

Use of Geospatial Analyses for Semantic Reasoning

Ashish Karmacharya^{1,2}, Christophe Cruz², Frank Boochs¹, Franck Marzani²

¹Institut i3mainz, am Fachbereich 1 - Geoinformatik und Vermessung, Fachhochschule Mainz, Lucy-Hillebrand 2, 55128 Mainz

{ashish, boochs}@geoinform.fh-mainz.de

²Laboratoire Le2i, UMR-5158 CNRS, UFR Sciences et Techniques, Université de Bourgogne, B.P. 47870, 21078 Dijon Cedex, France

{christophe.cruz, franck.marzani}@u-bourgogne.fr

Abstract. This work focuses on the integration of the spatial analyses for semantic reasoning in order to compute new axioms of an existing OWL ontology. To make it concrete, we have defined Spatial Built-ins, an extension of existing Built-ins of the SWRL rule language. It permits to run deductive rules with the help of a translation rule engine. Thus, the Spatial SWRL rules are translated to standard SWRL rules. Once the spatial functions of the Spatial SWRL rules are computed with the help of a spatial database system, the resulting translated rules are computed with a reasoning engine such as Racer, Jess or Pellet.

Keywords: OWL, SWRL, Spatial functions, GIS system, Built-ins, Spatial Knowledge Reasoning.

1 Introduction

This paper discusses a method to integrate the spatial technologies and Web semantic technologies. This is undertaken by using rules. Actually, they have always played an important role for knowledge-based systems [9]. In the semantic Web context, rules are defined with the help of the Rule Markup Language. The derived language “Semantic Web Rule Language” (SWRL) combines the RuleML and the OWL-DL [1]. The method consists in extending the SWRL language with spatial built-ins. The Spatial SWRL API, part of the project ArchaeoKM [5], [6], [7], provides an authoring environment for the definition of rules and allows the execution of these rules. The results of this work are applied to the domain of archaeology and the project ArchaeoKM. The main concept behind ArchaeoKM is to use knowledge possessed by archaeologists to manage the excavated information. ArchaeoKM facilitates archaeologists to manage the information and the knowledge concerning the findings and objects collected on the site. This is done by defining the geo-localization of objects, the enrichment and the population of an ontology of domain. Presently, it concerns the domain of industrial archaeology. This project has already been presented in CAA 2009 [6].

The Open Geospatial Consortium (OGC) plays a major role to develop a consensus among different stakeholder on various aspect of geospatial technology. The OGC is concerned by the data interoperability and has developed different standards for this. In addition, groups like Geospatial Incubator have taken the works of OGC to formulate steps in updating the W3C geo vocabulary and preparing the groundwork to develop comprehensive geospatial ontology [11]. The domain of archaeology benefits from this work and could surely be of benefit for lot of others domain. As a proof of concept, we present an example of what is possible to compute with our work. For instance, it is possible to determine to identify possible flooding zones according to river bank bursts due to excessive water during rainy season. This is a very common exercise for a flood management system in hydrology and it provides interesting clues for industrial archaeology.

$$\text{River}(?x) \wedge \text{Building}(?y) \wedge \text{spatialswrlb:Buffer}(?x, 50, ?z) \wedge \text{spatialswrlb:Intersection}(?z, ?y, ?res) \rightarrow \text{isLiableToFloodingBy}(?y, ?x) \quad (1)$$

The next section covers the knowledge representation and the reasoning process by presenting the Semantic Web technologies starting from the OWL language to the SWRL Built-ins. Section 3 presents the cutting edge technologies. Section 4 deals with the spatial representation in GIS systems. This section includes the presentation of the spatial relationship functions and the spatial processing functions. Section 5 presents the ontology adjustment process which is necessary to do before the processing of spatial rules. Section 6 gives a description of the Spatial Built-ins related the spatial functions. The last section concludes the papers.

3 The cutting edge technologies

This section deals with a short introduction to the main Semantic Web technologies. The OWL language that allows the definition of ontologies of domain, SWRL that allows the definition of rules on ontologies and SWRL built-ins that allow to compute advance processes.

3.1 OWL

OWL is a knowledge representation language and a standard (W3C recommendation) for expressing ontologies in the Semantic Web. The OWL language facilitates greater machine understandability of Web resources by providing additional constructors for building class, property descriptions and new axioms, along with a formal semantics. Concepts are sets of classes of individual objects. Classes provide an abstraction mechanism for grouping resources with similar characteristics [4]. In any graphical representation of knowledge classes are represented through the nodes. Descriptions on OWL classes are discussed in details in [4]. A property restriction is an unnamed class containing all individuals that satisfy the restriction. Properties are binary relationships between two objects. In

general they are the relationships between two classes which apply to the individual of those classes. They are known as roles in description logic and are represented through links in the graphical representation. OWL provides two main categories of properties: Object properties – relationships between concepts and consequently instances of the concepts and Data properties – relation of an instance to the data value.

