

From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach

Christophe Cruz, Helmi Ben Hmida, Frank Boochs, Christophe Nicolle

▶ **To cite this version:**

Christophe Cruz, Helmi Ben Hmida, Frank Boochs, Christophe Nicolle. From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach. Semantics - Advances in Theories and Mathematical Models, 2012, pp.Muhammad Tanvir Afzal, ISBN 978-953-51-0535-0. 10.5772/37633 . hal-00778533

HAL Id: hal-00778533

https://u-bourgogne.hal.science/hal-00778533

Submitted on 20 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

9

From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach

Helmi Ben Hmida, Christophe Cruz, Frank Boochs and Christophe Nicolle
[1]Laboratoire Le2i, UFR Sciences et Techniques
Université de Bourgogne, Dijon,
[2]Fachhochschule Mainz, Institut i3mainz,
amFachbereichGeoinformatik und Vermessung, Mainz,
[1]France
[2]Germany

1. Introduction

Over the last few years, formal ontologies has been suggested as a solution for several engineer problems, since it can efficiently replace standard data bases and relational one with more flexibility and reliability. In fact, well designed ontologies own lots of positive aspects, like those related to defining a 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 represent formally 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, (Boochs, et al., 2011). As most recent works, the tendency related to the use of semantic has been explored, (Ben Hmida, et al., 2010) (Hajian, et al., 2009) (Whiting, 2006) where the automatic data extraction from 3D point clouds presents one of the new challenges, especially for map updating, passenger safety and security improvements. However such domain is characterized by a specific vocabulary containing different type of object. In fact, the assumption that knowledge will help the improvement of the automation, the accuracy and the result quality is shared by specialists of the point cloud processing.

As a matter of fact, surveying with 3D scanners is spreading all domains. Terrestrial laser scanners have been established as a workhorse for topographic and building survey from the archaeology (Balzani, et al., 2004) to the architecture (Vale, et al., 2009). Actually, with every new scanner model on the market, the instruments become faster, more accurate and can scan objects at longer distances. Such technology presents a powerful tool for many applications and has partially replaced traditional surveying methods since it can speed up field work significantly. Actually, this powerful method allows the creation of 3D point clouds from objects or landscapes. However, the huge amount of data generated during the process proved to be costly in post-processing. The field time is very height since in most cases; processing techniques are still mainly affected by manual interaction of the user. Typical operations consist to clean point clouds, to delete unnecessary areas, to navigate in an often huge and complicated 3D structure, to select set of points, to extract and model

geometries and objects. At the same time, it would be much more effective, to process the data automatically, which has already been recorded in a very fast and effective way.

From another side, the technical survey of facility aims to build a digital model based on geometric analysis. Such a process becomes more and more tedious. Especially with the new terrestrial laser scanners where a huge amount of 3D point clouds are generated. Within such scenario, new challenges have seen the light where the basic one is to make the reconstruction process automatic and more accurate. Thus, early works on 3D point clouds have investigated the reconstruction and the recognition of geometrical shapes (Pu, et al., 2007) to resolve this challenge. In fact, such a problematic was investigated as a topic of the computer graphic and the signal processing research where most works focused on segmentation or visualization aspects. As most recent works, the new tendency related to the use of semantic has been explored (Ben Hmida, et al., 2010). As a main operation, the technical survey relies fundamentally on the object reconstruction process where considerable effort has already been invested to reduce the impact of time consuming, manual activities and to substitute them by numerical algorithms. Unfortunately, most of such algorithmic conceptions are data-driven and concentrate on specific features of the objects being accessible to numerical models. By these models, which normally describe the behavior of geometrical (flatness, roughness…) or physical features (color, texture…), the data is classified and analyzed. Such strategies are static and not to allow a dynamic adjustment to the object or initial processing results. In further scenarios, an algorithm will be applied to the data producing better or minor results depending on several parameters like image or point cloud quality, the completeness of object representation, the viewpoints position, the complexity of object features, the use of control parameters and so on. Consequently, there is no feedback to the algorithmic part in order to choose a different algorithm or reuse the same algorithm with changed parameters. This interaction is mainly up to the user who has to decide by himself, which algorithms to apply for which kind of objects and data sets. Often good results can only be achieved by iterative processing controlled by a human interaction.

These problems can be solved when further information is integrated into the algorithmic process chain for object detection and recognition allowing supporting the process of validation. Such information might be derived from the context of the object itself and its behavior with respect to the data and/or other objects or from a systematic characterization of the parameterization and the effectiveness of the algorithms to be used. As programming languages used in the context of numerical treatments are not dedicated to process knowledge, their condition of use is not flexible and makes the integration of semantic aspects difficult.

As a matter of fact, the goal of our proposition is to develop efficient and intelligent methods for an automated processing of terrestrial laser scanner data, Fig 1. The principle our solution is a knowledge-based detection of objects in point clouds for AEC (Architecture, Engineering and Construction) engineering applications in correspondence to a project of the same name "WiDOP". In contrast to existing approaches, the project consists in using prior knowledge about the context and the objects. 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 recognition of objects in point clouds. In such scenario, knowledge about such objects

have to include detailed information about the objects' geometry, structure, 3D algorithms, etc.

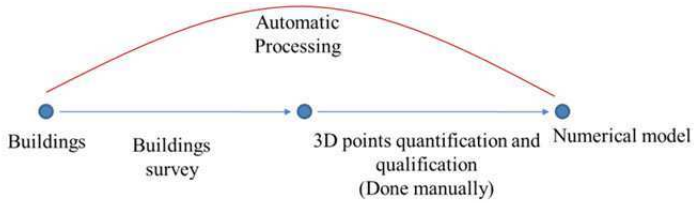


Fig. 1. Automatic processing compared to the manual one.

The present chapter aims at building a bridge between the semantic modelling and the numerical processing to define strategies based on domain knowledge and 3D processing knowledge. The knowledge will be structured in ontologies structure containing a variety of elements like already existing information about objects of that scene such as data sources (digital maps, geographical information systems, etc.), information about the objects' characteristics, the hierarchy of the sub-elements, the geometrical topology, the characteristics of processing algorithms, etc. In addition, all relevant information about the objects, geometries, inter and intra-relation and the 3D processing algorithms have been modeled inside the knowledge base, including characteristics such as positions, geometrics information, images textures, behavior and parameter of suitable algorithms, for example.

By this contribution, an approach on achieving the object detection and recognition within those 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 heterogynous environment. It primarily focuses on 3D point clouds and its management through the available processing technologies for object detection and recognition incorporated through the knowledge. As the Web technologies get matured through its approach in the Semantic Web, the implementation of knowledge in this domain seems to be more appropriate.

This research puts forward the views and result of the research activities in the backdrop of the Semantic Web technologies and the knowledge management aspect within it. The suggested system is materialized via WiDOP project (Ben Hmida, et al., 2011). Furthermore, the created WiDOP platform is able to generate an indexed scene from unorganized 3D point clouds visualized within the virtual reality modelling language (W3C, 1995).

The following chapter is structured into section 2 which gives an overview of actual existing strategies for reconstruction processes, section 3 highlight the adopted languages and technologies for knowledge and semantic modeling, section 4 explains the suggested architecture for the WiDOP solution, section 5 presents an overview of the related knowledge model, section 6 emphasizes the intelligent process. Section 7 shows different strategies and level of knowledge for the processing, section 8 present the developed platform and gives first results for a real example, and finally section 9 concludes and shows next planned steps.

2. Background concept and methodology

The technical survey of facilities, as a long and costly process, aims at building a digital model based on geometric analysis since the modeling of a facility as a set of vectors is not sufficient in most cases. To resolve this problem, a new standard was developed over ten years by the International Alliance for Interoperability (IAI). It is named the IFC format (IFC - Industry Foundation Classes) (Vanland, et al., 2008). The specification is a neutral data format to describe exchange and share information typically used within the building and facility management industry. This norm considers the building elements as independent objects where each object is characterized by a 3D representation and defined by a semantic normalized label. Consequently, the architects and the 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 Signal is not just a simple collection of lines and geometric primitives recognized as a signal; it is an "intelligent " object signal which has attributes linked to a geometrical definition and function. IFC files are made of objects and connections between these objects. Object attributes describe the "business semantic" of the object. Connections between objects are represented by "relation elements". This format and its semantics are the keystone of our solution.

The problematic of 3D object detection and scene reconstruction including semantic knowledge was recently treated within a different domain, basically the photogrammetry one (Pu, et al., 2007), the construction one, the robotics (Rusu, et al., 2009) and recently the knowledge engineering one (Ben Hmida, et al., 2010). Modeling a survey, in which low-level point cloud or surface representation is transformed into a semantically rich model is done in three tasks where the first 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 data processing, in which the sets of point clouds from the collected scanners are processed. Finally, modeling the survey in which the low-level point cloud is transformed into a semantically rich model. This is done via modeling geometric knowledge, qualifying topological relations and finally assigning an object category to each geometry (Boochs, et al., 2011). Concerning the geometry modeling, we remind here that the goal is to create simplified representations of facility components by fitting geometric primitives to the point cloud data. The modeled components are labeled with an object category. Establishing relationships between components is important in a facility model and must also be established. In fact, relationships between objects in a facility model are useful in many scenarios. In addition, spatial relationships between objects provide contextual information to assist in object recognition (Cantzler, 2003). Within the literature, three main strategies are described to rich such a model where the first one is based on human interaction with provided software's for point clouds classifications and annotations (Leica, 2011). While the second strategy relies more on the automatic data processing without any human interaction by using different segmentation techniques for feature extraction (Rusu, et al., 2009). Finally, new techniques presenting an improvement compared with the cited ones by integrating semantic networks to guide the reconstruction process have seen the light.

