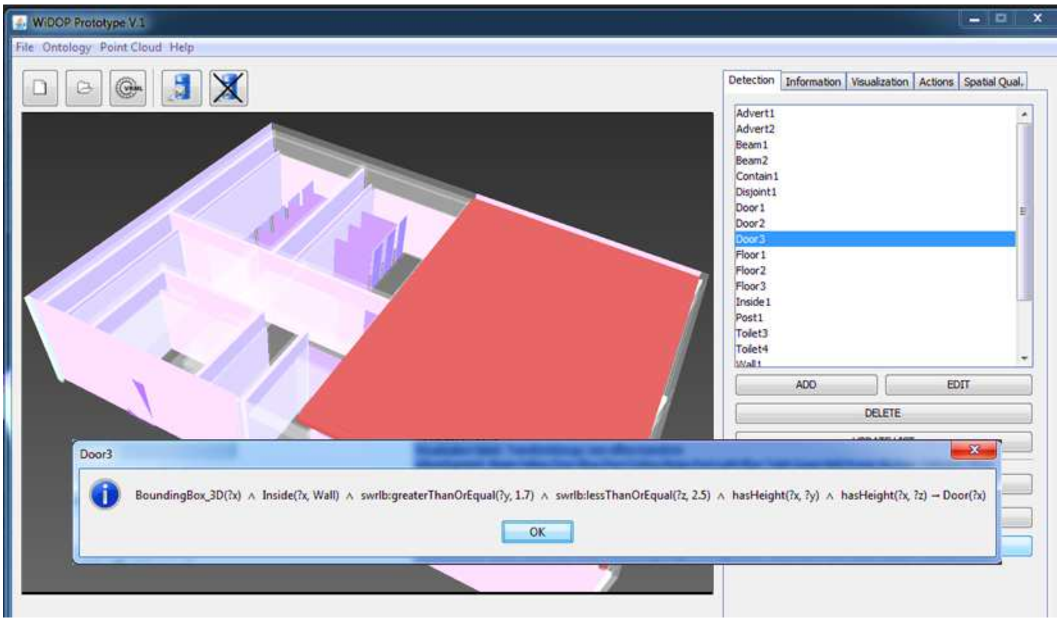


Figure 5-16. 3DSQ Rules Execution

5.3.3 Produced results and knowledge visualization

The created 3DSQ platform gives the opportunity to load 3D geometries (CAD/IFC/OFF…), populate the ontology, and execute Built-Ins and rules achieving the different programmed tasks. Once loaded, a new scene individual is created. It aims at creating a semantic environment where the different geometry can be located, like the individual “Gate2” in our case. Once done, the created CAD geometries are loaded as individuals in the ontology knowledge base that belong to the 3D geometry class. The created scene will be visualized based on the owl ontology individuals already populated as seen in Figure 5-17, Figure 5-18. Added to that, it enables to export populated scene elements such as VRML files for visualisation purposes where the 3D scene and the spatial qualification results are presented in different colours depending on the semantic of the objects

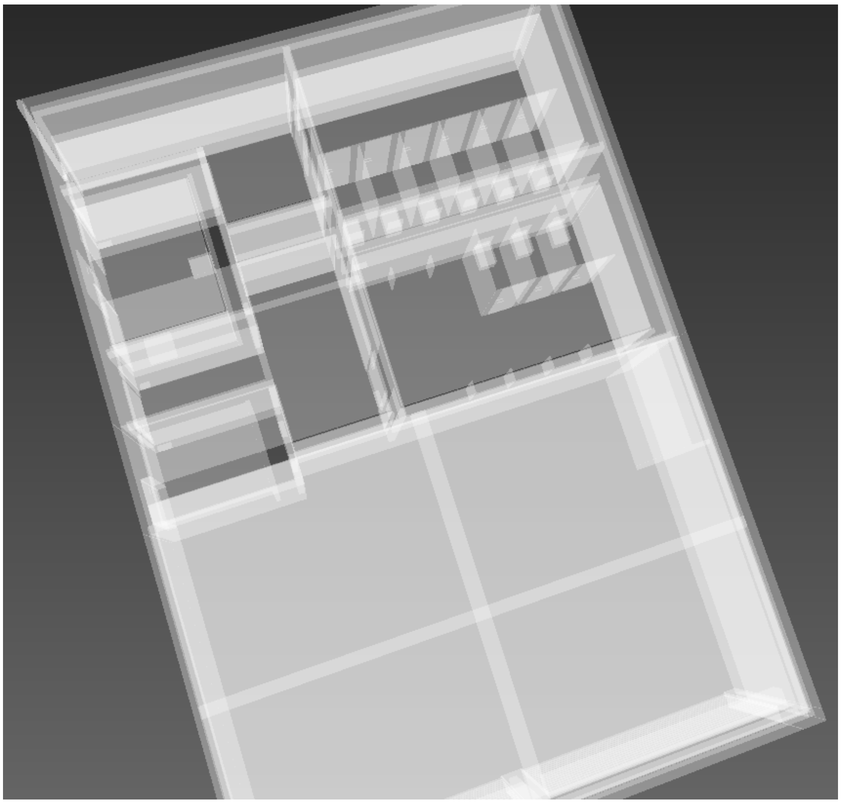


Figure 5-17. The Fraport scene before the qualification process

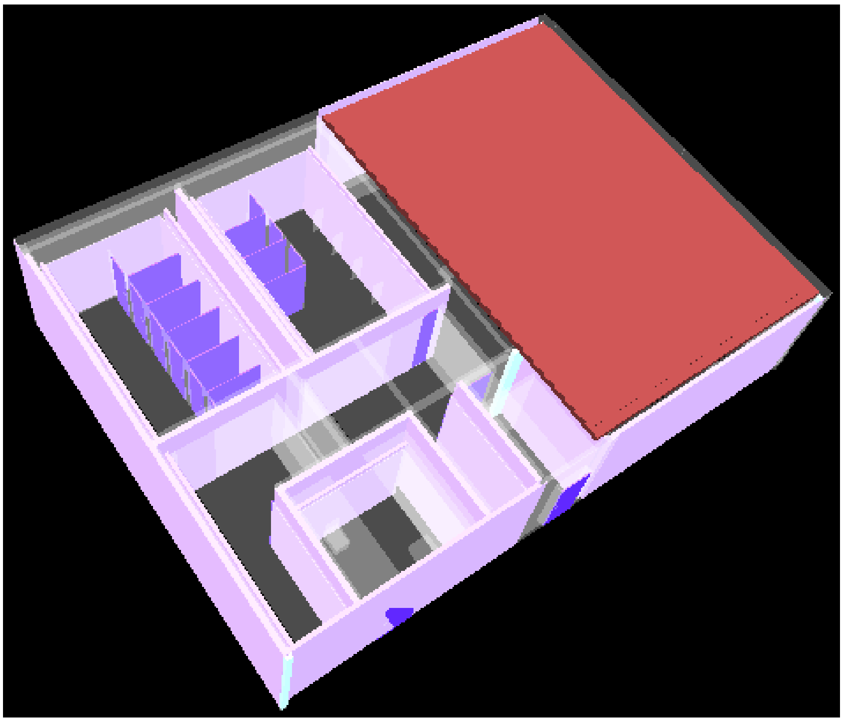


Figure 5-18. The Fraport scene after qualification process

5.4 Conclusion

This chapter has discussed and showed how a step forward can be taken from the qualification of spatial relations to maximize the use of such a 3D spatial knowledge mainly in the architectural domain. Via the different subsections, a demonstration of the possibility to reach the high level advancement suggested by the literature is done. In addition, this chapter shows how to make it optimal and accurate from one side, and mainly to extent new semantic knowledge related to 3D spatial relationships. From the other side, it shows the qualification process via the resulted relationships. The suggested flexible innovative solution to perform object qualification in 3D data makes use of available knowledge in a specific domain or scene. This prior knowledge has to be modelled in ontology, representing a basis for decisions processed during the object detection. Semantic rules are used to control the 3D spatial relation qualification, to annotate the 3D geometry, enrich the knowledge base and drive the inference of new relation, characteristic and semantic objects. The presented solution offers a flexible conception for different application scenarios, for example, for updating existing plans or reconstructing buildings based on standard “building knowledge”.

References

Armstrong, C., Monaghan, DJ., Price, MA., Ou, H., Lamont, J ., 2002. Integrating CAE concepts with CAD geometry. In Engineering computational technology. Edinburgh, UK. pp.75--104.

Baader, F., Horrocks, I., Sattler, U., 2005. Description logics as ontology languages for the semantic web. Mechanizing Mathematical Reasoning, pp.228--248.

Baader, F., Horrocks, I., Sattler, U., 2008. Description logics. Foundations of Artificial Intelligence, 3, pp.135--179.

Ben Hmida, H, Cruz, C, Nicolle, C., Boochs, F., 2012. From 9-IM Topological Operators to Qualitative Spatial Relations using 3D Selective Nef Complexes and Logic Rules for bodies. In KEOD 2012, 4th International Conference on Knowledge Engineering and Ontology Development. Barcelona, Spain, 2012.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2012. From Quantitative Spatial Operator to Qualitative Spatial Relation Using Constructive Solid Geometry, Logic Rules and Optimized 9-IM Model. IEEE International Conference on Computer Science and Automation Engineering, vol 3, pp.453- 458.

Borrmann, A. & Rank, E., 2009. Topological analysis of 3D building models using a spatial query language. Advanced Engineering Informatics, 23, pp.370--385.

Borrmann, A., Van Treeck, C., Rank, E., 2006. Towards a 3D spatial query language for building information models. Montreal, Canada, 2006. Proc. of the 11th Int. Conf. on Computing in Civil and Building Engineering.

Eastman, C.M., Teicholz, P., Sacks, R. & Liston, K., 2008. BIM handbook. Wiley Online Library.

Fraport, 2012. Fraport AG | de. [Online] Available at: http://www.fraport.de/content/fraport-ag/de.html [Accessed 17 September 2012].

Hallberg, D. & Tarandi, V., 2011. On the use of open bim and 4D visualisation in a predictive life cycle management system for construction works. Electronic Journal of Information Technology in Construction, 16, pp.445--466.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission, 21, p.79.

OConnor, M. & Das, A., 2009. SQWRL: a Query Language for OWL. In 6th International Workshop on OWL: Experiences and Directions (OWLED 2009). Chantilly, VA, United States. Vrije Universiteit Amsterdam.

Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y., 2007. Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web, 5, pp.51-53.

U. Hustadt., 2010. KAON2. [Online] Available at: http://kaon2.semanticweb.org/.

Vanlande, R., Nicolle, C., Cruz, C., 2008. IFC and building lifecycle management. Automation in Construction, 18, pp.70--78.

W3C, 1995. VRML Virtual Reality Modeling Language. [Online] Available at: http://www.w3.org/MarkUp/VRML/.

Chapter 6

Application of Spatial Analysis to Geometry Detection and Qualification in 3D point cloud data

6.1 Introduction

After adding the 3D geometry qualification capacities to the 3DSQ platform, it will be extended through this chapter to a knowledge-based detection and qualification approach of objects based on the Semantic Web technologies. In fact, the purpose behind is to share our experience regarding the creation of a 3D semantic facility model out of unorganized 3D point clouds and geometries (Ben Hmida et al., 2012). Thus, a knowledge-based detection approach of objects using the OWL ontology language is presented as a second extension of the 3DSQ platform. This knowledge is used to define the scene elements and a suitable manner for their detection and qualification semantically. In fact, the already extended 3DSQ prototype for semantic 3D spatial relation qualification will be re-extended to support a variety of input, mainly 3D point clouds added to BIM and IFC geometries already discussed, and produced as output a populated ontology corresponding to an indexed scene visualized within VRML language.

Such a problematic is located in the context of the WiDOP project: knowledge-based detection of objects in point clouds. The goal is to develop efficient and intelligent methods for an automated processing of terrestrial laser scanner data. The principle of the WiDOP project is a knowledge-based detection of objects in point clouds data (Ben Hmida et al., 2011) for AEC (Architecture, Engineering and Construction) (Rezgui et al., 2010) engineering applications using the IFC format (Bazjanac, 2008). In contrast with existing approaches, the project consists in using prior knowledge about the context and objects. This knowledge is extracted from databases, CAD plans, Geographic Information Systems (GIS) (Chang, 2010), and technical reports or domain experts. Therefore, this knowledge is the basis for a selective knowledge-oriented detection and recognition of objects in point clouds. The WiDOP project is funded by the German government. However, the partners are the Frankfurt Airport manager company (Fraport) (Fraport, 2012), the German railway company (Deutsche Bahn) (Bahn, 2012), and the Metronome company (Automation, 2012) who is specialized in 3D point cloud processing. As a main motivation, the Deutsche Bahn's main concern is the management of the railway furniture. Currently, the environment of the railway is constantly changing while the cost of keeping these plans up to date is increasing. The present-time solution adopted by the Deutsche Bahn (DB) consists on fixing a 3D terrestrial laser scanner on the train and to survey the surrounding landscape (Railway, signals and green trees on the borders). Metronome automation is a DB subcontractor specialized in 3D data processing. This partner takes the survey point clouds as input and detects the different existent elements manually helped with a 3D process such as signal detection. The main objective of the Deutsch Bahn project consists in detecting automatically the objects in the 3D point clouds to feed the position and the semantic definition of objects into a GIS system. In the next section we will be presenting the adaptation process for the already extended 3D Spatial Qualification approach (chap 3, 4 and 5). We will present in general the main ideas and the suggested solution to the problematic of object detection and qualification in 3D data. In a second time, we will be demonstrated through detailed case studies related to the Railway scene, having as main data set a 3D point cloud and presenting almost a linear scene with very specific domain vocabulary.

This chapter is structured as follows: An overview of the relevant literature on the topic is presented in section 2. The proposed solution and the built knowledge based on 3DSQ base will be outlined in section 3. Section 4 is dedicated to the Railway context as a use case to our knowledge-based strategy for object detection and qualification. Finally, the conclusion and future issues are discussed in Section 5.

6.2 Background on detection strategies

The problematic of 3D object detection and scene reconstruction, including semantic knowledge was recently dealt with within different domains - photogrammetry (Pu & Vosselman, 2007), construction and robotics (Rusu et al., 2009)… Modelling a 3D survey, in which a low-level point cloud or a geometry surface representation is transformed into a semantically rich model, is done through three main tasks. The first one is the data collection, in which dense point measurements of the facility are collected using laser scans taken from key locations throughout the facility; Then, there is data processing, in which the sets of point clouds from the collected scanners are processed. Finally, survey modelling, in which the low-level point cloud is transformed into a semantically rich model. Knowledge processing is achieved via modelling geometric knowledge, qualifying spatial relations (Cantzler, 2003), and finally assigning an object category to geometry (Boochs et al., 2011).