3.2 SWRL

Semantic Web Rule Language (SWRL) [1] is a rule language based on the combination of the OWL-DL (SHOIN(D)) with Unary/Binary Datalog RuleML which is a sublanguage of the Rule Markup Language. One restriction on SWRL called DL-safe rules was design 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* \rightarrow *consequent*, where both antecedent and consequent are conjunctions of atoms written $a_1 \wedge \dots \wedge a_n$. Atoms in rules can be of the form $C(x)$, $P(x,y)$, $Q(x,z)$, *sameAs*(x,y), *differentFrom*(x,y), or *builtIn*(*pred*, z_1, \dots, z_n), where C is an OWL description, P is an OWL individual-valued property, Q is an OWL data-valued property, *pred* is a datatype predicate URIref, x and y are either individual-valued variables or OWL individuals, and $z, z_1, \dots z_n$ are either data-valued variables or OWL data literals. An OWL data literal is either a typed literal or a plain literal [2]. Variables are indicated by using the standard convention of prefixing them with a question mark (e.g., $?x$). URI references (URIrefs) are used to identify ontology elements such as classes, individual-valued properties and data-valued properties. For instance, the following rule asserts that one's parents' brothers are one's uncles where parent, brother and uncle are all individual-valued properties.

$$\text{parent}(?x, ?p) \wedge \text{brother}(?p, ?u) \rightarrow \text{uncle}(?x, ?u) \quad (2)$$

3.3 SWRL Built-ins

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

- *Comparisons*
- *Math Built-Ins*
- *Built-Ins for Boolean Values*
- *Built-Ins for Strings, etc.*

The next SWRL rule is an example using the Math built-in “swrlb:greaterThan”. If the result of the built-in is true for a Person ?p then this Person ?p is a member of the of the concept Adult.

$$\text{Person(?p) \wedge hasAge(?p, ?age) \wedge swrlb:greaterThan(?age, 18) \rightarrow Adult(?p)} \quad (3)$$

4 Spatial components

This section discusses the spatial components within GIS technology and the database system. It is important to evaluate the spatial features within the existing technologies in order to take the advantage from their developments. Additionally, the spatial functions of database system are utilized to execute spatial rules within spatial built-ins.

Today most of database systems provide support to the spatial extension. This paper uses *PostGIS* a spatial extension *PostgreSQL* for the arguments but same could be applied in other database system too. PostGIS supports the storage of *point*, *line*, *polygon*, *multipoint*, *multiline*, *multipolygon*, and *geometrycollections*. It follows the specification provided by OGC for the *simple features* to store these objects. Those are specified in the Open GIS *Well Know Text (WKT)* or *Well Known Binary (WKB)* Formats. It stores 3Dimensional coordinates as *Extended Well Known Text (EWKT)* and *Extended Well Known Binary (EWKB)* – the extensions it defined. They are different from *Simple Feature Specification* by OGC as they embed *Spatial Reference Identifier (SRID)* within them. Besides providing functionalities for storing the geometries and exporting/importing geometries from/into the database, *PostgreSQL* with its spatial extension *PostGIS* provides a range of spatial functions which are spatial relationship functions and spatial processing functions.

- The **spatial relationship functions** are generally binary functions. These functions return a Boolean value. However, when they are used with a proper SQL statement, these functions can be used to identify the objects with which they are related to. The functions are used as SQL statement. The examples of spatial functions under this category are *touch*, *disjoint*, *overlap*, *within* and are used through *st_touch*, *st_disjoint*, *st_overlap*, *st_within* respectively in PostGIS.
- The **spatial processing functions** provided in this section allow the processing of the object geometries. The results themselves are sets of geometries. The spatial built-ins *Buffer* and *Intersection* discussed in (1) belong to this category. Besides *buffer* and *intersection* there are functions like *Difference*, *Union* under this category. Those functions are executed through *st_buffer*, *st_intersection*, *st_difference* and *st_union* respectively in PostGIS.

5 Ontological Adjustment

The adjustment consists in enriching [8] an existing ontology that describes a specific domain. This paper uses the domain ontology described in ArchaeoKM [5], [6], [7] and consists in adding new axioms (concepts, relations, attributes, etc.) for our purpose. Once the Spatial SWRL rules are executed, the results of these rules will generate information that have to be stored in the enriched part of the ontology. The main process of enriching the ontology schema consists in adding the concept *feat:siteFeature*. All the objects, that define a domain concept and have a geometrical definition in the spatial database, requires to be instances of the concept *feat:siteFeature*. This concept is important as it allows the definition of links between the adjusted domain ontology and the spatial functions. These spatial analysis properties are specializations the relationship *sa:hasSpatialRelAnalysis*. The concept *sa:spatialAnalysis* refers to the spatial functions as its specialized concepts and are defined through its inheritance. In addition, the links between the ontology and the database are defined using the link *feat:hasAnnotation*. The *shape:feature* relates to the geometrical definition of excavated objects and the *an:tag* refers to the same geometrical definition but stored into the database. Details on how *an:tag* or *feat:Annotation* functions can be read in [5], [6], [7].