2.1 Manual survey model creation

In current practice, the creation of a facility model is largely a manual process performed by service providers who are contracted to scan and model a facility. In reality, a project may

require several months to be achieved, depending on the complexity of the facility and the modeling requirements. Reverse engineering tools excel at geometric modeling of surfaces, but with the lack of volumetric representations, while such design systems cannot handle the massive data sets from laser scanners. As a result, modelers often shuttle intermediate results back and forth between different software packages during the modeling process, giving rise to the possibility of information loss due to limitations of data exchange standards or errors in the implementation of the standards within the software tools (Goldberg, 2005). Prior knowledge about component geometry, such as the diameter of a column, can be used to constrain the modeling process, or the characteristics of known components may be kept in a standard component library. Finally, the class of the detected geometry is determined by the modeler once the object is created. In some cases, relationships between components are established either manually or in a semi-automated manner.

2.2 Semi-Automatic and Automatic methods

The manual process for constructing a survey model is time consuming, labor-intensive, tedious, subjective, and requires skilled workers. Even if modeling of individual geometric primitives can be fairly quick, modeling a facility may require thousands of primitives. The combined modeling time can be several months for an average-sized facility. Since the same types of primitives must be modeled throughout a facility, the steps are highly repetitive and tedious (Hajian, et al., 2009). The above mentioned observations and others illustrate the need semi-automated and automated techniques for facility model creation. Ideally, a system could be developed that would take a point cloud of a facility as input and produce a fully annotated as-built model of the facility as output. The first step within the automatic process is the geometric modeling. It presents the process of constructing simplified representations of the 3D shape of survey components from point cloud data. In general, the shape representation is supported by Constructive Solid Geometry (CSG) (Corporation, 2006) or Boundary representation B-Rep representation (CASCADE, 2000). The representation of geometric shapes has been studied extensively (Campbell, et al., 2001). Once geometric elements are detected and stored via a specific presentation, the final task within a facility modeling process is the object recognition. It presents the process of labeling a set of data points or geometric primitives extracted from the data with a named object or object class. Whereas the modeling task would find a set of points to be a vertical plane, the recognition task would label that plane as being a wall, for instance. Often, the knowledge describing the shapes to be recognized is encoded in a set of descriptors that implicitly capture object shape. Research on recognition of facility's specific components related to a facility is still in its early stages. Methods in this category typically perform an initial shape-based segmentation of the scene, into planar regions, for example, and then use features derived from the segments to recognize objects. This approach is exemplified by Rusu et al. who use heuristics to detect walls, floors, ceilings, and cabinets in a kitchen environment (Rusu, et al., 2009). A similar approach was proposed by Pu and Vosselman to model facility façades (Pu, et al., 2009). To reduce the search space of object recognition algorithms, the use of knowledge related to a specific facility can be a fundamental solution. For instance, Yue et al. overlay a design model of a facility with the as-built point cloud to guide the process of identifying which data points belong to specific objects and to detect differences between the as-built and as-designed conditions (Yue, et al., 2006). In such cases, object recognition problem is simplified to be a matching problem between the scene model entities and the data points. Another similar approach is presented in

(Bosche, et al., 2008). Other promising approaches have only been tested on limited and very simple examples, and it is equally difficult to predict how they would fare when faced with more complex and realistic data sets. For example, the semantic network methods for recognizing components using context work well for simple examples of hallways and barren, rectangular rooms (Cantzler, 2003), but how would they handle spaces with complex geometries and clutter.

2.3 Discussion

The presented methods for survey modeling and object recognition rely on hand-coded knowledge about the domain. Concepts like "Signals are vertical" and "Signals intersect with the ground" are encoded either explicitly, through sets of rules, or implicitly, through the design of the algorithm. Such hard-coded, rule based approaches tend to be brittle and break down when tested in new and slightly different environments. Additionally, we can deduce that authors model the context but not the 3D processing algorithms, the geometry and the topology. Furthermore, it will be difficult in such a case to extend an algorithm with new rule or to modify the rules to work in new environments. To make it more flexible and efficient, and in contrast with the literature, we opt to use a new data structure labeled ontology. In fact, the last one presents a formal representation of knowledge by a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain where the basic strength of formal ontology is their ability to present knowledge within their taxonomy, relations and conditions, but also to reason in a logical way based on Description Logics DL concepts. Based on these observations, we predict that more standard and flexible representations of facility objects and more sophisticated guidance based algorithms for object detection instead of a standard one, by modeling algorithmical, geometrical and topological knowledge within an ontology structure will open the way to significant improvement in facility modeling capability and generality since it will allow as to create a more dynamic algorithm sequence for object detection based on object's geometries and to make more robust the identification process.

3. Knowledge and Semantic web

The growth of the World Wide Web has been tremendous since its evolvement both in terms of the content and the technology. The first Web generation was mainly presentation based. They provided information through the Web pages but did not allow users to interact with them. In short, they contained read only information. Moreover, they were only text pages and do not contain multimedia data. These Web sites have higher dependency on the presentation languages like Hypertext Markup Languages (HTML) (Horrocks, et al., 2004). With the introduction of e**X**tensible**M**arkup**L**anguage (XML), the information within the pages became more structured. Those XML based pages could hold up the contents in more structured method but still lack the proper definition of semantics within the contents, (Berners-Lee, 1998). For this reason, the needs of intelligent systems which could exploit the wide range of information available within the Web are widely felt. Semantic Web is envisaged to address this need.

The term "*Semantic Web*" is coined by Tim Berners-Lee in his work (Lee, et al., 2001) to propose the inclusion of semantic for better enabling machine-people cooperation for

handling the huge information that exists in the Web. The term "*Semantic Web*" has been defined numerous time. Though there is no formal definition of Semantic Web, some of its most used definitions are "*The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is a source to retrieve information from the Web (using the Web spiders from RDF files) and access the data through Semantic Web Agents or Semantic Web Services. Simply Semantic Web is data about data or metadata*" (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). In the next subsection, we discuss the different issues related to the definition of such a technology where we focus mainly on the Description Logic theory (DL) and its impact on the semantic web technology.

3.1 The description logics

Actually, the convergence of formal foundations for extensible, semantically understood structure within description logic and the overall usability targets of the predecessor of DL and the Web languages for broader usability of Web has led to the effort such as Ontology Interface Language (OIL) (Fensel, 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 the broader project called On-To-Knowledge funded by European Union. This is the first time that the concept within ontology is explicitly used within a Web based environment. However, it did not completely leave out the primitives of frame base languages with the formal semantics and reasoning capabilities by including them within the language. The syntax of OIL is based on RDF and XML with their limitations to provide complete semantic foundations at that time. However, it has started a trend of mapping description logic within the Web based language for Semantic Web. It maps description logic through $\mathcal{SHIQ}$. The derivation of $\mathcal{SHIQ}$ with respect to naming convention of the Description Logic is given as:

$\mathcal{S}$: Used for all $\mathcal{ALC}$ with transitive roles R+
$\mathcal{H}$: Role inclusion axioms R1⊑ R2 (is_component_of ⊑ is_part_of)
$\mathcal{I}$: Inverse Role R-(isPart0f = hasPart-)
$\mathcal{Q}$: Qualified number restrictions

3.1.1 The base languages

Complex descriptions can be built up through the above mentioned elementary descriptions of concepts and roles. These descriptions are given different notations over the time. The Attributive Language ($\mathcal{AL}$) has been introduced in 1991 as minimal language that is of practical interest (Schmidt-Schauß, et al., 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 descriptions. $\mathcal{ALC}$ is the important notation format to express Description Logics. Fig 2 illustrates the syntax rules on describing the concept.

Notation	Syntax	Semantics	Read-as
$\top$	$C, D \rightarrow \top$	$\top(x)$	Universal concept
$\bot$	$\bot$	$\bot\ (x)$	Bottom concept
$\sqcap$	$\mathrm{C} \sqcap D$	$\mathrm{C(x)}\ \sqcap \mathrm{D(x)}$	Intersection
$\sqcup$	$C \sqcup D$	$\mathrm{C(x)}\ \sqcup \mathrm{D(x)}$	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

Here C and D are concept description and R is role

Fig. 2. The syntax and semantics based on $\mathcal{ALC}$.

We introduce in this section 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 $C \sqsubseteq D, R \sqsubseteq S\ Or\ C \equiv D, R \equiv S$ where C, D are concepts (and R, S are roles). Axioms of the first kind are called inclusions, while axioms of the second kind are called equalities. An equality whose left-hand side is an atomic concept. It´s used to introduce symbolic names for complex descriptions e,g. $RailWorker \equiv Person \sqcap \exists haswork.RailWork$. It could be clearly seen within Fig 2 that these concept descriptions are built with the concept constructors. The first four constructors are not dependent on the roles whereas the last two utilizes the roles in the constructors. This dependency is called role restrictions. Formally, a role restriction is an unnamed class containing all individuals that satisfy the restriction. DLs expressed through $\mathcal{ALC}$ provide two such restrictions in Quantifier restriction and value restrictions.

The Quantifier restriction

It´s 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. Simplifying, the property restriction *P* relates to a concept of individuals *x* having at least one y which is either an instance of concept description or a value of data range so that *P(x,y)* is an instance of *P*.