In current practices, the creation of a facility model is largely a manual process, performed by service providers who are contracted to scan and model a facility where projects may require several months to be achieved, depending on the complexity of the facility and the modelling requirements. Ideally, a system could be developed that would take a point cloud of a facility as input and produce a fully qualified as-built model of the facility as output. According to the literature (Vosselman & Dijkman, 2001), two major approaches of object detection and qualification exists nowadays: the data-driven approach and the model-driven one. The first class of approaches relies on the automatic data processing by using different segmentation techniques for feature extraction (Rusu et al., 2009), where new techniques presenting an improvement compared with the described ones, by integrating models and information networks to guide the reconstruction process are presented within the second class of approaches (Andreas, 2005). In the next section, a survey on the different works on each approach will be highlighted and discussed.

6.2.1 Data Based strategies

The data-driven approach, also called the non-parametric modelling approach presents a technique that attempts to model a 3D point cloud scene by a sequence of more or less complex operations. These operations enable the generation of an information model without relying on a specific library aiming primarily at geometry detection. It presents the process of constructing simplified representations of the 3D shape for survey components from point cloud data. In general, the shape representation is supported by CSG representation (Corporation, 2006) or B-Rep one (Xu et al., 2007). Once geometric elements are detected and stored via a specific presentation, the final task within a facility modelling is the object qualification. It presents the process of labelling a set of data points or geometric primitives extracted from the data with a named object or object class. In the meantime, an important processing aspect refining the segmentation quality has appeared, particularly when dealing with data-driven approaches using artificial intelligence. It is based on a learning process and has to do with enforcing the robustness of such methods, so as to recognize the complex objects. In a typical paper, (Lee et al., 2008), object segmentation and classification are obtained through a learning procedure employing Markov Random Fields and quadratic programming. Another method proposed by Spinello et al (Spinello et al., 2010) and enables the classification of more complex objects based on a diverse set of features incorporated within the framework of associative Markov networks for training. However, such methods generally require a large number of training data sets in order to obtain good results. As a first impression, the data driven approach, mainly based on numerical processing, used to ignore all important information that can cause a better detection and qualification. In the meantime, and with the exponential increasing of the point clouds volume and scene complexities, such methods are becoming more and more useless. Improvements for the automatic processing and facility model creation can be expected from new strategies relying more and more prior information related to the target scene. Such information can be modelled within semantic networks, formal grammar, learning process, and ontologies, all combined with numerical processing and classification.

6.2.2 Model based strategies

Early 3D processing techniques were purely data-driven, exhibiting obvious limitations with the increasing complexity of the data and scene. Despite the robustness and efficiency of such processing algorithms, they alone cannot resolve existing ambiguities when qualifying objects in a digitized scene. Recently, new progress has been achieved by considering the use of prior information on the target scene, materialized through models approximating the geometrical characteristics of objects and the general scene architecture. Such prior information has been materialized through several techniques and technology wavering based on their ability to present the nearest model picture of the reality.

6.2.2.1 *Semantic graph-based approach*

First improvements based on semantic networks used to guide the reconstruction process have seen the light, like the work of Cantzler et al. (Cantzler et al., 2002), and Scholze et al. (Scholze et al., 2002) where certain architectural features like the orientations of a wall, for example, are used through semantic networks to detect and qualify geometries. First, architectural features are extracted from a triangulated 3D model, then constraints are generated out of the scene by matching planes against a semantic of the building mock up by a backtracking research tree. In this step, the semantic network concentrates on the definition of the 3D objects and the relationships among them. Constraints such as parallel or perpendicular to a wall are exploited. Finally validated constraints are applied enabling the extension and updating of the original model. Scholze et al. (Scholze et al., 2002), has extended this work into a model based reconstruction of complex polyhedral building roofs modelled as a structured collection of planar polygonal faces. The modelling is done in two different layers, one focuses on geometry whereas the other on rules by semantics. Concerning the geometry layer, the 3D line segments are grouped into planes and further into faces using a Bayesian analysis. The preliminary geometric model is subject to a semantic interpretation in the second layer. The knowledge gained in this step is used to infer missing parts of the roof model by invoking the geometric layer once more to adjust the overall roof topology. This work exemplarily shows the potential of semantic rules taking relations between certain characteristics into account. Although the used rules are simple, semantic tools meanwhile offer a broad framework to combine geometrical, topological, factual and logical aspects. Always in the context of semantic network, Andreas el al (Nuchter & Hertzberg, 2008) has presented an important semantic map creation approach for robot systems. As an input, they have make uses of 3D laser range and reflectance data. Assuming having a 3D geometry model of the scene, it´s interpretation refers to the process of labelling large meaningful structures in the 3D geometry model. Such structures would typically be represented by points in the model, and a large number of them at that. Examples are walls, floor, and ceiling inside a building where Walls are characterized by a flat shape and perpendicular orientation for example. Once the semantic elements are defined, the related planes are extracted via the Random Sample Consensus (RANSAC) algorithm (Chum & Matas, 2008) and then labelled. Once planes are detected, a generic model of an indoor scene is implemented as a constrained semantic network used to qualify detected objects, where nodes represent different plane types in a building and relations among them are encoded using different connections.

Rusu et al, (Rusu et al., 2009) investigate the following computational problem: given a 3D point cloud model of an environment, how is it possible first to segment the point cloud into sub segments that correspond to relevant objects and then to label the segments with the respective category label. The presented solution includes two components: the Semantic 3D Object Map which contains those of the environment and a Triangulated Surface Map continuously updated. The Semantic Object Map is built by classifying a set of planar regions with estimated 3D geometrical features, and serves as a semantic resource for an assistant mobile personal robot, while the Triangulated Surface Map supports 3D collision detection and path planning routines for a safe navigation and manipulation. The hybrid semantic object map in this work is comprised of two different types of maps where the first one presents a static semantic map comprised of the environment including walls, floor, ceiling, and all the objects which have utilitarian functions in the environment, such as fixed kitchen appliances, cupboards, tables, and shelves, which have a very low probability of having their position in the environment changed. As far as feature-based object recognition is concerned, some of the same approaches have been used in both 2D images and 3D data. For instance, Vosselman et al (Rutzinger et al., 2009) (Elberink & Vosselman, 2009) made use of higher level 3D features, mainly simple roof shapes that are generally present in building structures. The authors relied on the use of the 3D Hough transform to detect planar roof faces in point clouds, and hence to reconstruct the scene in a higher level of abstraction. The segmentation strategy was based on detecting intersecting lines and height jump edges between planar faces. While qualifying geometries, the author relied on graph matching techniques especially with incomplete 3D segmented data. However, the results were not satisfying when the data did not clearly describe the object, either in the presence of noise or because of occlusions. In others scenarios, Pu et al. (Pu, 2009) reconstructed building facades from terrestrial laser scanning data. Knowledge about size, position, orientation and topology is used to recognize features and also to hypothesise the occluded parts. In a similar paper, (Lee et al., 2010), a model-based reconstruction method was proposed. In this method, semantic knowledge is also used to infer missing parts of the roof and to adjust the overall roof topology. These approaches use knowledge to evaluate results from numerical processes, but do not integrate it into the processing as such.

As a conclusion for the presented approach, it relies more on static calculation and not on the semantic decision. In fact semantic network presents a directed graph, involving nodes and relations between nodes, where the structure of the network defines its meaning. Although network notations are easy for people to read, there is no formal semantics for such a presentation structure as there is in logic one for example. Likewise, it enables presenting static information without being able to develop constraints or rules which make its use for the detection and qualification task restricted to static definition of the theatrical model. Not far from the semantic graph and networks, some other approaches aim to describe hierarchically the attributes of an object based on semantic grammar allowing the manipulation of more generic rule systems compared to static networks. In this field, Teboul, et al (Teboul et al., 2010) has segmented the building facades using a tree to interpret procedural geometry, and connected grammar semantics and images using machine learning. This approach proposes a dynamic way to perform searches through a perturbation model. Likewise, Ripperda, et al (Ripperda & Brenner, 2007) also extracted building facades using structural description, and used Monte Carlo

Markov Chains (Brooks et al., 2011) to guide the application of derivation steps during the building of the tree. Although there is reasonable advancement in the field of prior information and knowledge modelling and uses in the field of object detection and qualification in data, the way information is presented is still far from real interpretation. The same goes for the manner that human analyse scene is still too ideal compared to the discussed literature where most approaches rely on bottom-up strategy detecting geometry, and mapping them to semantic schema in a later stage.

6.2.2.2 *Ontology-based approach*

A new vision of the human observation modelling was created while dealing with the semantic web and the ontology web language. Such mature technology presents the best solution to truly present human observation. In this field, not just objects and relations are statically presented by links, but constraints about them are defined, and rules are created. It results in a rich dynamic model where inferring on existing axioms is one of its based advantages. Although there is a reduced number of authors' looking on its impact on the target problematic, a big loss of the ontology capability is observed where they still consider it to be a semantic graph. In this field, Markus Eich et al (Eich & Kirchner, 2010) aim at the generation of semantic maps. Their works includes labelling metric maps which are provided by 3D point clouds. They have proposed an ontology-based description of an indoor environment and a probabilistic reasoning approach based on spatial feature descriptions. To enrich such a purpose, they have suggested a semantic classification based on object primitives. First, they introduced spatial feature descriptors which can be mapped directly to a symbolic level where spatial entities can be defined directly using domain knowledge and ontologies (Antoniou & Harmelen, 2009). In a first step, laser data is acquired using a tilting laser setup or 3D light detection and ranging (LIDAR) system (Alexander et al., 2009) and matched to an existing point cloud model. In a second step of the scene-recovery process, geometric information is extracted from the merged point cloud data. They achieved it by using 2D plane extraction or the direct extraction of 3D primitives. Some common surface reconstruction methods include the ball pivoting algorithm (Stelldinger, 2008) and the Delaunay triangulations (Bose et al., 2011) are also used. Once the shapes have been recovered from the unorganized point cloud, the goal is to classify the structure the robot perceives and to label the structure with semantics. To make semantic labelling possible in indoor environments, they have made use of some basic assumptions and consider a probabilistic likelihood function since, for instance, two shapes can be parallel with the certainty of 0.9 due to noise and rounding differences in the extraction process. Maillot et al (Eich & Kirchner, 2010) used a visual concept ontology composed of visible features (such as spatial relations, colour and texture) to recognise objects through matching among numerical features and visual concepts. Duran et al (Durand et al., 2007) proposed a recognition method based on an ontology which has been developed by experts of the domain; the authors also developed a matching process between objects and the concepts of the ontology to provide objects with a semantic meaning. However, knowledge in these approaches has not been fully exploited; other capabilities, such as guiding and controlling all the process through various level of knowledge have not been explored.

6.2.3 Conclusion and Discussion

This previous research shows that there have been various attempts at making the analysis of 3D data more robust and efficient. In this area, simple models are efficient and robust, but have limitations for more complex objects. Statistical methods are able to handle more complexity, but they also need large training efforts and are difficult to transfer. Information and Knowledge based methods, however, seem to have the potential to manage even more complex scenarios. Successful work uses geometric object characteristics for their identification, or tries to map the structure of a scene into a semantic framework, while other work introduces knowledge into the processing and allows the use of various characteristics of objects in order to improve their detection. Building on the above results, significant improvements have been brought to the processing of 3D data through additionally incorporating semantic aspects. In the meantime, the ability to exploit semantic knowledge is limited when the number of objects becomes large, requiring an adequate way for structuring properties of and relationships between objects. In fact, the presented methods for survey modelling and object recognition rely on knowledge about the domain. Concepts like “Signals are vertical” and “Signals intersect with the ground” are encoded explicitly through a set of rules. Such rule based approaches tend to be brittle and break down when they are tested in new and slightly different environments. Additionally, regarding the literature, people model the context by specifying the concepts and the relationships of objects to describe the world. However, no one mentions the knowledge about the 3D processing algorithms and the associated results such as geometry and spatial relation.

Based on these observations, flexible representations of facility objects and more sophisticated guidance for object detection by modelling algorithmic, geometric and spatial knowledge within an ontology structure based on the 3DSQ approach will present the way of a significant improvement. Actually, it will allow the 3DSQ process to control the object detections and to dynamic analysis the spatial relation and characteristics of the scene. It also guarantees an automatic detection and qualification of objects in 3D point clouds, materialized via the semantic qualification process. As a conclusion, and moving from traditional approaches, we present the knowledge driven approach to process the 3D point cloud. In fact, we plan to ensure a semantic interpretation of physical objects aiming at qualifying geometric elements semantically or verifying their existence based on the available knowledge. This chapter aims at developing a fully automatically semantic framework controlling and managing the different actors using the Semantic Web technologies.

6.3 3DSQ platform second extension and Object Detection and Qualification in 3D point Clouds Data

6.3.1 System Overview

The problem of automatic object reconstruction remains a difficult task to realize in spite of many years of research. Efficient strategies therefore have to be very flexible and in principle need to model almost all factors having impact of the representation of an object in a data set. This leads to the finding, that at first a semantic model of a scene and the objects existing therein is required. Such a semantic description should be as close to the reality as possible and as necessary to take more relevant factors into account, which may have impact on later analysis steps. At least this comprises the objects to be extracted with their most characteristic features like geometry, shape, texture, orientation... and relations among each other. This knowledge base will act as a basis for further extraction activities and has to work in cooperation with numerical algorithms and real data. This means to make use of the flexibility of knowledge processing for decisions and control purposes to manage data. Even a propagation of findings from processing results into new knowledge for subsequent steps should be possible, which would give a completely new degree of dynamics and stability into the evaluation process. Consequently a further knowledge base has to be developed which characterizes algorithms, their relation among each other and their relation to the scene knowledge. As a result, the processing will be no longer guided by numerical or geometrical processing and their results, but by a complete knowledge base comprising all available semantics, including defining objects via the scene knowledge, object knowledge, Spatial knowledge, algorithmic knowledge, and suggesting how they can be detected and qualified through semantic rules, description logic constraints and inference engines. Figure 6-1 illustrates the second main extension applied to the 3DSQ platform in order to overcome the new challenges. It presents the adopted strategy applied to the 3D point cloud through the control of prior knowledge about the scene, the 3D spatial relations and the 3D processing algorithmic ones to yield and qualify geometries.