5.1 Spatial relationship functions

The following four sub-relation of the relationship *sa:hasSpatialRelAnalysis* define spatial relationships between two objects. The result of a spatial function process between two objects of the kind of the concept *feat:siteFeature* can be a new link between them. This new link is of kind of e.g. Table 1.

Table 1. Ontology adjustment concerning the Spatial Relation Functions

Spatial Relationship Functions	ObjectProperties
Disjoint	<i>sa:hasDisjoint(x,y)</i>
Touches	<i>sa:hasTouch(x,y)</i>
Within	<i>sa:hasWithin(x,y)</i>
Overlaps	<i>sa:hasOverlaps(x,y)</i>

The variables *x* and *y* are of the type of the concept *feat:siteFeature*. It means that it could be an object or the result of a spatial processing function.

5.2 Spatial processing functions

The four spatial processing functions are Buffer, Union, Intersection and Difference. Contrary to the spatial relationship functions, they compute new spatial geometries. These new geometries are also stored in the spatial database in order to be computed

by future spatial functions. As a solution, we definition four new concepts called *feat:sp_buffer*, *feat:sp_union*, *feat:sp_Intersection* and *feat:sp_difference* which are of kind of *feat:siteFeature*. By inheritance, these four concepts have a spatial definition in the spatial database which are defined with the help of the relationship *feat:hasAnnotation* like any other finding objects. There is also four *sa:hasSpatialRelAnalysis* defined corresponding to each spatial processing function (*sa:hasBuffer*, *sa:hasUnion*, *sa:hasIntersection*, *sa:hasDifference*). They are used to keep a link between the first spatial geometry of the spatial function and the results of this spatial function (e.g. Table 2).

Table 2. Ontology adjustment concerning the Spatial Processing Functions

Spatial Processing Functions	Concept	Object Property
Buffer	feat:sp_Buffer	sa:hasBuffer(x,y)
Union	feat:sp_Union	sa:hasUnion(x,y)
Intersection	feat:sp_Intersection	sa:hasIntersection(x,y)
Difference	feat:sp_Difference	sa:hasDifference(x,y)

The variables x and y are of the type of the concept “feat:siteFeature”. It means that it could be an object or the result of a spatial processing function.

6 Definition of the Spatial SWRL Built-ins

At this point, the ontology adjustment is defined. From this adjustment, the Spatial SWRL Built-ins can be defined for each spatial function. Before the definition of these Built-ins, it is necessary first to explain how work the engine that translates Spatial SWRL rules into standard SWRL rules. The example (15) uses five axioms. The axioms River and Building is of the kind of the concept “feat:siteFeature”. It means that they have both a spatial geometry stored in the database. The axiom “isLiableToFloodingBy” is a relationship that links two object of the kind of the concept “feat:siteFeature”. It means that a building “?y” can be liable to flooding by a river “?x” if all the axioms of the antecedent are true. This rule is computed for every rivers and buildings that are present in the ontology. The axiom “spatialswrlb:Buffer” is to compute a buffer for the feature “?x”, and the axiom “saptialswrlb:Intersection” is used to compute the intersection of the second feature “?y” with the result of the buffer operation. If there is a result “?res” of the intersection function, then a new relation is created.

The role of the translation engine consists in

1. *interpreting the Spatial SWRL rules*
2. *computing the spatial functions within spatial database*
3. *updating the ontology and the spatial database with the results of the spatial functions*

4. translating the spatial SWRL rules into standard SWRL rules
5. running the rules with the help of a standard rule engine as Racer, Jess or Pellet

The two next sections explain how the spatial built-ins are translated into SWRL rules. The computing of the spatial functions is out of the scope of this paper. However, it uses SQL statements.

6.1 Spatial Relationship Built-ins

Concerning these built-ins, the translation engine computes the spatial function in the database within all the instances of the built-in parameters. For instance, the built-in *spatialswrlb:Disjoint(?x, ?y)* is interpreted by the translation engine and compute all the instances of the kind of the variables *?x* and *?y*. If the result is true for any couple of instances, then a new relationship *sa:hasDisjoint* is created in the ontology between the couple of instances. After what, the axiom *spatialswrlb:Disjoint(?x, ?y)* is replace in the rule by the axiom *sa:hasDisjoint(?x, ?y)*. Consequently, the rule is now a standard rule.