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

3.1.2 The description logics formalization

Description logics (DLs) are a family of logics which represents the structured knowledge. The Description Logic languages are knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way (McGuinness, et al., 2003), (Calvanese, et al., 2005). Description logics contain the formal, logic-based semantics, which present the major reason for its choice for Semantic Web languages over its predecessors. The reasoning capabilities within the DLs add a new dimension. Having these capabilities as central theme, inferring implicitly represented knowledge becomes possible. The movement of Description Logic into its applicability can be viewed in terms of its progression in Web environment (Noy, et al., 2001). Web languages such as XML or RDF(S) could benefit from the approach DL takes to formalize the structured knowledge representation (Lassila, 2007). This has laid background behind the emergence of Description Logic languages in Web. Actually, an agreement 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. As could be seen in Fig 3, $\mathcal{ALC}$ has been extended to transitive role and given abbreviation S in the convention. Where S is used in every DL systems and languages as it plays significant role in shaping the behavioural nature of every DL systems.

$\mathcal{AL}$	$:C, D \rightarrow \top \mid \bot \mid A \mid C \sqcap D \mid \neg A \mid \exists R.T \mid \forall R.C$
$\mathcal{C}$	: Concept negation $\neg C$. Thus, $\mathcal{ALC}$=$\mathcal{AL}$+$\mathcal{C}$
$\mathcal{S}$	: Used for $\mathcal{ALC}$ with transitive role R+
$\mathcal{U}$	: Concept disjunction $C \sqcup D$
ε	: Existential quantification, $\exists R.C$
$\mathcal{H}$	: Role inclusions axioms, R1⊑ R2, e.g is_component_of ⊑ is_part_of
$\mathcal{N}$	:Number Restrictions, (≥ nR) and (≤ nR), e.g (≥ has_Child) (has at least 3 child)
$\mathcal{Q}$	:Qualified number restriction, (≥n R.C) and (≤n R.C) , e.g (≤2 has_child.Adult) (has at most 2 adult Children)
$\mathcal{O}$	:Nominals (singleton class), {a}. e.g $\exists has_Child.\{mary\}$.
$\mathcal{I}$	:Inverse role R-, e.g isPartof=hasPart-
$\mathcal{F}$	:functional role, e.g functional(hasAge)
$\mathcal{R}$+	:Transitive role , e.g., transitive (isPartOf)
$\mathcal{R}$	:role inclusion with comparison, R1 o R2 ⊑ S, e.g, isPartOf o isPartOf ⊑ $isPartOf$

Fig. 3. Naming convention of Description Logic.

3.2 The knowledge base

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

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

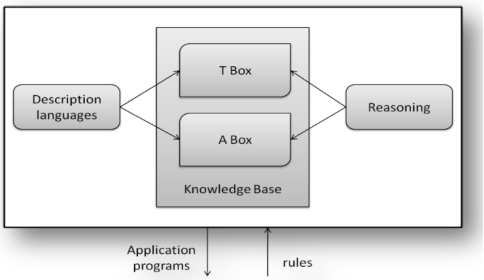


Fig. 4. The Architecture of a knowledge representation system based on DLs.

Baader (Baader, 2006) sketches the architecture of any KR system based on DLs. It could be seen the central theme of such a system is a Knowledge Base (KB). The KB constitutes of two components: the TBox and the ABox. Where **TBox** statements are the *terms* or the *terminologies* that are used within the system domain. In general they are statements describing the domain through the controlled vocabularies. For example in terms of a social domain the TBox statements are the set of concepts as *Rail, train, signal* etc. or the set of roles as *hasGeometry, hasDetectionAlg, hasCharacteristics* etc. ABox in other hand contains assertions to the TBox statements. In object oriented concept, **ABox** statements compliant **TBox** statements through instantiating what is equivalent to classes in TBox and relating the roles (equivalent to methods or properties in OO concept) to those instances.

The DLs are expressed through the *concepts* and *roles* of a particular domain. This complements well with the fact how knowledge is expressed in the general term. Concepts are sets of classes of individual objects. Where classes provide an abstraction mechanism for grouping resources with similar characteristics (Horrocks, et al., 2008). The concepts can be organized into superclass-subclass hierarchy which is also known as taxonomy. It shares the object-oriented concepts in managing the hierarchy of superconcept-subconcept. The sub-concepts are specialized concepts of their super-concepts and the super-concepts are generalized concepts of their sub-concepts. For an example all individuals of a class must be individuals of its superclass. In general all concepts are subsumed by their superclass. In any graphical representation of knowledge, concepts are represented through the nodes. Similarly the roles are binary relationship between concepts and eventually the relationships of the individuals of those concepts. They are represented by links in the graphical representation of knowledge. The description language has a model-theoretic semantics as

the language for building the descriptions is independent to each DL system. Thus, statements in the TBox and in the ABox can be identified as first-order logic or, in some cases, a slight extension of it (Baader, et al., 2008).

3.3 The Web Ontology Language (OWL)

The association of knowledge with Semantic Web has provided a scope for information management through the knowledge management. Since both the technologies use ontology to conceptualize the scenarios, Semantic Web technology could provide a platform for developments of knowledge management systems (Uren, et al., 2006). The ontologies are core to both the technologies in whichever methods they are defined. The Semantic Web defines ontologies, (Gruber, 2008) through XML based languages and with the advancements in these languages. 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). In any case, ontology can be considered as formalization of knowledge representation where the Description Logics (DLs) provide logical formalization to the Ontologies (Baader, et al., 2007).

OWL or the Web Ontology Language is a family of knowledge representation language to create and manage ontologies. It is in general term an extension of RDFS with addition to richer expressiveness that RDFS lacks through its missing features (Antoniou, et al., 2009). The OWL Working Group has approved two versions of OWL: OWL 1 and OWL 2, (Grau, et al., 2008). The Web Ontology Language (OWL) is intended to be used when the information contained in documents needs to be processed by applications and not by human (Antoniou, et al., 2009). The OWL language has direct influence from the researches in Description Logics and insights from Description Logics particularly on the formalization of the semantics. OWL takes the basic fact-stating ability of RDF (Allemang, et al., 2008) and the class- and property-structuring capabilities of RDF Schema and extends them in important ways. OWL own the ability to 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, here again extending RDFS.

Add to that, OWL pocess the ability to specify which objects (also called "individuals") belong to which classes, and what the property values are of 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 least or at most a certain number of distinct values.

3.4 Semantic Web Rule Language (SWRL)

An inference process consists of applying logic in order to derive a conclusion based on observations and hypothesis. In computer science, interferences are applied through inference engines. These inference engines are basically computer applications which derive answers from a knowledge base. These engines depend on the logics through logic programming. The horn logic more commonly known Horn clause is a clause with at most one positive literal. It has been used as the base of logic programming and Prolog languages (Sterling, et al., 2009) for years. These languages allow the description of knowledge with predicates. Extensional knowledge is expressed as facts, while intentional knowledge is defined through rules (Spaccapietra, et al., 2004). These rules are used through different Rule Languages to enhance the knowledge possess in ontology. The Horn logic has given a platform to define Horn-like rules through sub languages of RuleML (Boley, et al., 2009). There have been different rule languages that have emerged in last few years. Some of these languages that have been evolving rapidly are Semantic Web Rule Language (SWRL) and Jena Rule. Both have their own built-ins to support the rules. With the actual work, SWRL language is used to rich the target concepts but it could be applied to others rule language based on Horn clauses.

Semantic Web Rule Language (Valiente-Rocha, et al., 2010) is a rule language based on the combination of the OWL-DL with Unary/Binary Datalog RuleML which is a sublanguage of the Rule Markup Language. One restriction on SWRL called DL-safe rules was designed in order to keep the decidability of deduction algorithms. This restriction is not about the component of the language but on its interaction. SWRL includes a high-level abstract syntax for Horn-like rules. The SWRL as the form, *antecedent→consequent*, where both antecedent and consequent are conjunctions of atoms written a1 ... an. 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. An OWL data literal is either a typed literal or a plain literal. Variables are indicated by using the standard convention of prefixing them with a question mark (e.g., ?x). URI references (URI refs) are used to identify ontology elements such as classes, individual-valued properties and data-valued properties. For instance, the following rule, equation 1, asserts that one's parents' brothers are one's uncles where parent, brother and uncle are all individual-valued properties.

$$\text{Parent}(?x, ?p) \wedge \text{Brother}(?p, ?u) \rightarrow \text{Uncle}(?x, ?u) \quad (1)$$

The set of built-ins for SWRL is motivated by a modular approach that will allow further extensions in future releases within a hierarchical taxonomy. SWRL's built-ins approach is also based on the reuse of existing built-ins in XQuery and XPath, which are themselves based on XML Schema by using the Datatypes. This system of built-ins should also help in the interoperation of SWRL with other Web formalisms by providing an extensible, modular built-ins infrastructure for Semantic Web Languages, Web Services, and Web applications (OConnor, et al., 2008).

3.5 Swrl built-ins

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. Actually, *Comparisons Built-Ins, Math Built-Ins and Built-Ins for Strings* are already implemented within lots of platform for ontology management like protégé. In the actual work, new processing and topological built-in for the integration of 3D processing and topological knowledge are integrated respectively.

3.6 Discussion

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 actual research, materialized by WiDOP project relay on 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 implement new 3D processing and topological rule inference integrating the correspondent processing and topological built-Ins components in its structure to resolve the problem of object detection and annotation in 3D point clouds based on semantic knowledge.