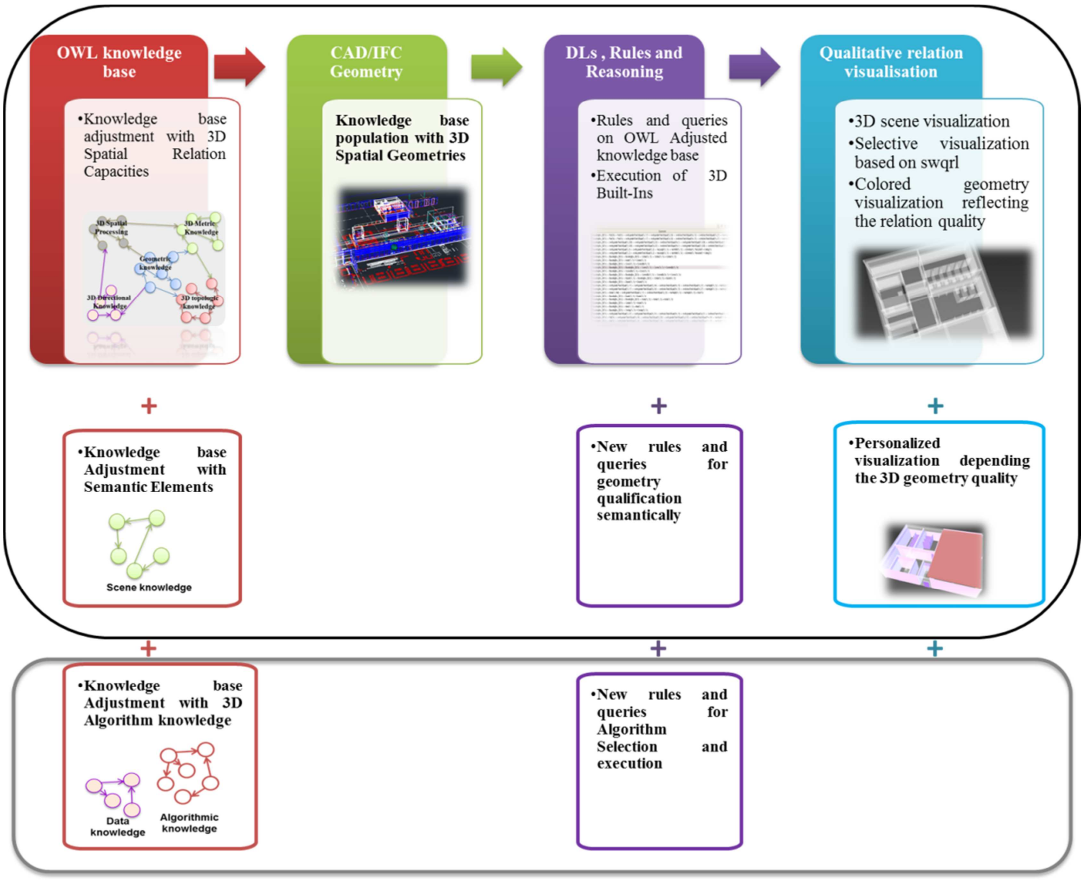


Figure 6-1. The 3DSQ platform with its second extension

As a main contribution compared to the 3DSQ V1, the updated solution takes into account the 3D processing algorithm knowledge and includes the real algorithmic execution to detect geometries in the platform. Once executed, the detected geometries will be populated in the OWL ontology and the SWRL rules and DLs constraint will run to qualify the detected geometry. Figure 6-2 summarizes the applied analysis to the 3D point clouds, but can also be extended to other useful data sources. It is based on explicitly formulized prior knowledge to the scene, on spatial relations of objects and on processing algorithms. It is a multi-stage concept based on three supports: the modelled knowledge (Figure 6-2 left side), the algorithms selection module (Figure 6-2 right side above), the spatial relation qualification (Figure 6-2 right side above) and the semantic qualification engine (Figure 6-2 right side below).

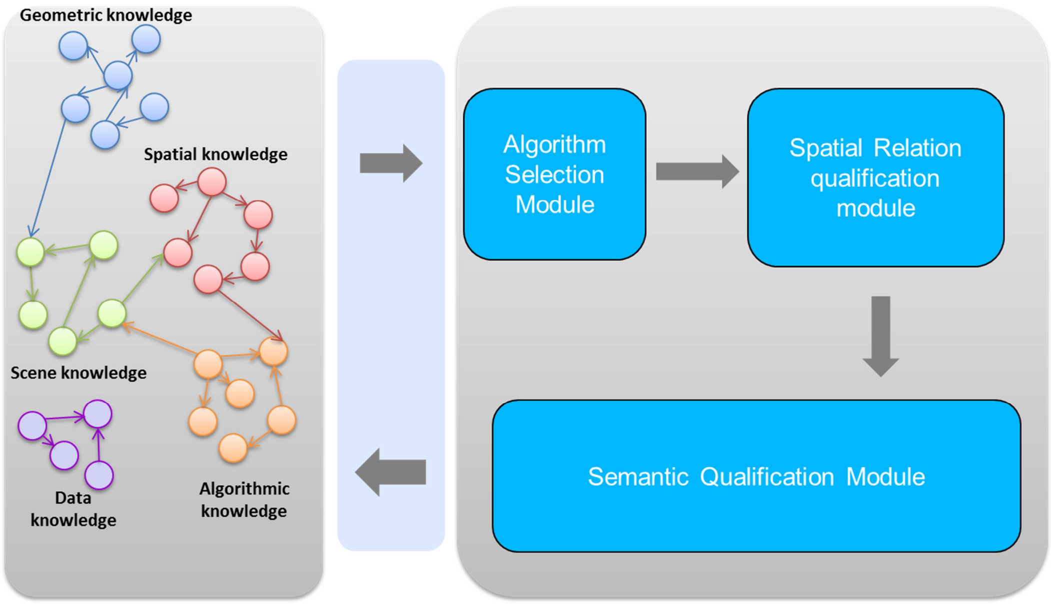


Figure 6-2. The Knowledge-Driven strategy applied to the 3D point cloud data

In the initial stage, the accessible knowledge is transferred into a corresponding knowledge base. Depending on the particularity of the prior knowledge, this base might be simply generic if no real object exists in the scene or it might be more concrete because of already addressed objects which were contained in the scene, Figure 6-3. Starting from this initial stage, an update process begins, which involves the algorithms and the qualification engine. After detecting geometries, these elements are passed to the qualification engine, which then tries, based on the existing knowledge expressed in the ontology, SWRL rules and DLs constraints, to identify the nature or object category of the elements. The result of the qualification step will update the knowledge base by entering newly qualified or updating already existing elements, and then entering the next stage of processing. As soon as no further refinement of the base is achieved, the process ends.

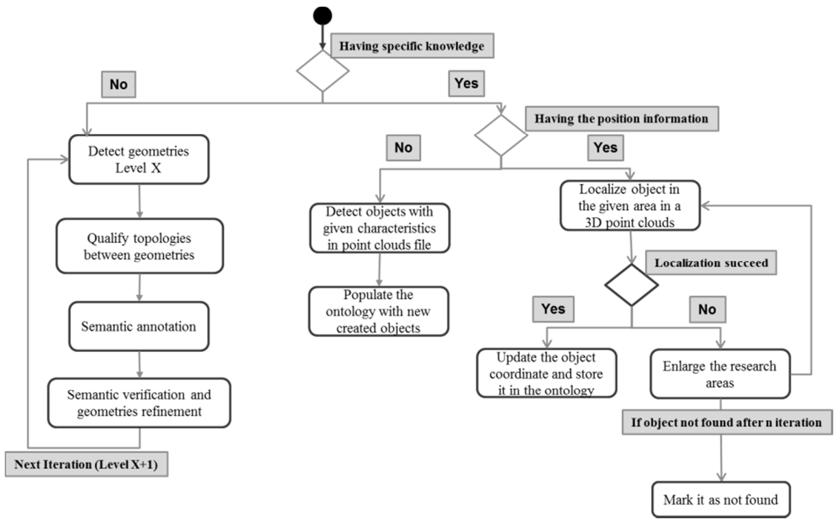


Figure 6-3. Activity diagram in case of specific knowledge and generic one and its impact on the detection and qualification process

Being based on human observation, Table 6-1, and compared to the 3DSQ V1, new knowledge has been added to the semantic framework. In fact, the defined solution relies on different knowledge categories, cooperating together to construct the core of the knowledge base: the Scene Knowledge (SK), the Geometric Knowledge (GK), the Spatial Knowledge (SpK), the Data Knowledge (DK) and the 3D Algorithmic Knowledge (AK). Each field of knowledge is represented by circles in the Figure 6-2 left side, where relations between these concepts are represented by edges. The scene knowledge contains information related to the content of the scene to be processed like important objects and characteristics. Such knowledge is not only important for processing the identification and qualification activities, but will also support the selection and guidance of the algorithmic processing. The geometry knowledge mainly characterizes the elements structure. The spatial knowledge models the relationships among objects in the scene. It presents a main key for the qualification process, since it yields to the objects state disambiguation based on its relation with the common environment. The data knowledge expresses important characteristics of the data itself. Finally, algorithmic knowledge characterises the behaviour of the algorithms and determines what kind of purpose they fulfil, which input is expected, which output is generated, and to which geometries they are designed for. Based on this knowledge, a dynamic algorithm selection is possible, and allows dynamic adaption for processing situations given from other domains, Figure 6-2. Let´s note that the Algorithm selection module, Figure 6-2, will be excluded from the thesis content since it presents an independent work done outside of the present research.

However, all the required knowledge for such a module will be included in the 3DSQ extended platform and the thesis overview. To go into more details, the next subsection deals with the created knowledge base presenting the core of the developed solution, once clarified, section 6.3.3 clarifies the interaction process via the semantic rule system while explaining the overview of the knowledge driven approach.

Object	Geometry	DL Constraints	3D Spatial relations
Electric Born	Vertical Lines	**Height**:Between 4m and 6m **Length**: max 0.5 **Width**:max 0.5 **Lines**:1 or 2 Vertical line	**Contains**: 2 parallel lines **MUST** be connected: to a Small Box **Distant**:50m from Electric Born **Right side** of the Rail

Table 6-1. Example of the knowledge description of an electric Born

6.3.2 Knowledge modelling

To build our targeted rich knowledge base, knowledge of different domains is acquired where sources such as domain experts are the most reliable knowledge source. However, other information sources such as CAD, GIS data, existing digital documents as CAD drawing or IFC files, or other available documents in the case of detailed input are used to extract knowledge. Likewise, the different required algorithmic knowledge is acquired from experts in numerical processing. The needed knowledge for such a purpose will be modelled within a top level ontology describing the general concept behind the knowledge domain. The suggested approach is intended to use semantics based on OWL technology for knowledge modelling and processing using classes, instances, relations and rules. Where an object within the 3DSQ can be modelled as presented; a room has elements composed of walls, a ceiling and a floor. The sited elements are basic objects. They are defined by their geometry (plane, boundary, etc.), features (roughness, appearance, etc.), and also the qualified relations between them (adjacent, perpendicular, etc.). The object "room" gets its geometry from its elements, and further characteristics may be added such as functions, in order to estimate the existent sub elements. For instance, a "classroom" will contain “tables", "chairs", "a blackboard", etc. The detection of the object "room" will be based on an algorithmic strategy which will look for the different objects contained in the point cloud. This means, using different detection algorithms for each element, based on the above mentioned characteristics, it will allow us to classify most of the point regions in the different element categories. It corresponds to the spatial structure of any facility, and it is an instance of semantic knowledge defined in the ontology. This instance defines the rough geometry and the semantics of the building elements without any real measurement.

This section discusses the different aspects related to the created top level ontology structure. It is composed mainly of classes and their relationships. The domain ontology presents the core of this research and provides a knowledge base to the created application. The global schema of the modelled ontology structure offers a suitable framework to characterize the different target scenes. The ontology created is used basically for two purposes:

- To guide the processing algorithm sequence creation based on the target object characteristics.
- To ensure the semantic qualification of the different detected objects inside the target scene through the analysis of the object spatial relation and characteristics.

The ontology is managed through different components of description logics where 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 prefixes for the class. Other advantages include the simplification in defining the resource and solving the ambiguity for different contexts. The hierarchical structure of the top level class axioms of the ontology is given in Figure 6-4, where we find main classes within other data and objects properties able to characterize the 3D scene facility. The main actors that have to be modelled are: processing algorithms, point cloud data or image resources, and target objects with their geometry, spatial relation and characteristics. The *DomainConcept* class represents the different objects found in the target scene and can be considered the main class in this ontology. This class is further specialized into classes representing the different detected objects. This class is the entry point for the adjustment process. In fact, any concept which requires a 3D model has to inherit the properties from this class to be able to benefit from the 3DQS framework. The other classes are used to either describe the object geometry through the Geometry class by defining its geometric component, or to describe its characteristics through the Characteristics class. Ultimately, the algorithms are recommended, based on their compatibility with the object geometry and characteristics via the Algorithm class. The *DomainConcept* class represents the different objects found in the target scene and can be considered to be the main class in this ontology. This class is further specialized into classes representing the different detected objects. This class is the entry point for the adjustment process. Actually, any concept which requires a 3D model has to inherit the properties from this class to be able to benefit from the 3DQS framework. The other classes are used to either describe the object geometry through the Geometry class by defining its geometric component, or to describe its characteristics through the Characteristics class. Ultimately, the algorithms are recommended based on their compatibility with the object geometry and characteristics via the Algorithm class.

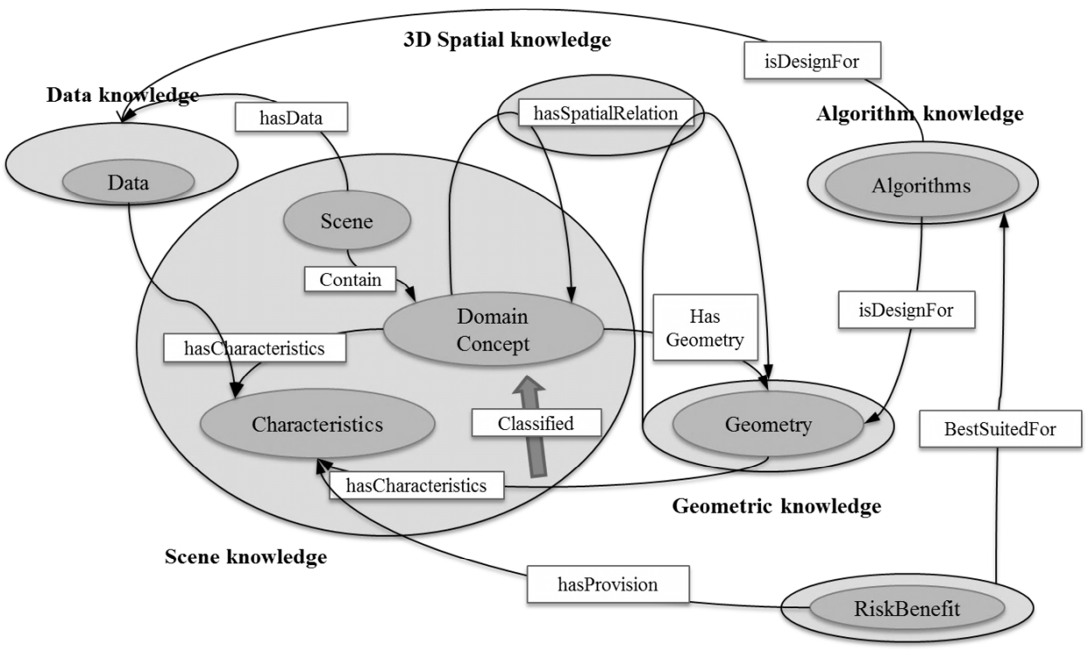


Figure 6-4. The General ontology schema

6.3.2.1 *Scene Knowledge*

The scene knowledge contains all relevant information about the objects and elements which might be found within a real scene that can vary from the architectural domain, for example, to the Railway. They are used to fix either the main scene within its point clouds file through attributes related to the scene class, or even to characterize detected element with different semantic and geometric characteristics. The created knowledge was inspired due to our discussion with the domain expert and with our theatrical study. An overview of the targeted elements, the most useful and discriminant characteristics to detect them and their inter-relationship is presented in the next part. The scene knowledge contains all relevant object elements which might be found within that scene and mainly composed of Object knowledge and Characteristics knowledge.

The object Knowledge is described in the schema of ontology and includes semantics of the objects, such as properties, restrictions and relationships. The more information about an object that is created and used, the more accurate the detection and qualification process is. In case of buildings, this might comprise a list like: {Building, Wall, Door, Window, Ground, …}. For a railway scene for example, a railway signal is one of the most important elements within the scene where we find Main_signals and Secondary_signals. For qualification purposes, for example we define a signal as:

$$Electric\_Born \sqsubseteq Geom{:}VerticalBB \sqcap \exists\ hasheight.\{>6\} \sqcap \exists\ hasDistanceFrom.DC{:}Electric\_Born\{>50\} \tag{31}$$

The above cited concepts are extended by relations to other classes or data. As an example, the data property Geom:has_position aims to store the placement of the detected object. To specify its semantic characteristics, new classes are created, aiming to characterize a semantic object by a set of features like colour, size, visibility, texture, orientation and its position in the point cloud after detection. To do so, new object properties axioms like Geom:has_Color, Geom:has_Size, Geom:has_Orientation, Geom:has_Visibility and Geom:has_Texture are created linking the DC:DomainConcept class to the Charac:color", Charac:size, Charac:Orientation, Charac:Visibility and Charac:Texture classes axioms respectively.

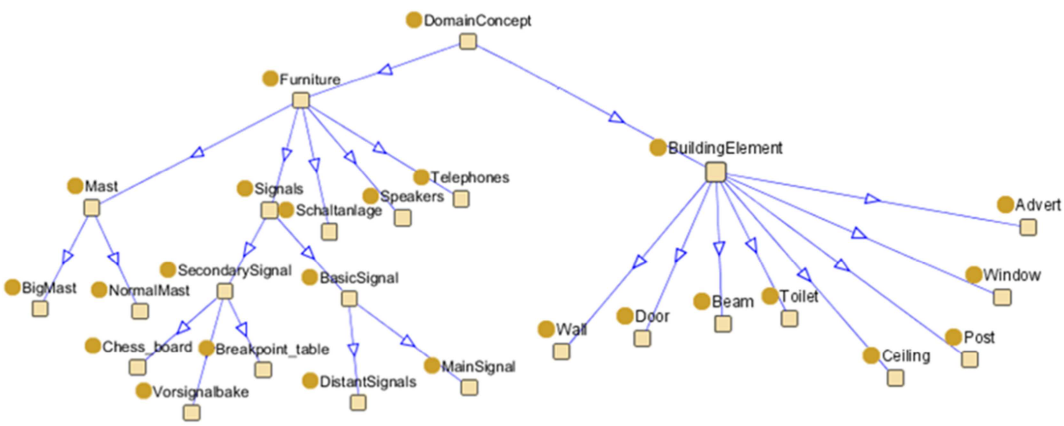


Figure 6-5. An example of scene object modelling

Figure 6-5 shows a possible collection of scene elements. They may be additionally structured in a hierarchical order, as might be seen convenient for a scene.

6.3.2.2 *Geometric knowledge*

Geometrical knowledge formulates geometrical characteristics of a physical property for scene elements. In the simplest case, this information might be limited to a few coordinates expressing the object position. However, for elements to be accessible to functional descriptions, additional knowledge will be mentioned. A signal, for example, has vertical lines, which needs to be described by a line equation and its values, and completed by width and height. In fact, we think that such knowledge can present a discriminant feature able to improve the automatic qualification process. For this reason, we opt to study the different geometric features related to the cited semantic elements, and then, only use the discriminant one as basic features for a given object.

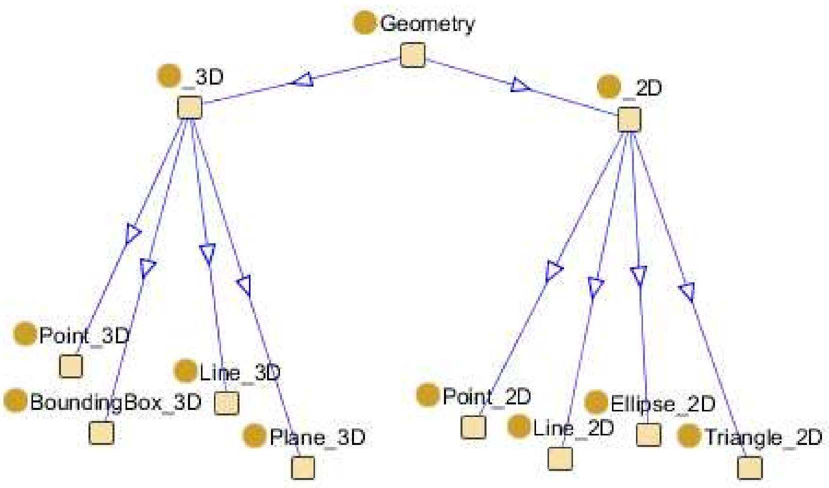


Figure 6-6. The hierarchical structure of the Geometry class

Figure 6-6 presents information about the different geometric elements composing a semantic object, like plane, line, sphere and others. For example, a wall has a planar geometry; moreover, a table consists of planar and linear geometries. Each one of the cited object classes can be described by a lot of characteristics defined in the Geometry class. The last cited one presents other classes capable of modelling the different characteristics that can be used in this context (size, shape, visibility, orientation and texture …)

6.3.2.3 *3D spatial knowledge*

As seen in the Chapter 4, and as the base for this chapter and the previous one, 3D spatial relation knowledge is used to enhance the qualification process. Information about how objects are dispersed in a 3D scene makes the detection and qualification easier. For instance, given the detection of a wall, there is more chance that a door or window will be detected within it. In fact, 3D spatial knowledge includes standards like the 3D topologic knowledge, 3D metric knowledge and 3D processing knowledge. Each one of the cited spatial knowledge contains a variety of relations modelled on the ontology structure. For example, the top level ontology is designed to include spatial relationships. This is then used to enrich an existing knowledge base to make it possible to define relations between objects in a specific case. At a semantic view, topological properties for example describe adjacency relations between classes. For example, the property Topo:isParallelTo allows characterizing two geometric concepts by the feature of parallelism. Similarly, relations like Topo:isPerpendicularTo and Topo:isConnectedTo will help to characterize and exploit certain spatial relations and make them accessible to reasoning steps. The purpose of this class is to spatially connect *Things* presented in the scene and in the geometry class.

6.3.2.4 *3D processing knowledge*

Regarding the numerical processing algorithm, its effectiveness depends on the quality of the data (resolution, noise), the characteristics of the object that need to be detected or other factors depending on a specific case. Algorithms are modelled under specialized classes of algorithms, sharing certain taxonomical and relational behaviours. The hierarchical representation of the algorithms is addressed through dividing the algorithms according to the contexts in which they are executed. Classes including “Geometry Detection”, “Appearance Detection”, “Image Processing” and “Noise Reduction” follow such a hierarchal structure. Likewise, relational semantics are represented through properties. In wider terms, there are two types of relationships: one which applies to the geometries that the objects in Domain Concept possess and other that applies against each other. The first category of relationship is used for detecting geometries. The object property “isDesignedFor” maps algorithms to the respective geometries. For example: Line Detection 1 (Ransac) isDesignedFor Lines. The second set of algorithm properties “input/output” are inter-relational properties to connect algorithms together, based on the compatibility of output from an algorithm to the outputs of others. To get more intelligence for the detection and qualification process, it is necessary to adapt processing to certain situations, depending on the data, the scene and the object characteristics. The created concept allows for these interactions, as it is able to automatically change the strategy based on a compromise of quality and risks. A part of the knowledge base is dedicated to risk-benefit factors that have influences on the algorithms, and have been deduced from the simulation's knowledge pattern. Since an algorithm could perform better with given parameters in one setting, and fail to deliver the same quality in other settings, it is important to evaluate the risk-benefit factors of every algorithm with various possible settings. The class "Risk Benefits" includes all of the risks and benefits possible, due to the previously mentioned reasons. The class contains instances such as "Distinct", "Illusive", "Noise", and "Error Detections". These instances are either the risks or the benefits that have influences on the algorithms as a whole, or at least the values of the parameters they contain. The 3D processing algorithmic class contains all relevant aspects related to the 3D processing algorithms. It contains algorithm definitions, properties, and geometries related to each defined algorithm. An important achievement is the detection and the identification of objects, which has a linear structure such as signal, indicator column, and electric pole, etc., through utilizing their geometric properties. Since the information in point cloud data sometimes is unclear and insufficient, the various methods of RANSAC (Tarsha-Kurdi et al., 2007) are combined and upgraded. This combination is able to robustly detect the best fitting lines in 3D point clouds for example. Figure 6-7 presents the Electric pole object constructed by linear elements, ambiguously represented in point cloud as blue points. Green lines are results of possible fitting lines and clearly show the shape of the object that is defined in the ontology. The object generated from this part is a bounding box that includes all inside geometries of the object, and a concept label.

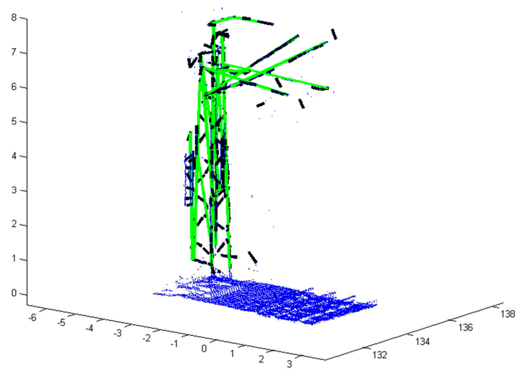


Figure 6-7. The electric pole detection

Next to the 3D expert recommendation, knowledge within the Table 6-1 is created linking a set of 3D processing algorithms to the target detected geometry; the input and output.

Algorithm name	has Input	hasOutput	isDesignedfor	hasSuccessor
Vertical Object Detection	PointCloud	Point_2D	Vertical geometry	None
Segmentationin2D	Point_2D PointCloud	SubPoint Cloud	Vertical geometry	VerticalObjectsDetection
BoundingBox	SubPointCloud	Point_3D	Vertical gemetry	Segmentationin2D
ApproximateHeight	SubPointCloud	Number	Geometry height	Segmentationin2D
RANSAC Line Detection	SubPointCloud	Line_3D	3D Lines	Segmentationin2D
FrontFaceDetection	SubPointCloud	Boolean	Geometry with front face	Segmentationin2D
...	...	...	...	...

Table 6-2. 3D processing algorithms and experts observations

The specialized classes of the Alg:Algorithm axiom represent all the algorithms developed within the extended 3DSQ. They are related to several properties which they are able to detect. These properties (Geometric and semantic) are shared with the DC:DomainConcept and the Geom:Geometry classes: By this way, a sequence of algorithms can detect all the characteristics of an element, Figure 6-8.

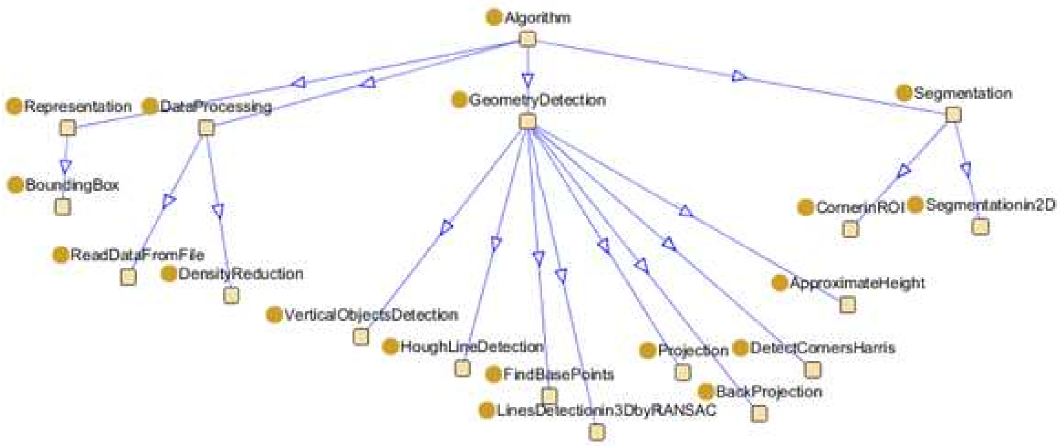


Figure 6-8. Hierarchical structure of the Algorithm class

6.3.3 The WiDOP Knowledge Driven Strategy

In order to manage the interaction between the knowledge part and the different actors (3D Data, processing algorithms, spatial relationship qualification algorithms, spatial characteristic extraction, etc.), the processing capacity of the 3DSQ materialized within semantic rules and DLs constraint will be used. It ensures the control and the management of the knowledge transaction and the decision taken based on semantic rule language and mainly SWRL languages and its extensions through several steps. For instance, the following rule asserts that Geometry with lines higher than 5m is an Electric pole where Electric pole, Bounding Boxes and Lines are all individual-valued properties. The DL syntax related to such an expression is

$$\mathrm{Electric\_Pole} \sqsubseteq \mathrm{BoundingBox}\ \sqcap \exists\ \mathrm{hasLine.Line}\ \sqcap \exists\ \mathrm{hasHeight.}\{>5\} \qquad (32)$$

While the equivalent SWRL rule of such an expression is

BoundingBox(?x) ∧ hasLine(?x,?y) ∧ hasHeight (?y,?h) ∧ swrlb:GreaterThan (?h, 5) (33)
→ Electric_ Pole (?x).

The set of built-ins for SWRL is motivated by a modular approach that will allow further extensions in future releases within 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. These built-ins are keys for any external integration where we take advantages of this extensional mechanism to integrate new Built-ins for 3D processing and spatial processing. Such an extension should help in the interoperation of SWRL rules with other Web formalisms by providing an extensible, modular built-ins infrastructure for Semantic Web Languages, Web Services, and Web applications via allowing the execution of real processing functions and methods inherited from others domains, mainly the 3D processing one and the geomantic one in our case.