4. Overview of the general WiDOP model

The problem of automatic object reconstruction remains a difficult task to realize in spite of many years of research. Major problems result from geometry and appearance of objects and their complexity, and impact on the collected data. For example, variations in a viewpoint may destroy the adjacency relations inside the data, especially when the object surface shows considerable geometrical variations. This dissimilarity affects geometrical or topological relations inside the data and even gets worse, when partial occlusions result in a disappearance of object parts. 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. That 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 most 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 (geometry, shape, texture, orientation,...) and topological relations among each other. The decision upon features to be modelled should be affected by other important factors in an analysis step like characteristics of the data, the algorithms and their important features. Such a model might be supported by a DL-OWL ontology structure formed out of RDFS nodes and properties where the nodes represent classes or objects as their instances and the links show relationships of various characteristics. Such a network then contains the knowledge of that type of scene, which has to be processed. This knowledge base will act as basis for further detection and annotation activities and has to work in cooperation with numerical algorithms.

Up to this point, the new conception is still in concordance to other knowledge related set ups, although the degree of modelling goes farther because all relevant scene knowledge

will be integrated. But another aspect will be considered also allowing to significantly improving processing strength. That is to integrate knowledge even on the algorithmic side. This means to make use of the flexibility of knowledge processing for decisions and control purposes inside the algorithmic processing chain. Even a propagation of findings from processing results into new knowledge for subsequent steps should be possible, what would give a completely new degree of dynamics and stability into the evaluation process.

It will finally leads to the conceptual view shown in Fig 5 where the general architecture for the suggested solutions is presented. It's composed of three parts: the knowledge model, the 3D processing algorithms execution and the interaction management and control part labelled WiDOP processing materialized within swrl rules and Built-Ins extensions, ensuring the interaction between the above sited parts. In contrast to existing approaches, we aim at the utilization of previous knowledge on objects. This knowledge can be contained in databases, construction plans, as-built plans or Geographic Information Systems (GIS). The suggested solution named as knowledge based detection of objects in point clouds (WiDOP) has its roots in the knowledge base which then guides individual algorithmic steps. Results from algorithms are also analyzed by the knowledge base and the reasoning engine, then deciding upon subsequent algorithmic steps is taken also from the knowledge base. Accordingly, detected objects and their features are populated to the knowledge base, which will permanently evolve until the work is done.

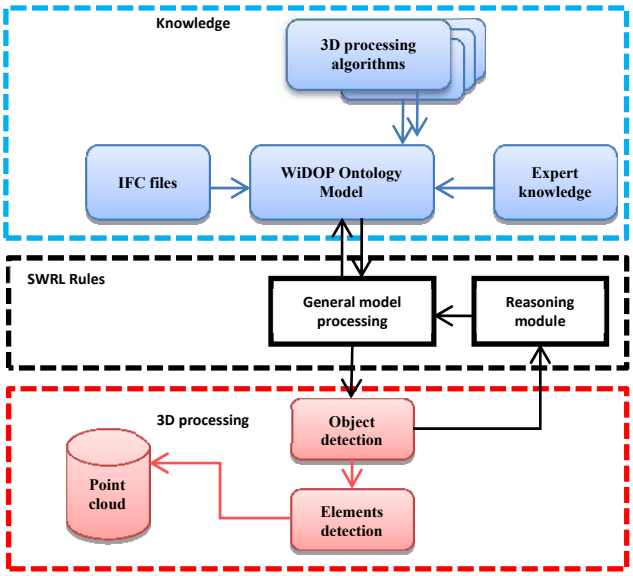


Fig. 5. WiDOP: Overview system.

4.1 The knowledge model

The needed knowledge for such 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. Knowledge will be structured and formalized based on IFC schema, XML files, the domain concept which is the Deutsche Bahn scene in this case and 3D processing domain experts, etc., using classes, instances, relations and rules. An object in the ontology 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, will allow us to classify most of the point region 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 model contains also knowledge extracted from the technical literature of the domain and knowledge from experts of the domain too. In addition, the ontology is as well enriched with knowledge about 3D processing algorithms and populated with the results of experiences undertaken on 3D point clouds, which define the empirical knowledge extracted from point clouds regarding a specific domain of application.

4.2 The 3D processing algorithms

Numerical processing includes a number of algorithms or their combination to process the spatial data. Strategies include geometric element detection (straight line, plane, surface, etc.), projection - based region estimation, histogram matrices, etc. All of these strategies are either under the guidance of knowledge, or use the modelled prior knowledge to estimate the object intelligently and optimally. Alongside with 3D point clouds, various types of input, data sets can be used such as images, range images, point clouds with intensity or color values, point clouds with individual images oriented to them or even stereo images without a point cloud. All sources are exploited for application to particular strategies. Knowledge not only describes the information of the objects, but also gives a framework for the control of the selected strategies. The success rate of detection algorithms using RANSAC (Tarsha-Kurdi, et al., 2007), Iterative Closest Point (Milella, et al., 2006) and Least Squares Fitting (Cantrell, 2008) should significantly increase by making use of the knowledge background. However, we are planning not only to process point data sets but also based on a surface and volume representation like mesh, voxels and bounding Boxes. These methods and others will be selected in a flexible way, depending on the semantic context.

4.3 The WiDOP processing

In order to manage the interaction between the knowledge part and the 3D processing one, a new layer labelled WiDOP processing materialized within rules is created. This layer ensures the control and the management of the knowledge transaction and the decision taken based on SWRL languages and its extensions through several steps explained within

the next section. The semantic within the ontologies expressed through OWL-DL language can be used for further inferences. For instance, the following rule asserts that a Bounding Box with lines higher then 5 m are masts where Masts, Bounding Boxes and lines are all individual-valued properties. The DL syntax related to such an expression is Mast $\sqsubseteq$ $(Bounding\ Box \sqcap \exists\ hasLine.\text{Line} \sqcap \exists\ \text{hasHeight}.\{> 5\})$ while the swrl conversion of such an expression is BoundingBox(?x) $\wedge$ hasLine(?x,?y) $\wedge$ hasHeight (?y,?h) $\wedge$ swrlb:GreaterThan (?h, 5) → Mast(?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. This system of built-ins should as well 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. Many built-ins are defined. 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 topological processing.

4.4 Interaction process

To focus on the suggested method for the combination of the Semantic Web technologies and the 3D processing algorithms, Fig 6 illustrates an UML sequence diagram that represents the general design of the proposed solution. Hence, the purpose is to create a more flexible, easily extended approach where algorithms will be executed reasonably and adaptively on particular situations following an interaction process.

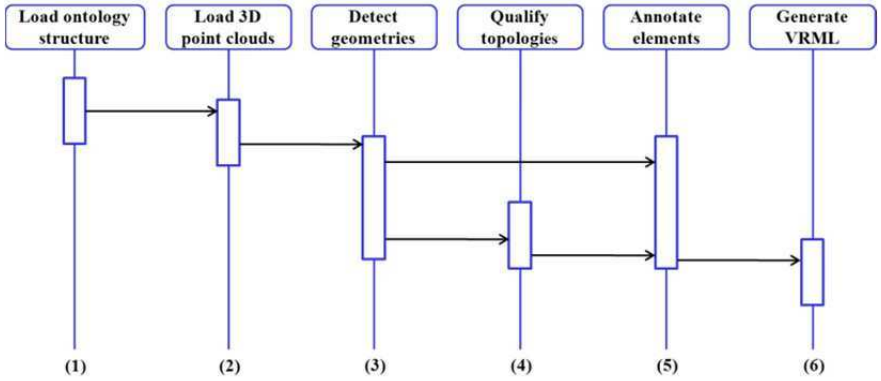


Fig. 6. The sequence diagram of interactions between the laser scanner, the 3D processing, the knowledge processing and the knowledge base

The processing steps can be detailed where three main steps aim at detecting and identifying objects.

(3) From 3D point clouds to geometric elements.
(4) From geometry to topological relations.
(5) From geometric and/or topological relations to semantic annotated elements.

As intermediate steps, the different geometries within a specific 3D point clouds are detected and stored within the ontology structure. Once done, the existent topological relations between the detected geometries are qualified and then populated in the knowledge base. Finally, detected geometries are annotated semantically, based on existing knowledge's related to the geometric characteristics and topological relations, where the input ontology contains knowledge about the Deutsche Bahn railway objects and knowledge about 3D processing algorithms.

5. Description of the WiDOP knowledge base

This section discusses the different aspects related to the domain concept top level ontology structure installed behind the WiDOP Deutsche Bahn prototype (Ben Hmida, et al., 2010). It´s composed mainly by the classes and their relationships. Hence, we try to discuss theses component in term of axiom representing them.

The domain ontology presents the core of our 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 Deutsche Bahn elements from the 3D processing point of view. The created ontology is used basically for two purposes:

- To guide the processing algorithm sequence creation based on the target object characteristics.
- To ensure the semantic annotation of the different detected objects inside the target scene.

In fact, the ontology is managed through different components of description logics where the class axioms contain their own prefixes used to define their names. One of the big advantages of using prefix is that the same class could be used by applying different prefix for the class. Other advantages include the simplification in defining the resource and to solve the ambiguity for different context. The hierarchical structure of the top level class axioms of the ontology is given in Fig 7, where we find five main classes within other data and objects properties able to characterize the scene in question.