6.3.3.1 *Integration of 3D processing operators within the extended 3DSQ platform*

The 3D processing knowledge contains all relevant aspects related to the 3D processing algorithms. Its integration into the suggested semantic framework is done by special built-ins. They manage the interaction between the processing level and the semantic one. In addition, it contains the different algorithm definitions, properties, and the related geometries to the each defined algorithm. An important achievement is the detection and the identification of objects with specific characteristics such as a signal, indicator columns, and electric pole, etc. through utilizing their geometric properties. Since the information in point cloud data sometimes is unclear and insufficient, the Semantic Web Rule Language within extended built-ins is used to execute a real 3D processing algorithm, and to populate the provided knowledge within the ontology. The equation (34) illustrating the "3D_swrlb_Processing:VerticalElementDetection" built-in, for example, was created aiming to detect geometries with vertical orientation. The prototype of the designed Built-in is:

$$\text{3D\_swrlb\_Processing:VerticalElementDetection(?Vert, ?Dir)} \tag{34}$$

Where the first parameter presents the target object class, and the last one presents the point clouds' directory defined within the created scene in the ontology structure. At this point, the detection process will result in geometric elements, representing a rough position and orientation of the detected object. Table 6-3 shows the mapping between the 3D processing built-ins, which is computed and translated to predicate, and the corresponding class.

3D Processing Built-Ins	Correspondent Simple class
3D_swrlb_Processing: VerticalElementDetection (?Vert,?Dir)	Geom:Vertical_BoundingBox(?x)
3D_swrlb_Processing: HorizentalElementDetection (?Vert,?Dir)	Geom:Horizental_BoundingBox(?y)

Table 6-3. 3D processing Built-Ins mapping

6.3.4 Knowledge guidance and Iterative process

Let's recall that the suggested semantic framework for the automatic detection and qualification approach of objects, through the extension of the 3DSQ V1 platform takes as input the 3D point clouds scenes, and an OWL ontology structure presenting a knowledge base to manipulate objects, geometries, and spatial relations, and produces as an output an qualified (annotated) scene within the same ontology structure where detected or already populated geometries are qualified as semantic elements. The presented approach is materialized via an iterative process. It aims to qualify and refines the detected and qualified geometries through the newly gained knowledge at every step of the iteration. Starting from the initial situation, the process iteratively updates the knowledge base (KB) at certain stages. At the beginning of each iteration, the content of the knowledge base is used to detect new features. This might be a new object or a new component of an object. These new feature geometries are then populated in the knowledge framework in order to extend the knowledge base for the next step of qualification. This qualification is performed through the content and the structure of the knowledge base, which has reasoning capacity, based on property restrictions or rule languages, and refines the actual content. This refined content enters into the next iteration. The process is repeated until all entities have been completely annotated, and meets the following convergence conditions:

- All objects defined on the knowledge side are detected and qualified (simple change detection).
- A predefined number of iterations without refinement for any entity are reached.

At the core of the first iteration, the most discriminant characteristics are extracted, such as the vertical elements, in the case of an architectural scene. Such information can control the processing, where algorithms designed for such type of geometry will be selected and executed. Based on SWRL rules enriched with the created 3D spatial Built-

Ins, a detected and populated geometry can be initially roughly qualified where its eventual class can be restricted to one or two eventual candidates, Figure 6-9. This first assumption will help us to detect more discriminant characteristics, guiding us to the final classification. During advanced iterations (Second iteration in this case), more precise geometry is detected and populated in the knowledge base through the enhanced knowledge once the previous iteration is achieved. In general, advanced iterations rely on the ability of the knowledge base to extract the discriminant characteristics pending the final qualification process. Assuming that the main difference between the two above mentioned semantic objects is the existence or not of perpendicular/parallel lines, such new generated knowledge will be verified. In case of the existence of perpendicular lines, the detected object will be finally classified as an electrical pole, Figure 6-9.

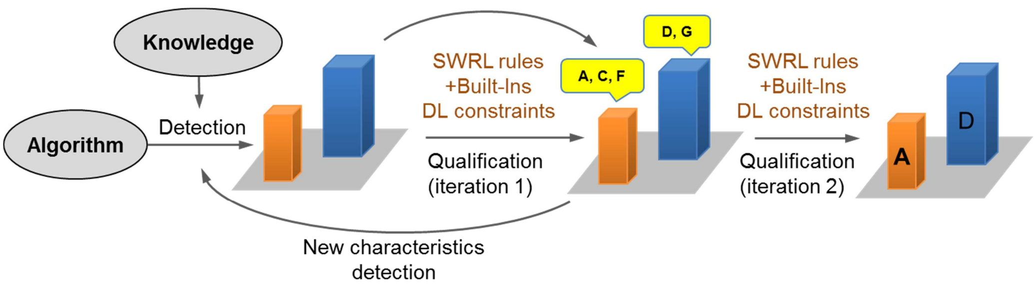


Figure 6-9. Iterative process for knowledge driven approach

In other scenarios, where geometric knowledge is not sufficient for the qualification process, spatial relation (Metric, Topologic, Directional) between the detected geometries has be initially qualified in the knowledge base. In order to combine SWRL rules with spatial operators, news built-ins are defined in order to compute the operator. Consequently, the results of the operators can be used to define queries or enrich the ontology with new spatial relationships between two objects. The following rule specifies that a "BoundingBox" respecting certain characteristics with a distance of 1km from a MainSignal is a "DistantSignal".

MainSignal(?y) ^ BoundingBox(?x) ^ hasHeight(?x, ?h) ^ swrlb:greatThan(?h, 4) ^ (35)
swrlb:lessThan(?h, 6) ^ 3D_swrlb_Topology:distance(?x, ?y, 1000, 10)
→DistantSignal(?x)

The previous section gave an idea on the manner on which the 3DSQ platform can be updated to support the object detection and qualification in several data, especially in the

3D point clouds one. The next question to answer is how the overall detection strategy might be influenced by knowledge and what this means for the design of a practical solution.

6.3.5 Impact of knowledge on the detection and qualification: 2 scenarios

As explained before, knowledge is the key element in this solution and it has to guide and control the process of detection and qualification. It has to be stored and organized in a specific way, in order to be accessible for the reasoning process. One of the aspects that was not considered up to now is how knowledge may guide the 3DSQ execution and to what extent it might be necessary to distinguish different degrees of available knowledge. This will be done in the following section, explaining two major strategies:

- use of well-defined specific knowledge
- use of generic knowledge

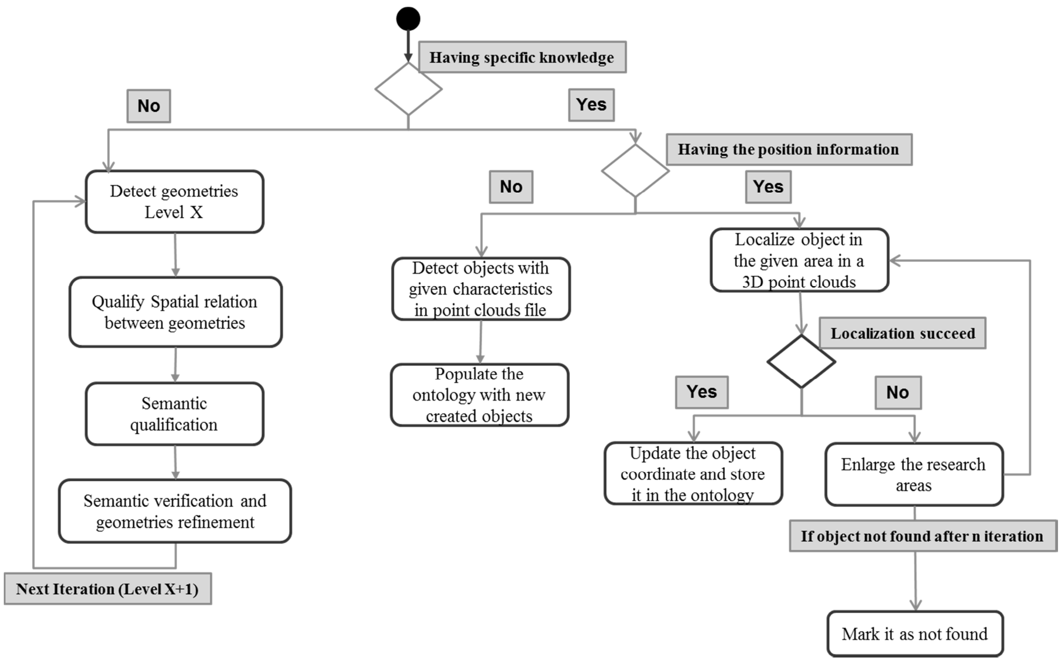


Figure 6-10. The general schema of the knowledge-driven approach

In fact, we have to accept that each individual application case has its own framework of knowledge. The content of such framework changes with the domain to which an application has to be referenced (architecture, industry, civil engineering,) and accordingly, knowledge models to be used must be different. In addition, the framework will be influenced by the amount of knowledge existing in a particular application. This may spread a large field, starting from extensive and actual data bases with more or less precise information, up to just some general ideas to objects in question and without any direct data on the other end. Such large differences in the knowledge base must clearly have impact on the guidance of algorithms and on the strategies used. In principle, the more knowledge existing, the more precisely and directly geometry detection and qualification. That is why there are strategically different concepts following the degree of quality for the knowledge. Hence, we distinguish between sparse knowledge cases (generic knowledge, cf. Figure 6-10 left side) and detailed knowledge cases (specific knowledge, cf. Figure 6-10, right side). Two scenarios that influence the object identification in a point cloud can be identified.

6.3.5.1 *Case of generic knowledge integration of Unknown objects, and unknown positions*

The case of unknown object and position presents the most complex case. In such cases, the type of objects and their positions are both not known beforehand. However, the nature of the scene is already known. In such cases, the algorithmic processing generates geometries that would be used by the knowledge base to check their nature and recommend the object types. This case asks to scan every modelled object in the knowledge base and check for their geometric characteristics. This will help in classifying the geometries found to their respective objects. In this scenario a number of iterations are needed to confirm the objects to the geometries found completely. Looking from a procedural perspective,

Figure 6-11 shows a corresponding strategy. Here, each iteration is composed of four different steps. The first tries to detect basic geometrical elements, which may be part of a physical object (like planes, lines, for example). At this moment, geometry information is available, but it is unclear to which object the elements found may belong. This has to be answered using a different generic logic, as may be derived from spatial relation, for example. Thus, a next step verifies such a relations between detected elements and adds other aspects like orientation (vertical element, horizontal element,...). Based on results from this reasoning a semantic qualification process can be executed in order to obtain an initial mapping between elements derived from the data and the generic semantic. Such a mapping extends the knowledge in the ontology from a generic to a specific one, as real objects have now been created.

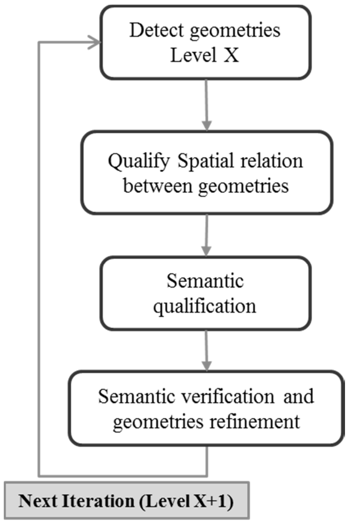


Figure 6-11. Activity diagram in case of generic knowledge

A successful detection then may lead to a subsequent refinement process, enabling to identify less prominent objects, which are smaller or more complex and therefore need more support for identification. This may even be simply based on generic knowledge, providing general concepts to objects and their relation among each other. For example, it is clear that a table has to sit on a ground floor and that chairs may have close adjacency to other chairs or to tables. As a consequence, generic knowledge may guide the detection process in an iterative way, leading from large and significant objects to smaller and more complex ones.

6.3.5.2 *Processing in case of detailed knowledge*

In the case of known objects within its position, the knowledge base supports the algorithmic processing to reconfirm their status and modify the databases if there are any changes in the positions of the objects. This case represents the ideal situation from the view point of existing knowledge. Remaining challenges for the guidance of the processing come mainly from the data to be analysed, possible incompleteness, lack of data quality, for example and the algorithmic knowledge needed to handle such situations. Figure 6-12 presents the adopted strategy in this case using point clouds as a data source.

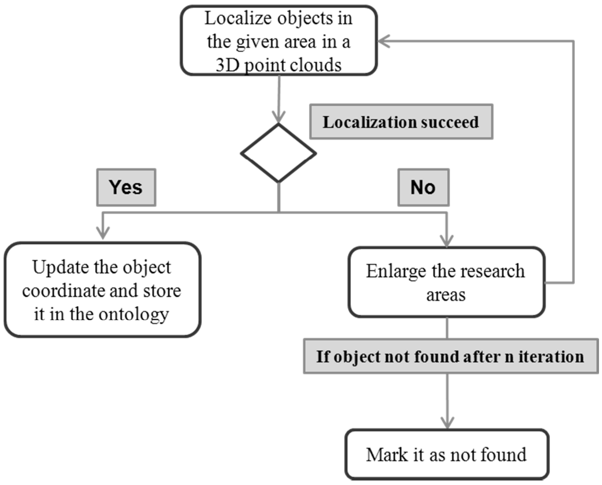


Figure 6-12. Activity diagram in case of specific knowledge with known object position

The first step localizes the target object in the data set (for example, a point cloud) based on previously mentioned 3D_Processing_Built-Ins. It aims to infer knowledge and executes one or more 3D processing algorithms with extracted knowledge from the ontology. Once the localization is done successfully, the object will be stored within its coordinates in the ontology. In case of a failure, the knowledge base has to decide upon the next step, what could be an enlargement of the research area. Such a step would assume that the reason for the failure is due to imprecise geometry data, why the process of localization should be re-executed. Finally, the object coordinate can be updated in case of a successful localization. If not, it will be marked as not found or further rules have to be applied.

In the case of known objects but with unknown positions, the prior knowledge about the type of objects that can be found in the point cloud already exists. However, their exact positions are not known. The knowledge base which provides the scene knowledge interacts with the processing knowledge to detect the objects in the point cloud and derive the positions of the objects. This helps in updating the objects with their corresponding positions in the databases.

6.4 The extended 3DSQ and the Railway uses case

As a partner for the WiDOP project, the German Railway (DB) is one of the world's leading companies in its domain. Its main activities are passenger transport and logistics, infrastructure and services on the German railroad network. The motivation of the Deutsche Bahn Company is the management of railway furniture. Actually, the cost of keeping their plans up to date is increasing. The solution consists of fixing a 3D terrestrial laser scanner on a locomotive, and to survey the surrounding landscape. To do so, the FTI Engineering Network GmbH (FTI, 2012) and Metronome Automation GmbH (Automation, 2012) has developed LIMEZ III (Horn, 2007), a new clearance profile measurement train for the German Railway company. This system records the geometrical data of the track and trackside objects, and even the adjacent track. The measurement system uses state-of-art laser technology in combination with high-speed video techniques, photogrammetry and light sheet technology to produce this measurement data with high precision. One of the results of LIMEZ III is a set of point clouds, covering the respective railway lines and its environment, Figure 6-13.