5.1 Class axioms

The class axiom `DC:DomainConcept` which represents the different object found in the target scene can be considered the main class in this ontology as it is the class where the target objects are modelled, this class is further specialized into classes representing the different detected object. However, the importance of other classes cannot be ignored. They are used to either describe the object geometry through the `Geom:Geometry` class axiom by defining its geometric component or the bounding box of the object that indicate its coordinates or to either describe its characteristics through the `Charac:Characteristics` class axiom. Additionally, the suitable algorithms are automatically selected based on its compatibility within the object geometry and characteristics via the `Alg:Algorithm` class. Add to that, other classes, equally significant, play their roles in the backend. The connection between the basic mentioned classes is carried out through object and data properties axioms.

5.2 Properties axioms

The properties axioms define relationship between classes in the ontology. They are also used to relate an object to other via topological relations. Actually, we found four major object properties axioms in the top level ontology which have their specialized properties for the specialized activities, Fig. 7, `DC:hastopologicRelation`, `Alg:isDeseignedFor`, `Geom:hasGeometry`, `Charac:hasCharacteristics`.

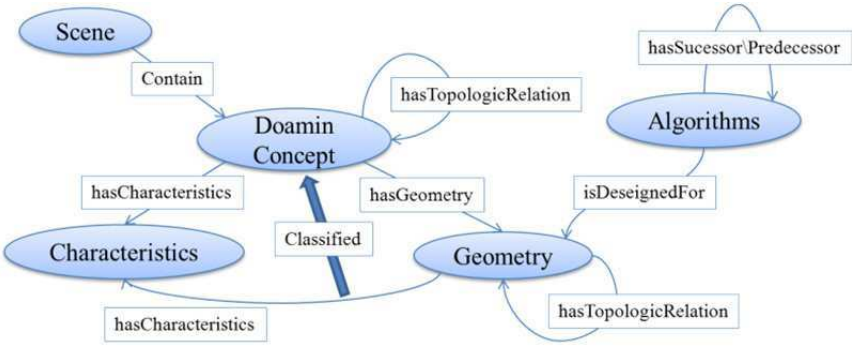


Fig. 7. Ontology general schema overview.

5.3 Created knowledge layers

Following to above considerations and with respect to technological possibilities, the current ontology will be modelled in various levels. In principle, we have to distinguish between object-related knowledge and algorithmic related knowledge. We therefore have a layer of the object knowledge and a layer of the algorithmic knowledge containing the respective semantic information.

5.3.1 Layers of object knowledge

The object knowledge layer will be classified in three categories: geometric, topological and semantic knowledge representing a certain scenario (Whiting, 2006) Therefore we distinguish between:

- Deutsche Bahn Scene knowledge
- Geometric knowledge
- Topological knowledge

Layer of the Deutsche Bahn Scene knowledge

The layer of object knowledge contains all relevant information about the objects and elements which might be found within a Deutsch Bahn scene. This might comprise a list such as: {Signals, Mast, Schalanlage, etc.}. They are used to fix either the main scene within its point clouds file and size through attributes related to the scene class, or even to characterize detected element with different semantic and geometric characteristics. The created knowledge base related to the Deutsche Bahn scene has been inspired next to our discussion with the domain expert and next to our study based on the official Web site for the German rail way specification ”http://stellwerke.de”. An overview of the targeted

elements, the most useful and discriminant characteristics to detect it and their inter-relationship is presented in Table 1 .

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	BigMast	More than 6m		
	NormalMast	Between 5 and 6		
Schaltanlage	Schalthause	Less than 1m		
	SchaltSchrank	Less than 0,5m		

Table 1. Example of the Deutsche Bahn scene objects

Table 1 shows a possible collection of scene elements in case of a Deutsche Bahn scene. They may be additionally structured in a hierarchical order as might be seen convenient for a scene while Fig8 shows the suggested taxonomical structure to model them within the OWL language.

Basically, a railway signal is one of the most important elements within the Deutsche Bahn 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 section of track 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. Actually, 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 electricity born for the energy alimentation. Usually, masts are distant from 50 m from to others. Finally, the `DC:Schaltanlage` elements present small electric born connected to the ground. For detection purpose, we define for example a signal as:

$$\text{DC: Signal} \sqsubseteq Geom{:}\,VerticalBB \sqcap \exists\, hasheight.\{> 6\} \sqcap \exists\, hasDistanceFrom.\,DC{:}\,Mast\{> 50\}$$

The above cited concepts are extended by relations to other classes or data. As an example, the data property `Geom:has_Bounding_Box` aims to store the placement of the detected object in a bounding box defined by its eight 3D points characterized by x, y and z values each one.

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. 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.

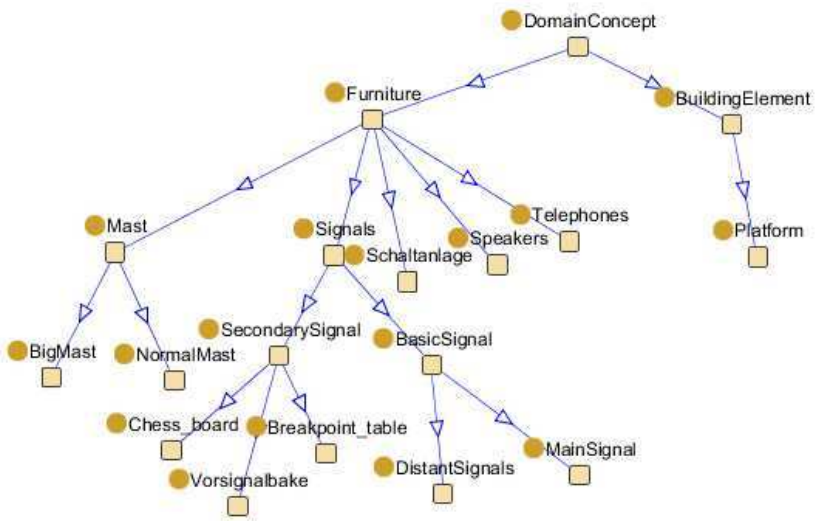


Fig. 8. Example of the DB scene objects modelling.

Layer of the geometric knowledge

Geometrical knowledge formulates geometrical characteristics to the physical properties of scene elements. In the simplest case, this information might be limited to few coordinates expressing a bounding box containing the object. However, for elements being 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, 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 annotation process. For this reason, we opt to study the different geometric features related to the cited semantic elements, then, use only the discriminant one as basic features for a given object. The following table gathers the object characteristics together regarding the properties of a bounding box, Table 2, Fig 9.

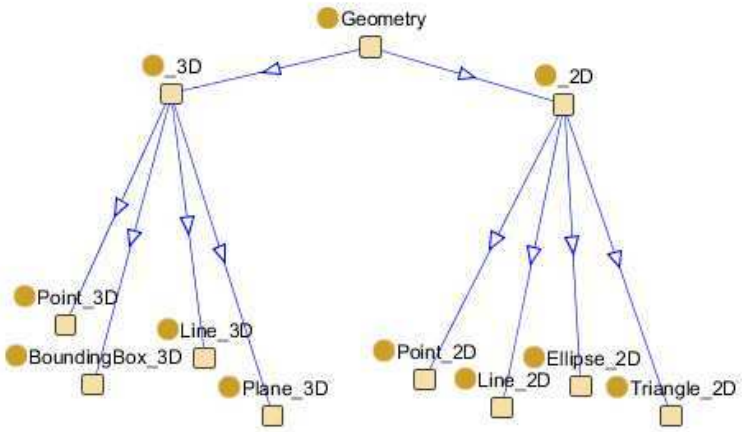


Fig. 9. The geometry class hierarchy.

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	More than 6m	2 or 4 vertical lines	0
	NormalMast	Between 5 and 6	2 or 4 vertical lines	0
Schaltanlage	Schalthause	Less than 1m		1 Vertical plane 1 Horizontal plane
	SchaltSchrank	Less than 0,5m		1 vertical plane

Table 2. Geometric characteristics overview.

Layer of the topologic knowledge

While exploring the railway domain, lots of standard topological rules are imposed; such rules are used to help the driver and to ensure the passengers' security. From our point of view, the created rules are helpful also to verify and to guide the annotation process. In fact, topological knowledge represents adjacency relationships between scene elements. For instance, and in case of the Deutsche Bahn scene, the distance between the distant signal and the main one corresponds to the stopping distance that the trains require. The stopping distance shall be set on specific route and is in the main lines often 1000 m or in a rare case 700 m. Add to that, three to five Vorsignalbake are distant from 75m while then the last one is distant 100m from the distant signal, Fig.11.

At semantic view, topological properties describe adjacency relations between classes. For example, the property `Topo:isParallelTo` allows characterizing two geometric concepts

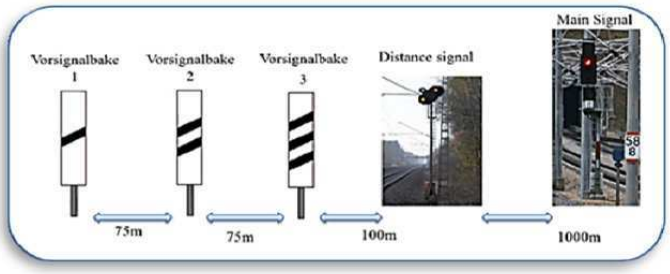


Fig. 10. Topological rules.

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.

5.3.2 Layer of 3D processing knowledge

The 3D processing algorithmic layer contains all relevant aspects related to the 3D processing algorithms. It contains algorithm definitions, properties, and geometries related to each defined algorithms. An importance 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 to 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. Fig11 presents the Mast 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.