Figure 6-13. Limez III simulation snapshots

After the first survey, the resulting data will be considered as a reference for comparisons with future surveys in order to detect changes. As a consequence, the company will benefit from an automatic object detection and qualification, because too much data has to be processed, and the amount of data leads to a tremendous management cost. In this field, the updated 3DSQ platform will be exploited to give a hand to the Railway domain need, in order to guide to detection and qualification of objects in 3D point clouds data semantically. At first, a survey of the existent technique adopted by the German railway company will be detailed.

6.4.1 Actual business process

Metronome Automation company, as a subcontractor of the German Railway, they offer several services for industrial measurements. It consists of the design, development and the support of systems to detect geometric entities. In order to achieve this, a new platform was created. In fact, the created Clear Suit platform is one of the most ambitious and important projects developed by the company. It aims to label different types of objects, based on the three dimensions point clouds data. As seen in Figure **6-14**, the correspondent image and front view are shown for each portion of point clouds. By selecting object in the front view, it will be directly localized in the 3D point cloud in the main window. To create a new object, a suitable label has to be affected to the selected area, based on the user observation added to a simple description to clarify any eventual ambiguity of the object, Figure 6-14.

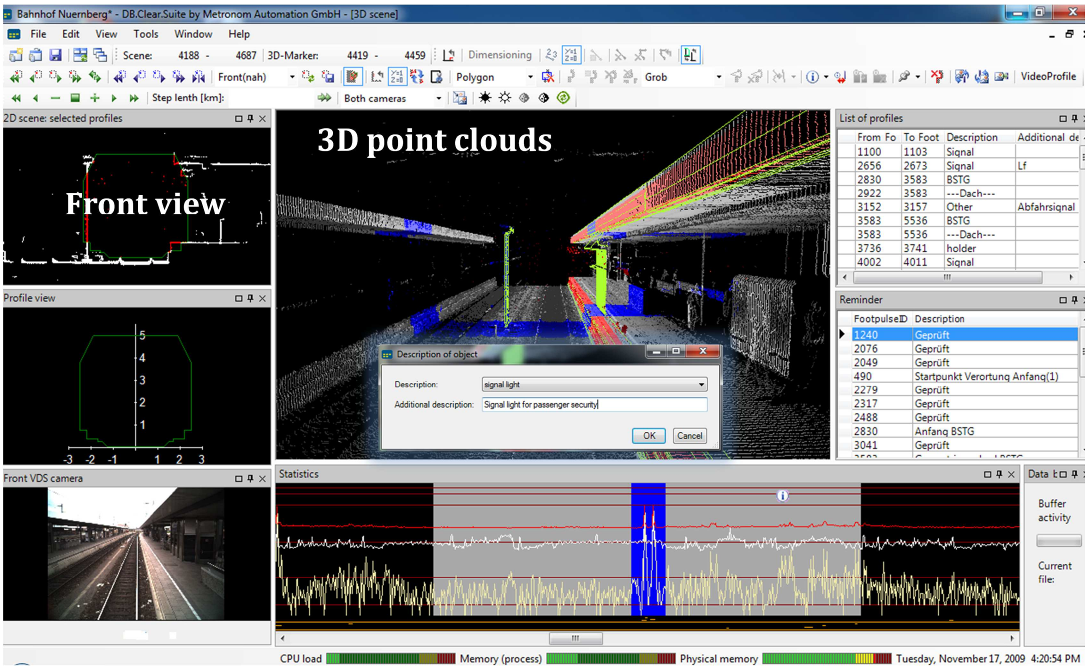


Figure 6-14. Plot screen of the DB Clear Suite software

The DB Clear Suite software uses an encrypted data base for images, video and 3D point clouds. The generated file has an XML structure where each object is created in an independent tag, as seen in Figure 6-15. Unfortunately, until now such software generates a static data base represented by the XML file, (Hunter et al., 2011), where such a data structure can be mapped to ontology, (Cruz, 2008). Figure 6-15 gives a general overview about the generated XML file structure. Each object in reality is presented in a profile characterized by the different attributes cited in Table 6-4. Each profile contains a semantic label showing the tag of the object and a small description, a geometric location or position and finally a picture. The geometric position can be divided into two sub characteristics: the cloud points construct such objects and their geographical location.

Concerning the last one, each object should have a start and end location, and one characterized by the foot pulse and the position. Each profile is characterized by a unique identifier, a description code presenting the label of the object, an object description and some other information mentioned in Table 6-4.


```xml
<profile number="31" descriptioncode="408" description="Signal light for passenger security" invalid="false"
  <geometries>
    <geometry number="1" object="line 1 circle 3">
      <locations>
        <start numberingarea="5900aa" orientation="79.7452" region="12" lineofvision="DescendingStation">
          <footpulse>4560</footpulse>
          <track distance="106180044.625" number="5320" directioncode="1" numberoftrack="" />
          <node startnodedepartment="221937" startnodeidentifier="270" startnodedescription="Nürnberg Hbf Bf" startnodetoken="N
          <position x="4077659.320" y="798727.299" z="4823190.999" system="ETRS89" />
        </start>
        <end numberingarea="5900aa" orientation="79.7452" region="12" lineofvision="DescendingStation">
          <footpulse>4560</footpulse>
          <track distance="106180044.625" number="5320" directioncode="1" numberoftrack="" />
          <node startnodedepartment="221937" startnodeidentifier="270" startnodedescription="Nürnberg Hbf Bf" startnodetoken="N
          <position x="4077659.320" y="798727.299" z="4823190.999" system="ETRS89" />
        </end>
      </locations>
      <points>
        <point number="1" y="-0.003" z="-0.187" x="0" marker="false" type="S" />
        <point number="152" y="0.002" z="-0.202" x="0" marker="false" type="S" />
      </points>
      <deletedpoints />
      <lines>
      </lines>
    </geometry>
  </geometries>
  <deletedgeometries />
  <metas>
    <meta name="simplification">
      <value xsi:type="xsd:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    </meta>
    <meta name="line4print">
      <value xsi:type="xsd:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">/r=0/u=7/</value>
    </meta>
    <meta name="trackgauge">
      <value xsi:type="xsd:int" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">1433</value>
    </meta>
    <meta name="objectpicture">
      <value xsi:type="xsd:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">tmp192F.tmp</value>
    </meta>
  </metas>
</profile>
```


Figure 6-15. A screen shot of the XML data base structure

Profile attributes	Attributes values
Number	"31"
Descriptioncode	"408"
Description	"Signal light for passenger security"
Invalid	"false"
Fromdatabase	"false"
Dateofmeasurement	"2007-06-26T08:42:17.9837527+02:00"
Tps	"0tps"
Recordtype	"0"
Bank	"-1"
Dateofacquisition	"2009-11-17T16:57:39.2071686+01:00"

Dateofchanging	"2009-11-17T16:57:39.2071686+01:00"
Agentofacquisition	"Helmi"
Agentofchanging	"Helmi"
Program	"LaserProfile"
Gisstate	"0"

Table 6-4. The main attributes of the XML file

The geometry of the object give a general description about the different features and their characteristics added to their location in a local system defined by the metronome company. Example: number="1" object="line 1 circle 3". An object is localized by reference to start localization, and an end one added to the begin foot pulse and the end one. Each localization tag is characterized by the following parameters, Table 6-5.

Localization attribute	Attribute value
Numberingarea	="5900aa"
Orientation	="79.7452"
Region	="12"
Lineofvision	="DescendingStation"
Footpulse	>4560
Distance	="106180044.625"
Number	="5320"
Directioncode	="1"
Startnodedepartment	="221937"
Startnodeidentifier	="270"
Startnodedescription	="Nürnberg Hbf Bf"
Startnodetoken	="NN"
Startnodedate	="0001-01-01T00:00:00"
Endnodedepartment	="221937"
Endnodeidentifier	="31D"
Endnodedescription	="Nürnberg Hbf Bf"
Endnodetoken	="NN"
Endnodedate	="0001-01-01T00:00:00"
Distance	="217.384"

Table 6-5. The localization attributes

Finally, the meta-tag presents additional tags containing additional information about the attached images for each profile. It is presented by

```
<meta name="objectpicture">
        <value xsi:type="xsd:string"> tmp265.tmp</value>
</meta>
```

Where tmp265.tmp present the correspondent profiles image.

As a first impression, the developed prototype is manually driven, where a specialist has to interact within the system interface in order to achieve the required tasks. From another side, the created XML information base seems to be very complex where lots of useless data exists there. Likewise, such a created data structure presents a main base for the information system management. Moreover, it just allows to manage data in a portable manner without giving a new meaning to the existing schema, except for the taxonomical one presenting the heritage concept. In that field, presenting a fully automatized new approach where users' skills can be automatically included there seems to be very relevant.

6.4.2 Knowledge guidance for 3D point clouds geometry detection and qualification

6.4.2.1 *System architecture overview*

As seen in Figure 6-16, the whole updated 3DSQ process takes the 3D point clouds as input, and an ontology structure presenting a knowledge base to manipulate objects, geometries, spatial relation and Object and data properties and produces a qualified architectural scene as an output. The first step aims at the geometric element detection from a specified 3D point clouds file, based on specific semantic rules and 3D processing Built-Ins. Different scenario for lines and plan detection will take place, based on available knowledge. Once the geometries are detected, a qualification process of the eventual spatial relation within is required. It aims to characterize the relation between geometries. In this context, different spatial relationships presented by specific built-ins are taken into consideration. Once the spatial relationships between detected geometries are qualified, a final step aims at the semantic qualification of geometries. To do so, two different technologies are used in this context. The processing steps can be detailed where three main steps aim at detecting and identifying objects.

- From 3D point clouds to geometric elements.
- From geometry to topologic relations.
- From geometric and/or spatial relations to semantic qualified elements.

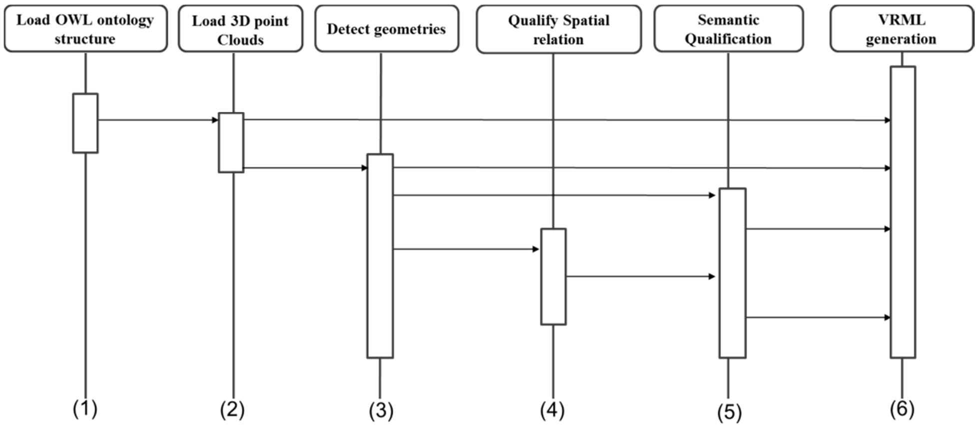


Figure 6-16. The sequence diagram of interactions between the laser scanner, 3D processing, knowledge processing and the knowledge base

6.4.2.2 *Modelled Knowledge base*

In fact, the context of a railway scene is more specific than any normal architectural scene, since it's related to outdoor element detection with very specific rules. The basic target elements are Signals, Electric Pole… This section discusses the different aspects related to the Deutsche Bahn scene ontology structure installed behind the 3DSQ Railway prototype (Ben Hmida, 2010). To ensure the target tasks, knowledge of different domains is acquired from the relevant sources. Sources such as domain experts are the most reliable knowledge foundation. In fact, the Scene Knowledge will be described in the schema of ontology, and includes semantics of the objects, such as properties, restrictions, and relationships between objects and geometries. The more information about an object that is created and used, the more accurate the detection and qualification process is. An example of defining a semantic object is the following: an electric pole in a railroad has a height of 4m to 6m; it is constructed by a vertical structure that connects to a cube on the ground., there are two parallel linear structures at the top and the distance from an electric pole to a signal column is 1000m along the track. The scene is modelled thought axioms of the DLs and presents the behaviour of objects. For instance, an Electrical terminal presents a subclass of the Domain Concept one.

$$\text{ElectricPole} \sqsubseteq \text{DomainConcept} \tag{36}$$

The object knowledge contains all relevant information about objects and elements that could be found within a Deutsch Bahn scene. This could comprise a list such as: {Signals, Electric pole, Electric box, etc.}. They are used to fix either the main scene within its point clouds file and its size, through attributes related to the scene class, or even to characterize detected elements with different semantic and geometric characteristics. The created knowledge base related to the Deutsche Bahn scene was inspired due to our discussion with the domain expert and due to our study based on the official Web site for the German rail way specification, (Bahn, 2012). An overview of the targeted elements, the most useful and discriminant characteristics to detect it and their inter-relationship is shown hereafter.

Class	Sub Class	Subsub Class	Height	Correspondent image
Signals	Basic Signals	Main Signal	Between 4 and 6 m	
		Distant Signal	Between 4 and 6 m	
	Secondary signal	Vorsignalbake	between 1,5 and 2.5 m	
		Breakpoint_table	between 1 and 2 m	
		Chess_board	between 1 and 1,5 m	
Mast (Electric Pole)	BigMast		More than 6m	
	NormalMast		Between 5 and 6	
Schaltanlage (Electric Box)	Schalthause		Less than 1m	
	SchaltSchrank		Less than 0,5m	

Table 6-6. An example of the German Railway scene objects

Table 6-6 shows an example of the possible collection of scene elements in the case of a railway scene. They may be additionally structured in a hierarchical order as could be seen convenient for a scene. Basically, a railway signal is one of the most important elements within the railway scene where we find *DC:main_signals* and *DC:secondary_signal*. The main signals are classified onto *DC:primary_signal* and *DC:distant_signal*. In fact, the primary signal is a railway signal indicating whether the subsequent track section may be driven on. A primary signal is usually announced through a distant signal. The last one indicates which image signal to be expected, that will be associated to the main signal in a distance of 1 km. In fact, a big variety of secondary signals exists like the *DC:Vorsignalbake*, the *DC:Haltepunkt* and others. From the other side, the other discriminant elements within the same scene are the *DC:Masts* presenting an electricity pole for the energy supply. Usually, masts have a distance of 50 m from each other. Finally, the *DC:Schaltanlage* elements present small electric box connected to the ground.

Class	SubClass	Subsub Class	Restriction on Line number	Restriction on Planes number
Signals	Basic Signals	Main Signal	1 or 2 Vertical line	0
		Distant Signal	1 or 2 Vertical line	0
	Secondary signal	Vorsignalbake	1 Vertical line	1 Vertical plane
		Breakpoint_table	2 Vertical lines	1 Vertical Plan
		Chess_board	1 Vertical line	1 Vertical plane
Mast	BigMast		2 or 4 vertical lines	0
	NormalMast		2 or 4 vertical lines	0
Schaltanlage	Schalthause			1 Vertical plane 1 Horizontal plane
	SchaltSchrank			1 vertical plane

Table 6-7. Geometric characteristics overview

Finally, concerning the required 3D spatial knowledge, 3D Metric knowledge presents important information, since the different elements respect very strict metric rules. Such knowledge is used to enhance the qualification process, since information about how objects are dispersed in a 3D scene makes the detection and qualification easier. As the example of the railway scene, the distance between a Distance Signal and a Main Signal should be an average of 1000m. Because of outside factors, such as data noise and the uncertainty of the measurement, the knowledge allows tolerances while executing the correspondent built-ins depending on the quality of data, Figure 6-17.