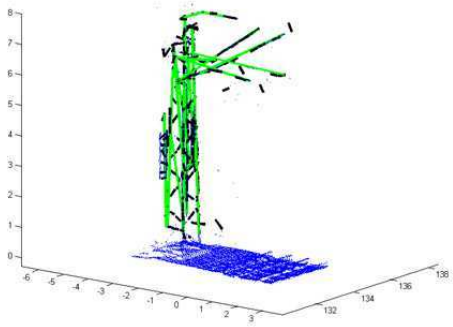


Fig. 11. Mast detection.

Next to the 3D expert recommendation, knowledge within the Table 3 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 Objects Detection	PointCloud	Point_2D	Vertical gemetry	None
Segmentationin2D	Point_2D PointCloud	SubPointCloud	Vertical gemetry	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
CheckPerpendicular	Line_3D	Boolean	Geometry containing Perpendicular elements	LinesDetectionin3DbyRANSAC
CheckParallel	Line_3D	Boolean	Geometry containing Parallel elements	LinesDetectionin3DbyRANSAC

Table 3. 3D processing algorithms and experts observations

The specialized classes of the `Alg:Algorithm` axiom are representing all the algorithms developed in the 3D processing layer. 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.

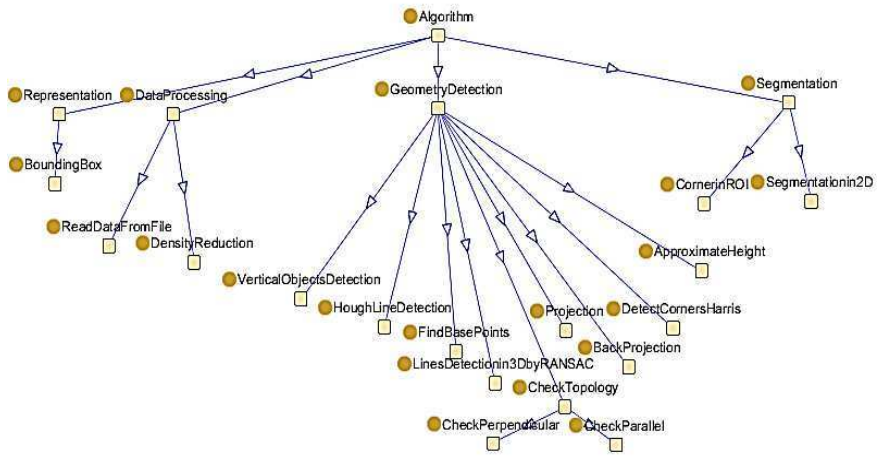


Fig. 12. Hierarchical structure of the Algorithm class.

The following section presents in details the semantic integration process undertaken in the WiDOP solution to detect and annotate semantically the eventual semantic elements.

6. Intelligent process

The basic strength of formal ontology is their ability to reason in a logical way based on Descriptive Logic language DL. As seem, 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 OWL's 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 relationship in term of conditions or even rules. These rules are used through different rules languages to enhance the knowledge possess in an ontology.

Within the actual research, the domain ontologies are used to define the concepts, and the necessary and sufficient conditions on them. These conditions are of value, because they are used to populate new concepts. For instance, the concept `Goem:Vertical_BoudinBox` can be specialized into `DC:Signal` if it contains a `Goem:VerticalLines`. Consequently, the concept `DC:Signal` will be populated with all `Goem:Vertical_BoudinBox` if they are linked to a `Goem:VerticalLines` with certain parameters. In addition, the rules are used to compute more complex results such as the topological relationships between objects. For instance, the relations between two objects are used to get new efficient knowledge about the object. The ontology is than enriched with this new relationship. The topological relation built-ins are not defined in the SWRL language. Consequently, the language was extended. To support the defined use cases, two basic further layers to the semantic one are added to ontology in order to ensure the geometry detection and annotation process tasks. These operations are the 3D processing and topological relations qualification respectively.

6.1 Integration of 3D processing operations

The 3D processing layer 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 processing layers and the semantic one. In addition, it contains the different algorithm definitions, properties, and the related geometries to the each defined algorithms. An importance 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 (e.g. Table 4). The equation 2 illustrate the "3D_swrlb_Processing: VerticalElementDetection" built-ins for example, it aims at the detection of geometry with vertical orientation. The prototype of the designed Built-in is:

$$\text{3D_swrlb_Processing:VerticalElementDetection(?Vert, ?Dir)} \quad (2)$$

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 bounding boxes, representing a rough position and orientation of the detected object. Table 4 shows the mapping between the 3D processing built-ins, which is computer 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 4. 3D processing Built-Ins mapping.

6.2 Integration of topological operations

The layer of the topological knowledge represents topological relationships between scene elements since the object properties are also used to link an object to others by a topological relation. For instance, a topological relation between a distant signal and a main one can be defined, as both have to be distant from one kilometer. The qualification of topological relations in to the semantic framework is done by new topological Built-Ins. This step aims at verifying certain topology properties between detected geometries. Thus, 3D_Topologic built-ins have been added in order to extend the SWRL language. Topological rules are used to define constrains between different elements. After parsing the topological built-ins and its execution, the result is used to enrich the ontology with relationships between individuals that verify the rules. Similarly to the 3D processing built-ins, our engine translates the rules with topological built-ins to standard rules, Table 5.

Function	Correspondent topologic Built-Ins	Correspondent object property	Characteristics
Upper	3D_swrlb_Topology:Upper(?x, ?y)	Upper(?x,?y)	Transitive
Intersect	3D_swrlb_Topology:Intersect(?x, ?y)	Intersect(?x,?y)	Symmetric
Distance	3D_swrlb_Topology: Distance (?x, ?y,?d)	Distance(?x, ?y, ?d)	Symmetric

Table 5. Example of topological built-ins.

6.3 Guiding 3D processing algorithms

Actually, the created knowledge base aims to satisfy two basic purposes, which are, guiding the processing algorithm sequence creation based on the target object characteristics, and facilitate the semantic annotation of the different detected objects inside the target scene. Let's remember that one of the main ideas behind our suggestions is to direct, adapt and select the most suitable algorithms based on the object's characteristics. In fact, one algorithm could not detect and recognize different existent objects in the 3D point clouds, since they are distinguished by different shapes, size and capture condition. The role of knowledge is to provide not only the object's characteristics (shape, size, color...) but also

object's status (visibility, correlation) to algorithmic part, in order to adjust its parameters to adapt with a current situation. Based on these observations, we issue a link from algorithms to objects based on the similar characteristics as Fig.13 shows.

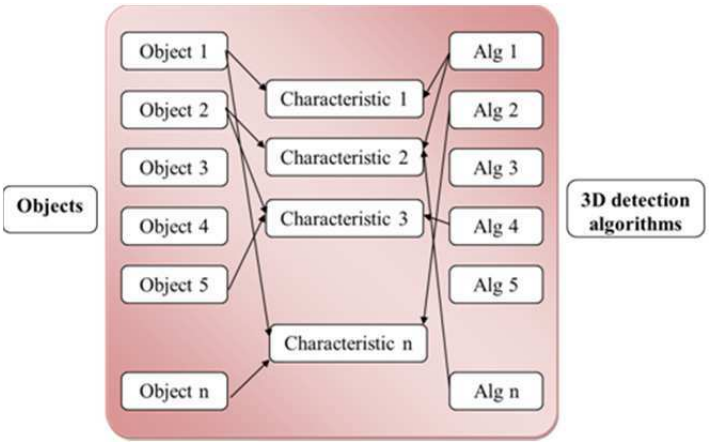


Fig. 13. Algorithms selection based on object's characteristics.

In fact, knowledge controls one or more algorithms for detecting an object. To do, a match between the object's characteristics and characteristics that a certain algorithm can be used for is achieved. For example, object O has characteristics: C_1, C_2, C_3; and algorithm A_i can detect characteristic C_1, C_3, C_4, while the algorithm A_j can detect characteristic C_2, C_5. Then, the decision algorithm will select A_i and A_j since these algorithms have the capability to detect the characteristics of an object O. The set of characteristics are determined by the object's properties such as geometrical features and appearance. Once done, selected algorithms will be executed and target characteristics will be detected. Let's recall that the whole process takes as input, the 3D point clouds scenes, and an ontology structure presenting a knowledge base to manipulate objects, geometries, topologies and relations (Object and data property) and produces as an output, an annotated scene within the same ontology structure. As intermediate steps, the different geometries within a specific 3D point cloud scene are detected and stored in the ontology structure. Once knowledge about geometries and the topologies are experienced, SWRL rules aim at qualifying and annotating the different detected geometries. The following equation 3 illustrates the DL definition of a Mast element while the simple example, equation 4, shows how a SWRL rule can specify the class of a VerticalBoundingBox, which is of type Mast regarding its altitude. The altitude is highly relevant only for this element.

$$\text{DC. Mast} \sqsubseteq \mathit{G}\text{eom. VerticalBB} \sqcap \exists \text{ hasheight.}\{> 6\} \tag{3}$$

$$\text{3DProcessingSWRL: VerticalElementDetection(? Vert, ? dir)} \quad \text{altitude (? x, ? alt)}$$

$$\text{^swrlb: moreThan (? alt, 6)} \quad \rightarrow \text{ DC: Mast (? Vert)} \tag{4}$$

In other cases, geometric knowledge is not sufficient for the previous process. In such scenario, the topological relationships between detected geometries are helpful to manage

the annotation process, equation 5. Equation 6 demonstrates how semantic information about existing objects is used conjunctly with topological relationships in order to define the class of another object.

$$\text{DC: Mast} \sqsubseteq DC{:}\,Mast \sqcap Geom{:}\,VerticalBB \sqcap \exists\ hasheight\{> 6\}$$
$$\sqcap \exists hasDistanceFrom.\,DC{:}\,Mast.\{> 50\} \quad (5)$$

$$\text{DC: Mast (? vert1)} \quad \text{VerticalBB (? Vert2)} \quad \text{hasDistanceFrom}$$
$$(\text{? vert1, ? vert2, 50}) \rightarrow \text{DC: Mast(? vert2)} \quad (6)$$

7. WiDOP prototype

The created WiDOP prototype takes in consideration the adjustment of the old methods and, in the meantime, profit from the advantages of the emerging cutting-edge technology. From the principal point of view, the developed system still retains the storing mechanism within the existent 3D processing algorithms; in addition, suggest a new field of detection and annotation, where we are getting a real-time support from the created knowledge. Add 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 the 3D packages results.

The process enriches and populates the ontology with new individuals and relationships between them. In addition, the created WiDOP platform offers the opportunity to materialize the annotation process by the generation and the visualization based on a VRML structure, (W3C, 1995) alimented from the knowledge base. It ensures an interactive visualization of the resulted annotation elements beginning from the initial state, to a set of intermediate states coming finally to an ending state, Fig 17 where 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. The created system is composed of three main parts.

- Generation of a set of geometries from a point could file based on the target object characteristics.
- Computation of business rules with geometry, semantic and topological constrains in order to annotate the different detected geometries.
- Generation of a VRML model related to the scene within the detected and annotated elements.

As a first impression, the system responds to the target requirement since it would take a point cloud of a facility as input and produce a fully annotated as-built model of the facility as output.

7.1 System evaluation

For the demonstration of the created system, a scanned point cloud section related to Deutsch Bahn scene in the city of *Nürnberg* was extracted. While the last one measure 87 kms, we have just taken a small scene of 500m. It contains a variety of the target objects. The

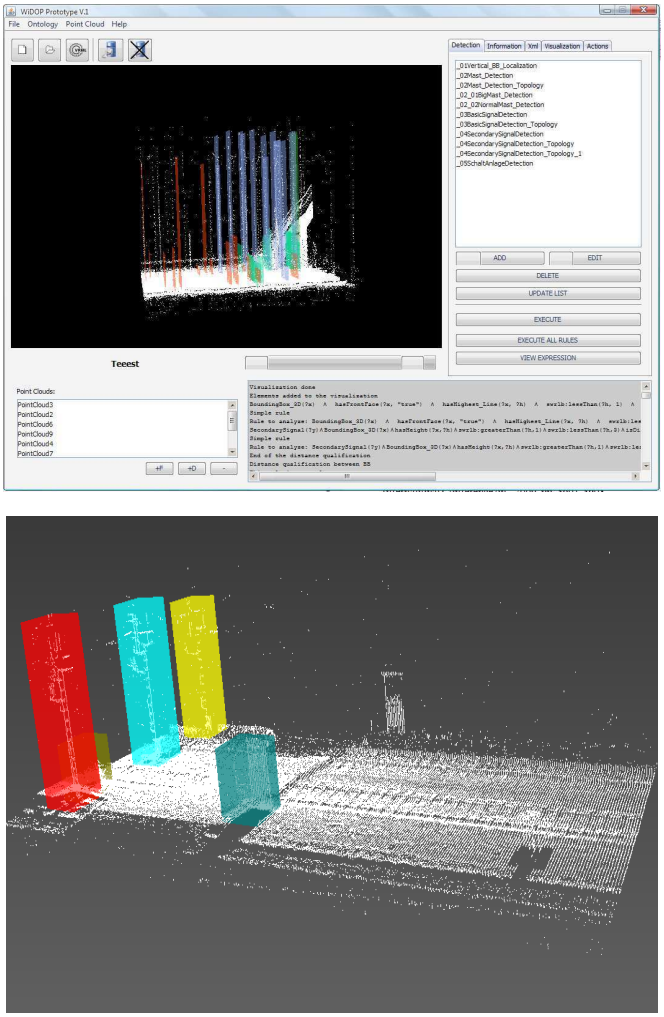


Fig. 14. Snapshot of the WiDOP prototype (top), Detected and annotated elements visualization within VRML language (bottom).

whole scene has been scanned using a terrestrial laser scanner fixed within a train, resulting in a large point cloud representing the surfaces of the scene objects. Within the created prototype, different SWRL rules are processed. First, geometrical elements will be searched in the area of interest based on dynamic 3D processing algorithm sequence created depending on semantic object properties. The second step aims to identify existing topologies between the detected geometries. Thus, useful topologies for geometry annotation are tested. Topological Built-Ins like `topo:isConnected`, `topo:touch`, `topo:Perpendicular`, `topo:isDistantfrom` are created. As a result, relations found between geometric elements are propagated into the ontology, serving as an improved knowledge base for further processing and decision.

The last step consists in annotating the different geometries. Vertical elements with certain characteristics can be annotated directly. Subsequently, further annotation 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, our prototype allows the combination of semantic information and topological ones that can deduce more robust results minimizing the false acceptation rate. The extracted scene contains 37 elements. As well, in most cases, our annotation process is able to affect the right label to the detected Bounding box based on knowledge on its component, its internal and external topology where among 13 elements are classified as Masts, three as a SchaltAnlage and 18 signals, Table 6, Table 7, Fig.15.

	Scene Size	Detected	Bounding Box	Annotated elements	Truth data
Scene1	500m		105	34	37

Table 6. Detected Element within the scene and annotated ones.

	Masts	signal	Schaltanlage
Annotated	13	18	3
Truth data	12	20	5

Table 7. Detected and annotated elements within the scene1.

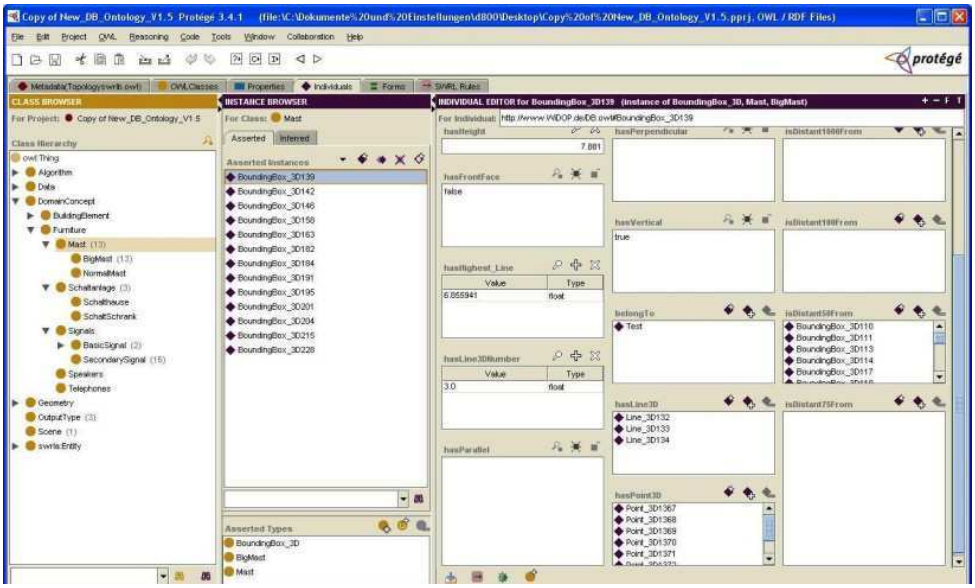


Fig. 15. Knowledge base after the annotation process.

Some limits are detected while making extra tests, especially with the SchaltAnlage object detection where the rate of false detection still high. Before explaining the reason behind this false detection, let's recall that the Schaltanlage present very small electronic boxes installed on the ground. In the some cases, lots of bounding boxes are detected where a high average of them presents small noise on the ground. The reason for the false annotation is the lack of semantic characteristics related to such elements since, until now; there is no real internal or external topology neither internal geometric characteristic that discriminate such an element compared to others.

8. Conclusion and discussion

By this chapter, we tried to contribute on the ongoing enhancement of the Semantic Web technologies through focusing on the possibility of integrating 3D processing and spatial topological components within its framework. It makes an attempt to cross the boundary of using semantics within the 3D processing researches to provide interoperability and takes it a step forward in using the underlying knowledge technology to provide 3D processing and topological analysis through knowledge.

The presented 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, the processing algorithms and the 3D spatial topological relations.

The WiDOP framework is primarily designed to facilitate the object detection and recognition in 3D point clouds. It being 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 modeled within an ontology structure. SWRL rules are used to control the 3D processing execution, the topological qualification and finally to annotate 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 produce fully annotated scene within a VRML model file. The suggested solution for this challenging problem has proven its efficiency through real tests within the Deutsche Bahn scene. The creation of processing and topological 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.

The benefits of the emerging Semantic Web technology through its knowledge tools are quite visible to the convention technologies that rely heavily on database systems. More precisely, the benefits that have been experienced during the design and development of the WiDOP platform is quite strong. The flexibility nature of ontology based system allows integrating new components at any time of development and even implementations.