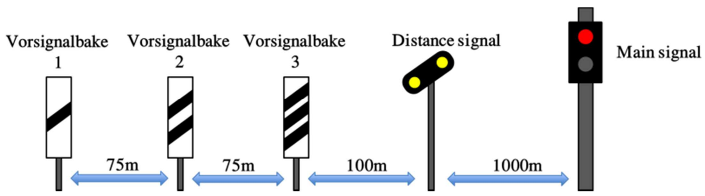


Figure 6-17. Metric rules for the railway scene

To make it more concrete, the human observation already defined in the Table 6-7 will be modelled through different DLs expression, based on the defined ontology schema,. For instance, the taxonomical behaviour of *BasicSignal* is that it is a subclass of *Signal* and ultimately a type of *DomainConcept*.

$$\text{DomainConcept} \sqsubseteq \text{Signale} \sqsubseteq \text{Hauptsignale} \tag{37}$$

Likewise, *Signale* is related to class *Line_3D* (which is a type of class *Geometry*) through the relationship *hasLine3D as a* subproperty of the *hasGeometry* one.

$$\mathit{BasicSignal} \sqsubseteq \exists \text{hasLine3D.Line\_3D} \tag{38}$$

Last, the restriction axioms define the semantics of *BasicSignal*. For example, it should have a height of at least 4 meters and should contain exactly one parallel line and two perpendicular lines and so on. It should be noted that this is a simplified example. In reality, *BasicSignal* can have a variety of different characteristics.

$$\mathit{BasicSignal} \sqsubseteq \exists \text{hasHeight:} \geq 4 \tag{39}$$

$$\mathit{BasicSignal} \sqsubseteq \exists{=}1.\text{hasParallel} \tag{40}$$

Likewise, it is related to the class Line_3D (SubClass of class Geometry) through the relationship hasLine3D (subproperty of hasGeometry).

$$\text{Electric Pole} \sqsubseteq \exists \text{hasLine3D.Line\_3D} \quad (41)$$

For instance, the following DLs constructor defines the semantic of the "ElectricPole". It means should be a vertical bounding box with a height of more than 5 m, and contains at least 2 parallel lines.

$$\text{ElectricPole} \sqsubseteq (\exists\ \text{hasHeight}.\{> 5\} \sqcap \geq 2.\mathit{hasParallel}.\text{Line} \sqcap\ \mathit{VerticalBoundingBox} \sqcap \exists\ \mathit{hasDistanceFrom}.\mathit{DC}{:}\ \text{ElectricPole}\{> 50) \quad (42)$$

6.4.2.3 *Iterative process for object detection and qualification in the railway scene*

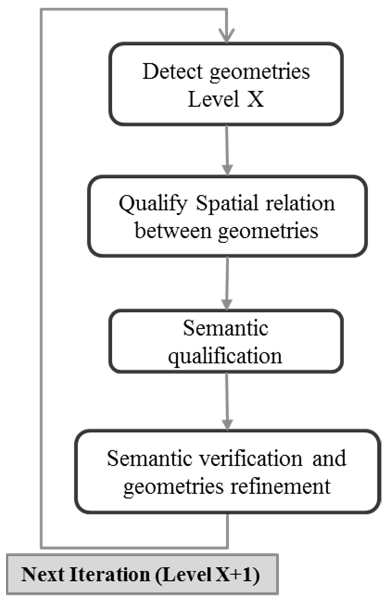


Figure 6-18. The iterative process in the generic case

As a concrete solution, at the core of the first iteration, the railway scene is almost characterized by vertical linear structure. Once detected, and based on SWRL rules enriched with the created 3D spatial Built-Ins, it can be initially qualified as a Signal and an electrical pole for example, Figure 6-19. During advanced iterations (Second iteration in this case), more precise geometry is detected and populated in the knowledge base through the enhanced knowledge once the previous iteration is achieved. Assuming that the main difference between the two above mentioned semantic objects is the existence or not of perpendicular/parallel lines, such a new knowledge will be approved through more sophisticated SWRL rules. In case of the existence of perpendicular lines, the detected object will be finally qualified as an electrical pole.

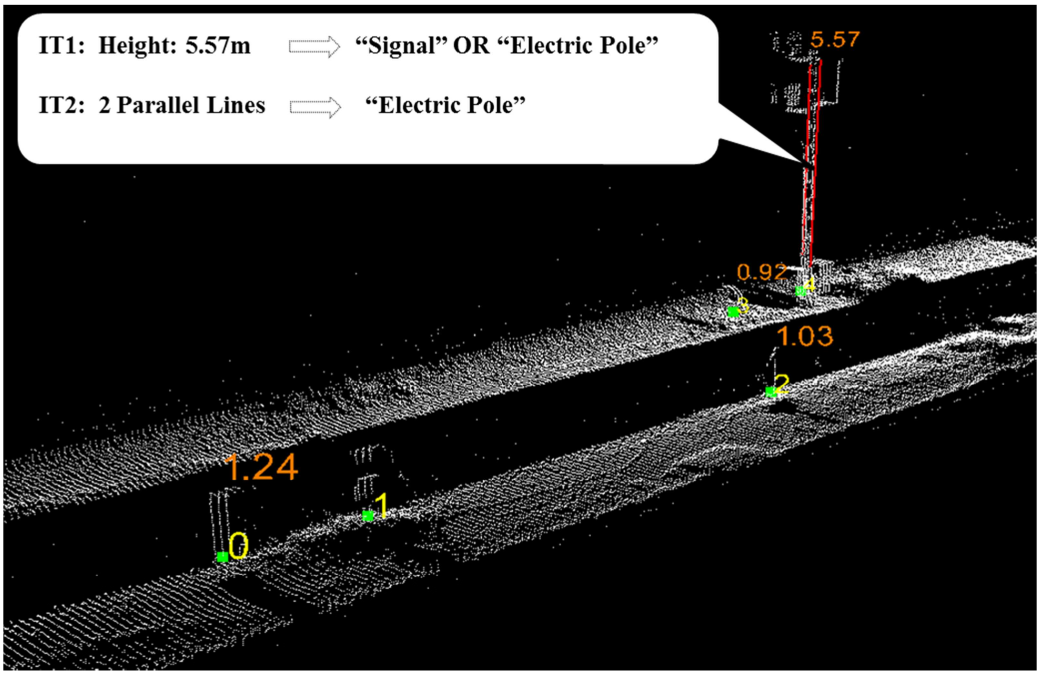


Figure 6-19. Iterative semantic qualification example

6.4.3 The Results

For the demonstration of the extended 3DSQ platform, a scanned point clouds section related to the Deutsch Bahn scene in the city of Nürnberg was extracted. While the last one measured 87 km, our tests were made on two different data bases with a length of 500 m, extracted from the whole scanned point clouds data. The first scene contains 37 elements and the second one contains just 13 elements. At a first impression, it is totally reasonable that the number of elements varies from one scene to another, because we are near the railway station, where the scene is rich and vice versa., Different SWRL rules are processed within the extended 3DSQ platform, where further qualification may be relayed on aspects expressing facts to orientation or size of elements, which may be sufficient to finalize a decision upon the semantic of an object or, in more sophisticated cases, the extended 3DSQ platform allows the combination of semantic information and spatial ones that can deduce more robust results minimizing the false acceptation rate, where it´s clear from Table 6-8, how our knowledge base could recognize which geometry represents a real element from those which are noise.

	Scene Size	Detected Geometry	Qualified Geometry	Truth data
Scene1	500m	105	35	34
Scene2	500m	63	15	13

Table 6-8. Detected Element within the scene and the qualified ones

To evaluate such an extension, several evaluation algorithms and metrics exist nowadays. Among them, qualification recall and precision metric will be adopted. The Qualification precision (QP) presents the fraction of retrieved instances that are relevant, while the Qualification recall (QR) presents fractions of relevant instances that we are able to retrieve. From Table 6-9 and Table 6-10, in most cases the extended 3DSQ platform is able to allocate the right class identity to the detected geometry, based on knowledge related to its component and spatial relation. Based on precision measurement, It is clear that the presented 3DSQ extension is able to discriminate real elements in a precise way, with high accuracy and exactness, and then minimize the false positive qualification. Likewise, the returned value of the Recall measurement reflects the ability of the system to qualify the maximum set of elements that exist in the true data. Otherwise, the rejection process of false geometries that may present noise is done in a very secure manner and with high sureness. As a compromise between the recall and the precision of the extended 3DSQ platform, the system highly responds to the users' needs.

Moreover, some restrictions are observed mainly for geometries which are qualified as Schaltanlage in a false manner. Before explaining the reason behind this false qualification, let's recall that the Schaltanlage presents very small electronic boxes installed on the ground. In the case of scene 1 which is near the railway station, the level of the ground is higher compared to the other scenes. For this reason, lots of geometries are detected where a high number of them present low noise on the ground. The reason for the false qualification is the lack of semantic characteristics related to such elements, since until now; there is no real internal or external spatial relation, nor internal geometric characteristics that discriminate such an element.

		Masts	Signal	Schaltanlage
Truth Data		13	18	3
Qualified Geometry	**True**	11	16	2
	False	1	3	2
Recall (%)		84,61	88,88	66,66
Precision (%)		91,66	84,21	50

Table 6-9. Detected and qualified geometry within the scene 1

		Masts	Signal
Truth Data		6	7
Qualified Geometry	**True**	5	6
	False	2	2
Recall (%)		83,33	85,71
Precision (%)		71,42	75

Table 6-10. Detected and qualified geometry within the scene 2

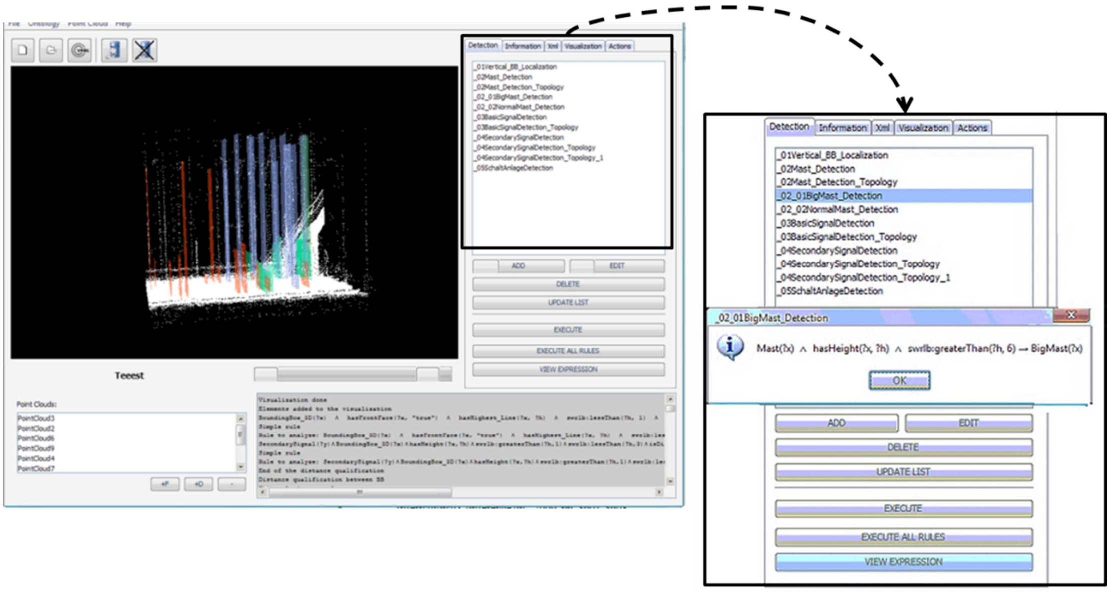


Table 6-11. Detected and qualified geometry visualized within the extended 3DSQ platform

We have presented a comprehensive full automatized system for 3D object detection and qualification, mainly inspired from 3DSQ tool and its ability to process spatial knowledge in a qualitative manner. The adopted prototype is based on semantics of different associated domains, which assist in detection and qualification of objects. Unlike other approaches, knowledge provides an overall base, and is integrated into all the processing steps. This provides the flexibility to infer the strategy from existing knowledge. In the case of 3D point clouds as input data, and different from the CAD/IFC case, the quality of results clearly depends on the robustness of the implemented algorithms to detect correct and precise geometry. Contrary to the standard approach, our knowledge driven methods rely more on knowledge engineering capability, and, as a result, avoid any human intervention. It includes four main components: the Knowledge base, the 3D processing algorithm, the 3D spatial relation, and finally the qualification process. The scene index is built by detecting different geometries and qualifying them via the extended SWRL rules. The developed Java platform provides an efficient demonstration tool taking a set of 3D point clouds within an empty OWL knowledge base as input, and producing a populated ontology with the detected and qualified object.

6.5 Conclusion

In this chapter, we tried to contribute to the on-going enhancement of the Semantic Web technologies, by focusing on the possibility of integrating 3D processing and spatial relation components within its framework. We make an attempt to cross the boundary of using semantics within the 3D processing research, provide interoperability and take it a step forward in using the underlying knowledge technology to provide 3D processing and spatial analysis through knowledge. The presented 3DSQ contribution raises the issue of object detection and recognition in 3D point clouds within the laser scanner, by using available knowledge on the target domain, processing algorithms and the 3D spatial topologic relations, (Ben Hmida et al., 2012). The 3DSQ framework is primarily designed to facilitate the object detection and recognition in 3D point clouds. It is based on Semantic Web technologies and has ontology in its core. The top level ontology provides the base for functionalities of the application. This prior knowledge modelled within an ontology structure. SWRL rules are used to control the 3D processing execution, the 3D Spatial qualification and finally to qualify the detected elements in order to enrich the ontology and to drive the detection of new objects. The designed prototype takes 3D point clouds of a facility, and produces fully annotated scenes within a VRML model file. The suggested solution for this challenging problem has proven its efficiency through real tests within architectural scenes and the Fraport example (Chap 5). The creation of processing and Spatial Built-Ins has presented a robust solution to resolve our problematic and to prove the ability of the semantic web language to intervene in any domain and create the difference. More precisely, this chapter has discussed and proved how we can take a step forward from the qualification of Spatial relation, so as to maximize the use of such a 3D Spatial knowledge mainly in the architectural domain. Via the different subsections, we primary demonstrate the possibility to reach the high level advancement suggested by the literature, but also, to make it optimal and accurate from one side, and mainly to extent new semantic knowledge related to 3D Spatial relation and to the geometric elements identity qualification via the resulted relations from another side. The presented examples within this section aim to raise the issues for the Spatial Relation integration within a relatively new semantic technology. Initially addressed to deal with heterogeneity in the web technology, the knowledge engineering technology is more and more deduced for collaborative approach between humans and machines, where human intelligence and reasoning are injected inside actual solutions.

References

Alexander, C., Smith-Voysey, S., Jarvis, C., Tansey, K., 2009. Integrating building footprints and LiDAR elevation data to classify roof structures and visualise buildings. Computers, Environment and Urban Systems, 33, pp.285--292.

Andreas, N., 2005. Automatic Model Refinement for 3D Reconstruction with Mobile Robots. Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003., pp.394-401.

Antoniou, G. & Harmelen, F., 2009. Web ontology language: Owl. Handbook on ontologies, pp.91--110.

Automation, M., 2012. Metronom Automation. [Online] Available at: http://www.metronom-automation.de/en/ [Accessed 17 September 2012].

Bahn, D., 2012. Deutschen Bahn AG. [Online] Available at: http://www.deutschebahn.com/de/start.html [Accessed 17 September 2012].

Bazjanac, V., 2008. IFC BIM-based methodology for semi-automated building energy performance simulation.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2010. Semantic-based Technique for the Automation the 3D Reconstruction Process. In SEMAPRO 2010, The Fourth International Conference on Advances in Semantic Processing. Florence, Italy, 2010.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2012. Knowledge Base Approach for 3D Objects Detection in Point Clouds Using 3D Processing and Specialists Knowledge. International Journal On Advances in Intelligent Systems, 5, pp.1--14.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs, F., 2011. From 3D point clouds to semantic object. In KEOD, theInternational Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. Paris, 2011.

Ben Hmida, H., Cruz, C., Nicolle, C., Boochs., 2012. From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach. In I. 978-953-51-0535-0. Semantics - Advances in Theories and Mathematical Models. InTech. pp.213-46.

Boochs, F., Marbs, A., Ben Hmida, H., Truong, H., Karmachaiya, A., Cruz, C., Habed, A., Nicolle, C., Voisin, Y., 2011. Integration of knowledge to support automatic object reconstruction from images and 3D data. In International Multi-Conference on Systems, Signals & Devices. Sousse Tunisia, 2011.

Bose, P., Devroye, L., Loffler, M., Snoeyink, J., Verma, V., 2011. Almost all Delaunay triangulations have stretch factor greater than pi/2. Computational Geometry, 44, pp.121--127.

Brooks, S., Gelman, A., JONES, G. & Meng, X.L., 2011. Handbook of Markov Chain Monte Carlo: methods & applications. Recherche, 67, p.02.

Cantzler, H., 2003. Improving architectural 3D reconstruction by constrained modelling. College of Science and Engineering. School of Informatics.

Cantzler, H., Fisher, R., Devy, M., 2002. Quality enhancement of reconstructed 3D models using coplanarity and constraints. Pattern Recognition, pp.34-41.

Chang, K., 2010. Introduction to geographic information systems. McGraw-Hill New York, NY.

Chum, O. & Matas, J., 2008. Optimal randomized RANSAC. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30, pp.1472--1482.

Corporation, L., 2006. What is Constructive Solid Geometry? [Online] Available at: http://www.leadwerks.com/files/csg.pdf [Accessed 23 September 2012].

Cruz, C., 2008. Ontology Enrichment and Automatic Population From XML Data. Proc. of ODBIS.

Durand, N., Derivaux, S., Forestier, G., Wemmert, C., Gancarski, P., Boussaid, O., Puissant, A., 2007. Ontology-based object recognition for remote sensing image interpretation. In 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007.

Eich, M. & Kirchner, F., 2010. Reasoning about geometry: An approach using spatial-descriptive ontologies. In Workshop AILog, 19th European Conference on Artificial Intelligence.

Elberink, S.O. & Vosselman, G., 2009. Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations. Sensors, 9, pp.6101--6118.

Fraport, 2012. Fraport AG | de. [Online] Available at: http://www.fraport.de/content/fraport-ag/de.html [Accessed 17 September 2012].

FTI, 2012. FTI Engineering Network GmbH. [Online] Available at: http://www.ftigroup.net/en/ [Accessed 17 September 2012].

Horn, M., 2007. Smooth Journey. Fraunhofer magzine: Research technology and innovation, p.48.

Hunter, D., Rafter, J., Fawcett, J., Van Der Vlist, E., Ayers, D., Duckett, J., Watt, A., McKinnon, L., 2011. beginning XML. Wrox.

Lee, K., Anguelov, D., Sumengen, B., Gokturk, S.B., 2008. Markov random field models for hair and face segmentation. In 8th IEEE International Conference on Automatic Face and Gesture Recognition.

Lee, K., Johnson, R.K., Yin, Y., Wahle, A., Olszewski, M.E., Scholz, T.D., Sonka, M., 2010. Three-dimensional thrombus segmentation in abdominal aortic aneurysms using graph search based on a triangular mesh. Computers in biology and medicine, 40, pp.271--278.

Nuchter, A. & Hertzberg, J., 2008. Towards semantic maps for mobile robots. Robotics and Autonomous Systems, 56, pp.915--926.

Pu, S., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 64, pp.575--584.

Pu, S. & Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. Intl Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36, pp.12--14.

Rezgui, Y., Hopfe, C.J., Vorakulpipat, C., 2010. Generations of knowledge management in the architecture, engineering and construction industry: An evolutionary perspective. Advanced Engineering Informatics, 24, pp.219--228.

Ripperda, N. & Brenner, C., 2007. Data driven rule proposal for grammar based facade reconstruction. Photogrammetric Image Analysis, 36, pp.1--6.

Rusu, R.B., Marton, Z.C., Blodow, N., Holzbach, A., Beetz, M., 2009. Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. IEEE/RSJ International Conference on Intelligent Robots and Systems. pp.3601--3608.

Rutzinger, M., Oude Elberink, S., Pu, S., Vosselman, G., 2009. Automatic extraction of vertical walls from mobile and airborne laser scanning data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 38.

Scholze, S., Moons, T., Van Gool, L., 2002. A probabilistic approach to building roof reconstruction using semantic labelling. In Proceedings of the DAGM Conference. Zürich, 2002.

Spinello, L., Triebel, R.. Siegwart, R., 2010. Multiclass multimodal detection and tracking in urban environments. The International Journal of Robotics Research, 29, pp.1498--1515.

Stelldinger, P., 2008. Topologically correct surface reconstruction using alpha shapes and relations to ball-pivoting. In 19th International Conference on Pattern Recognition, ICPR 2008.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from Lidar data.

Teboul, O., Simon, L., Koutsourakis, P., Paragios, N., 2010. Segmentation of building facades using procedural shape priors. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Vosselman, G. & Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34, pp.37--44.

Xu, X.G., Taranenko, V., Zhang, J. & Shi, C., 2007. A boundary-representation method for designing whole-body radiation dosimetry models: pregnant females at the ends of three gestational periods-RPI-P3,-P6 and-P9. Physics in medicine and biology, 52, p.7023.

Chapter 7

Conclusion

7.1 Contribution

Nowadays, 3D spatial reasoning with its three derivations (3D Topological reasoning, 3D metrological reasoning and 3D directional reasoning) is an important part of Artificial Intelligence, where existing approaches in this field are numerically based. Geometric properties and spatial relations between building elements play a major role in the design processes of the Architecture, Engineering and Construction domain. However, so far, spatial relations are not supported by existing building information models. To fill this technological gap, a bridge between a qualitative spatial relation and the quantitative one has been developed.

In fact, the current research paper make an attempt to emphasize the possibility of combining 3D spatial technology and the above mentioned 3D spatial reasoning by the integration of such a technology in the semantic web framework. Such integration will comply with the way human reason about qualitative spatial relationships. This paper discusses the 3D spatial operators and their integration within a qualitative manner for building information models. The semantics of the spatial predicates are formally implemented and defined in an OWL knowledge base, and linked to a quantitative layer through the real execution of the correspondent spatial operation. The Selective Nef Polyhedron based implementation technique of spatial operators which was highlighted in this paper is the most efficient one compared to the CSG and Octree implementation, supporting a big variation of non-uniform geometries with high discrimination of the Interior, Boundary and exterior of each of them. Likewise, such a structure overcomes the exponential behaviour of Octree based algorithms. This paper moves outside of the range of data interoperability while presenting the concepts, and makes an effort to utilize others areas of semantic web technology. The basic capacity of knowledge processing provides the capability to the semantic web to process the semantics of the information, through close collaboration with the machine. In fact, it makes not only the perceptive of 3D spatial data easier for interoperability among different data sources, but mainly provides helpful knowledge which is able to enrich the knowledge base. Such a process primarily helps to understand spatial data and relations in a better way. This helps the users better understand the data. The underlying knowledge technology makes it stand out among its contemporaries. From our point of view, it is extremely important to have standard terms for every created spatial relation and built-in to process the 3D spatial knowledge. To do so, we tried to rely on the tools standardized by W3C and OCG. Finally, the created OWL knowledge base also enables to store the full set of information available in BIMs or IFC, including attributes and relationships, which makes it possible to use such information added to the Spatial one for further process, especially as in the geometry qualification in our case.

To highlight the utilisability of the 3D Semantic Spatial Qualification Approach (3DSQ), we decided to extend the research by taking a step forward from the qualification of the spatial relation semantically, to the extension of the semantic rules and query language. Such an improvement supports the inference on 3D spatial knowledge and enables final querying of the spatial knowledge base. The main advantage of using the Semantic approach compared to standard SQL ones is its simplicity. Added to its ability to process spatial data in our case, the semantic approach ensures a common understanding of the spatial domain between Humans and machines, via ensuring the Semantic inference and queries using spatial knowledge. The 3DSQ translation engine enables the computation of spatial SWRL rules which can also be queries. It 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 SNC level. The results are populated in the knowledge base, thus making it spatially rich. After that, the spatial statements are translated to standard ones for the execution through their respective engines. With the inference engine, the enrichment and population of the ontology through the results of the inference process is eventually stored in the knowledge base. From another angle, this paper has presented a synchronized knowledge driven approach to qualify 3D spatial relation initially. This has been extended later on to qualify the 3D geometry based on different characteristics. It is based on the semantics of different associated domains, which assist in the qualification process. Unlike other approaches, knowledge provides an overall base and is integrated into all the steps of the processing, including the detection one, in the case of 3D point clouds data. As a conclusion, the current paper raises the issue of 3D geometry processing, by using available knowledge on the target domain and 3D spatial relations.

The benefits of the emerging Semantic Web technology through its knowledge tools are quite visible compared to convention technologies which rely heavily on database systems. More precisely, the benefits that were achieved during the design and development of the 3DSQ platform and it´s extension are quite high. The flexible nature of an ontology based system enables the integrating of new components at any time of development and even implementation. The created prototype is easy to use, and the user does not need specific knowledge to understand the results.

The 3DSQ framework is primarily designed to facilitate object manipulation, and mainly recognition in 3D data for BIM purposes. It is based on Semantic Web technologies, and has ontology in its core. Top level ontology provides the basis for the functionalities of the application. This prior knowledge is modelled within an ontology structure. SWRL rules are used to control the processing chain, spatial qualification and finally 3D object qualification. The designed prototype takes 3D facility data as input and produces fully qualified scenes within a VRML model file. The suggested solution for this challenging problem has proven its efficiency through real tests within the Deutsche Bahn, and mainly the Frankfurt airport scene. The creation of Spatial Built-Ins has proven to be a robust solution in resolving our problematic, and proving the ability of the semantic web language to intervene in any domain and create the difference. In more concrete discussion and compared to the works of Borrmann et al, new efficient data structure able to optimize the spatial quantification process is introduced. In the same axe, the quantification process of spatial relation is semantically integrated within the Semantic Web platform through a knowledge based approach opening the door for a new area of research and application. This can be seen as a major contribution in our thesis. From an applied area point of view, the 3D qualitative data integration in the semantic web framework has been proved through different uses cases, mainly the CAD/IFC 3D geometry qualification and the 3D object detection and qualification in 3D point clouds data. In this field, and compared to Prof Nüchter et al and Eich et al works, we share from one side the common purpose materialized through geometry detection and qualification in 3D data. As a main steps, the last author relay on an ontology describing the scene elements and mainly the directional relation among them (Orthogonal, parallel, perpendicular). Once created, a probabilistic reasoning approach based on spatial feature descriptions is taking place aiming at finding the best match between the model entity and the extracted spatial ones. From our side, a more knowledge based approach is presented where the maximum of capacities of the 3D spatial domain with its different derivations has been expressed. In fact, our contributions aim at presenting a complete semantic based approach where the OWL ontology presents the core of the proposition. It´s based on the DLs rules and inference system from one side and the SWRL and SWQRL rules with extended Built-Ins from another side where the semantic platform guide and control the different required processing.

From the other side of the equation, the actual developed 3DSQ still suffer from some drawbacks and limitations. In fact, the actual version of the prototype doesn´t supports the processing of 2D geometries where it has first to be converted to 3D ones by adding a small noise inside it. Second, the developed 3D spatial characteristics extraction built-Ins (Height, orientation, length…) are used to deal with linear and uniform geometries, where further solutions have to presents answers for non-uniform data. Some limitations appear visible also within the semantic rules point of view, and where, while qualifying geometries, ambiguous qualification takes place in some cases. In such a scenario, human intervention can be useful to guide the system. Finally, more sophisticated output and visualization engines can take place, where the flexible generated and enriched knowledge base can be re-transformed to standard IFC format which increases its utilisability degree.

7.2 Future work

Throughout this paper, we tried to contribute to the on-going enhancement of the Semantic Web technologies, by focusing on the possibility of integrating 3D spatial components within its framework. This makes an attempt to cross the boundary of using semantics within 3D research to provide interoperability, and takes it a step forward in using the underlying knowledge technology to provide spatial analysis through knowledge. The 3DSQ approach is of a general nature, and can be adapted for several fields of application by its flexibility to include new knowledge, new Built-Ins interacting with different processing areas and new rules managing such Built-Ins.

Future work will include the integration of new knowledge that can intervene within the qualification and the update of the general platform architecture, by ensuring more interaction between the scene knowledge and the 3D geometries. Added to that, it will include a more robust qualification process of objects, based on each object characteristics, to make the process more flexible and intelligent. Likewise, further research will concentrate on the description logic (DL) formalisation of the difference modelled 3D spatial knowledge, and on the creation of a formal semantic presentation of the different adopted primitives’ solid geometry. Likewise basic rules will be applied, that can be defined and depicted in the next SWRL rule. This can also be done by a composition of relations, $\mathrm{meet} \circ \mathrm{contains} \sqsubseteq \mathrm{Disjoint}$. Likewise, future work will also include the expansion of the ontology, further implementation and testing of the rules, as well as the improvement of the existing JAVA prototype application, and the improvement and addition of 3D spatial qualification algorithms.