Future work will include the integration of new knowledge intervening during the annotation process. It will also include the update of the general platform architecture, by ensuring more interaction between the scene knowledge and the 3D processing. Add to that, it will include a more robust identification and annotation process of objects based on each object characteristics add to the integration of new 3D parameter knowledge's that can

intervene within the detection and annotation process to make the process more flexible and intelligent.

9. Acknowledgment

This paper chapter work performed in the framework of the research project funded by the German ministry of research and education under contract No. 1758X09. The authors cordially thank for this funding.

10. References

Allemang, D. & Hendler, J A. (2008). *Semantic web for the working ontologist: modeling in RDF, RDFS and OWL,* Morgan Kaufmann, ISBN 9780123735560, Amsterdam.

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

Horrocks, I. Peter, F. Schneider, P. Deborah, L. McGuinness & Christopher, A. (2007). *OWL: a Description Logic Based Ontology Language for the Semantic Web*, Proc of the 21st Int Conf on Automated Deduction.

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

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

Balzani, M. Santopuoli, N. Grieco, A. & Zaltron, N. (2004) Laser Scanner 3D Survey in Archaeological Field, *the Forum Of Pompeii*, pp. 18-21, 2004.

Ben Hmida, H. Cruz, C. Nicolle, C. & Boochs, F. (2010). From 3D point clouds to semantic object, *KEOD, theInternational Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management,* Paris, 2011.

Ben Hmida, H. Cruz, C. Nicolle, C. & Boochs, F. (2010). Semantic-based Technique for the Automation the 3D Reconstruction Process, *SEMAPRO 2010, The Fourth International Conference on Advances in Semantic Processing*, Florence, Italy, pp. 191-198, 2010.

Berners-Lee, T. (1998). Semantic web road map, 1998.

Boley, H. Osmun, T M. & Craig, B L. (2009). WellnessRules: A Web 3.0 Case Study in RuleML-Based Prolog, *N3 Profile Interoperation*, Vol. 5858, p 43, 2009.

Boochs, F. Marbs, A. Ben Hmida, H. Truong, H. Karmachaiya, A. Cruz, C. Habed, A. Nicolle, C. & Voisin, Y. (2010). Integration of knowledge to support automatic object reconstruction from images and 3D data. *International Multi-Conference on Systems, Signals & Devices*, Sousse Tunisia : pp. 1-13, 2011.

Bosche, F. & Haas, CT. (2008). Automated retrieval of 3D CAD model objects in construction range images *Automation in Construction,* Elsevier, Vol. 17, pp. 499-512, 2008.

Calvanese, D. De Giacomo, G. Lembo, D. Lenzerini, M & Rosati, R. (2005).DL-Lite: Tractable description logics for ontologies, Vol. 20, p. 602, 2005.

Campbell, RJ. & Flynn, PJ.(2001). A survey of free-form object representation and recognition techniques, *Computer Vision and Image Understanding*, Vol. 81, pp. 166-210, 2001.

Cantrell, CA. (2008). Technical Note: Review of methods for linear least-squares fitting of data and application to atmospheric chemistry problems, *Atmospheric Chemistry and Physics Discussions*. Vol. 8, pp. 6409-6436, 2008.

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

CASCADE OPEN. (2000). OpenCascade-an open source library for BRep solid modeling, 21.10.2010, Available from http://www.opencascade.org.

Corporation Leadwerks. (2006) . What is Constructive Solid Geometry?, 21.10.2010, Available from http://www.leadwerks.com/files/csg.pdf.

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*, Vol. 4, pp. 63-73, 2000.

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*, Vol. 16, pp. 38-45, 2001.

Goldberg H.E. (2005). State of the AEC industry: BIM implementation slow, but inevitable, *Revista CADalyst*, maio 2005.

Grau, B.C. Horrocks, I. Motik, B. Parsia, B. Patel-Schneider, P. & Sattler, U. (2008). OWL 2: The next step for OWL ,*Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, Vol. 6, pp. 309-322, 2008.

Gruber, T. (2008). What is an Ontology, *Encyclopedia of Database Systems*, Springer-Verlag, Vol. 1, 2008.

Hajian, H & Becerik-Gerber, B. (2009). A Research Outlook for Real-Time Project Information Management by Integrating Advanced Field Data Acquisition Systems and Building Information Modeling, *Computing in Civil Engineering*, p 83-94, 2009.

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*, Vol. 21. - p. 79, 2004.

Horrocks, I, Bechhofer, S. (2008). Semantic Web, *Web Accessibility*. Springer, pp. 315-330, 2008.

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*, Elsevier, Vol. 5, pp. 16-27, 2007.

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

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

Leica, Cyclone. (2011), Leica Geosystems Leica Geosystems, 21.10.2010, Available from http://hds.leica-geosystems.com/en/Leica-Cyclone_6515.htm, 2011.

McGuinness, D L. & Patel-Schneider, P F.(2003). From description logic provers to knowledge representation systems. pp. 265-281, 2003.

Milella, A & Siegwart, R.(2006). Stereo-based ego-motion estimation using pixel tracking and iterative closest point, IEEE Computer Society, p. 21-21, 2006.

Noy, N .F. & McGuinness .(2001). D L. Ontology development 101: A guide to creating your first ontology. *Citeseer*, 2001.

OConnor, M.J. Shankar, R. Nyulas, C. Tu, S. & Das, A.(2008). Developing a Web-based application using OWL and SWRL, *AAAI Spring*, 2008.

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

Pu, S. & Vosselman, G. (2009). Knowledge based reconstruction of building models from terrestrial laser scanning data, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 64, pp. 575-584, 2009.

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, IROS*, 2009.

Schmidt-Schauß, M, & Smolka, G.(2009). Attributive concept descriptions with complements, *Artificial intelligence, Elsevier,* Vol. 48, pp. 1-26, 1991.

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, Elsevier, Vol. 5, pp. 51-53, 2007.

Spaccapietra, S. Cullot, N. Parent, C. & Vangenot, C.(2004): On spatial ontologies, Citeseer, 2004.

Sterling, L. & Taveter, K A.(2009). Logic Programming Perspective on Rules, *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*, IGI Global snippet, Vol. 1. p. 195, 2009.

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, Science And Technology, Vol 36, p. 407- 412 - 2007.

Hustadt, U. Motik, B & Sattler, U. (2006). KAON2, 21.10.2010, Available from http://kaon2.semanticweb.org.

Uren, V. Cimiano, P. Iria, J. Handschuh, S. Vargas-Vera, M. Motta, E. & Ciravegna, F. 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*, Elsevier, Vol. 4, pp. 14-28.

Vale, S & Hammoudi, S.(2009). An Architecture for the Development of Context-aware Services based on MDA and Ontologies, *Proceedings of the International MultiConference of Engineers and Computer Scientists,* Citeseer, Vol. 1, 2009.

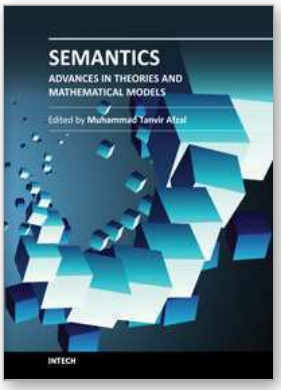
Valiente-Rocha, P & Lozano-Tello, A.(2010). Ontology and SWRL-Based Learning Model for Home Automation Controlling, *Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 2010)* ,pp. 79-86, 2010.

Vanland, R. Cruz, C. & Nicolle, C. (2008). IFC and building lifecycle management, *Automation in Construction. Elsevier,* Vol. 18. - pp. 70-78, 2008.

W3C, VRML Virtual Reality Modeling Language, , 21.10.2010, Available from http://www.w3.org/MarkUp/VRML/.

Whiting, E J.(2006). Geometric, Topological & Semantic Analysis of Multi-Building Floor Plan Data, phdthesis, 2006.

Yue, K. and Huber, D. and Akinci, B. and Krishnamurti, R. The ASDMCon project: The challenge of detecting defects on construction sites, *International Symposium on 3D Data Processing Visualization and Transmission,* IEEE Computer Society, Vol. 0, pp. 1048-1055, 2006.



Semantics - Advances in Theories and Mathematical Models

Edited by Dr. Muhammad Tanvir Afzal

ISBN 978-953-51-0535-0
Hard cover, 284 pages
Publisher InTech
Published online 25, April, 2012
Published in print edition April, 2012

The current book is a nice blend of number of great ideas, theories, mathematical models, and practical systems in the domain of Semantics. The book has been divided into two volumes. The current one is the first volume which highlights the advances in theories and mathematical models in the domain of Semantics. This volume has been divided into four sections and ten chapters. The sections include: 1) Background, 2) Queries, Predicates, and Semantic Cache, 3) Algorithms and Logic Programming, and 4) Semantic Web and Interfaces. Authors across the World have contributed to debate on state-of-the-art systems, theories, mathematical models in the domain of Semantics. Subsequently, new theories, mathematical models, and systems have been proposed, developed, and evaluated.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Helmi Ben Hmida, Christophe Cruz, Frank Boochs and Christophe Nicolle (2012). From Unstructured 3D Point Clouds to Structured Knowledge - A Semantics Approach, Semantics - Advances in Theories and Mathematical Models, Dr. Muhammad Tanvir Afzal (Ed.), ISBN: 978-953-51-0535-0, InTech, Available from: http://www.intechopen.com/books/semantics-advances-in-theories-and-mathematical-models/from-unstructured-3d-point-clouds-to-structured-knowledge-a-semantics-approach

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821