6.1 Spatial Processing Built-ins

Concerning these built-ins, the translation is a bit more complex. Actually, the translation engine has to interpret the spatial built-ins and to compute the new geometry for each built-in. The resulting geometries are stored in the spatial database and a new individual of the kind of the *feat:sp_Buffer*, for example, is created in order to keep a link with the database. In addition, a link of the kind of the relationship *sa:hasBuffer*, for example, is created in order to keep a relationship between the first individual parameter of the built-in and the new individual *feat:sp_Buffer*. Once the ontology is updated, the axiom *spatialswrlb:Buffer(?x, ?value, ?res)*, for instance, is replace by the following two axioms *sa:hasBuffer(?x, ?res) ^ feat:sp_Buffer(?res, bufDistance(?value))*. The parameter *?res* is to refer the resultant instances of *feat:sp_Buffer*, for instance. Similarly *bufDistance(?value)* defines the buffering distance. It is a data property but is important factor defining a buffer zone. Due to a lack of space, the complete translation table is not given.

The example (1) is a Spatial SWRL rule and the example (5) is its translation into a standard SWRL rule done by the translation engine. Meanwhile, the translation engine has computed the necessary geometries and has updated the domain ontology with individuals and relationships allowing the run of the translated rule by a reasoning engine. Thus, a spatial reasoning is done on the domain ontology.

$$\begin{aligned} & \text{River}(?x) \wedge \text{Building}(?y) \wedge \text{sa:hasBuffer}(?x, ?z) \wedge \text{feat:sp_Buffer}(?z) \wedge \\ & \text{sa:hasIntersection}(?z, ?res) \wedge \text{sa:hasIntersection}(?y, ?res) \wedge \\ & \text{feat:sp_Intersection}(?res) \rightarrow \text{isLiableToFloodingBy} (?y, ?x) \end{aligned} \quad (4)$$

7 Conclusion

This has presented the integration of the spatial functions into domain ontology via its adjustment. The ideas presented here could contribute to the development of analysis solution for the GIS technology. The combination of a rule language with spatial functions will add a new dimension in which users interpret their views. A layer in between the data layer and the visualization layer could be added in the existing GIS system which performs the ontological operations. This layer will act as the facilitating tool for the spatial knowledge base in the current system. The integration of such layer in the existing GIS system will provide a firm base by providing much needed dynamism to the system.

References

1. W3C, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, <http://www.w3.org/Submission/SWRL/>, Last Visited: 25 November (2008)
2. Pan, J.Z., Horrocks, I.: OWL-Eu: Adding Customised Datatypes into OWL, In Proc. of Second European Semantic Web Conference (ESWC 2005) (2005)
3. Smith, M.J., Goodchild, M.J., Longley, P.A.: Geospatial Analysis: A Comprehensive guide to Principles, Techniques and Software Tools. Metador. (2007)
4. Bechhofer, S., Harmelen, F.V., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider P.F., et al.: OWL Web Ontology Language. Retrieved November 27, from W3C Recommendation: <http://www.w3.org/TR/owl-ref/> (2009)
5. Karmacharya, A., Cruz, C., Boochs, F., Marzani, F.: Support Of Spatial Analysis Through A Knowledgebase - A New Concept To Exploit Spatial Information Shown For Industrial Archaeology, The International 24th Cartographic Conference, Chili, 16-21 November (2009)
6. Karmacharya, A., Cruz, C., Boochs, F., Marzani, F.: ArchaeoKM: toward a better archaeological spatial datasets management, Computer Applications and Quantitative Methods in Archaeology (CAA), Williamsburg, Virginia, USA (2009)
7. Karmacharya, A., Cruz, C., Boochs, F., Marzani, F.: Managing Knowledge for Spatial Data – A Case Study with Industrial Archaeological Findings, Digital Heritage in the new knowledge environment: shared spaces & open paths to cultural content, Athenes, Grece, (2008)
8. Cruz, C., Nicolle, C.: Ontology Enrichment and Automatic Population From XML Data, 4th ODBIS Workshop on Ontologies-based Techniques for DataBases in Information Systems and Knowledge Systems, Co-located with VLDB - August, 23rd (2008)
9. Boley, H.: The Rule Markup Initiative, <http://ruleml.org/>, Last Visited 11 February (2010)
10. Lieberman, J., Singh, R., Goad, C.: W3C Geospatial Ontologies – W3C Incubator Group Report”, W3C, <http://www.w3.org/2005/Incubator/geo/XGR-geo-ont-20071023/> (2009)
11. PostgreSQL, PostGIS Manual, PostgreSQL documentation (2008)

Appendix: Additional figures

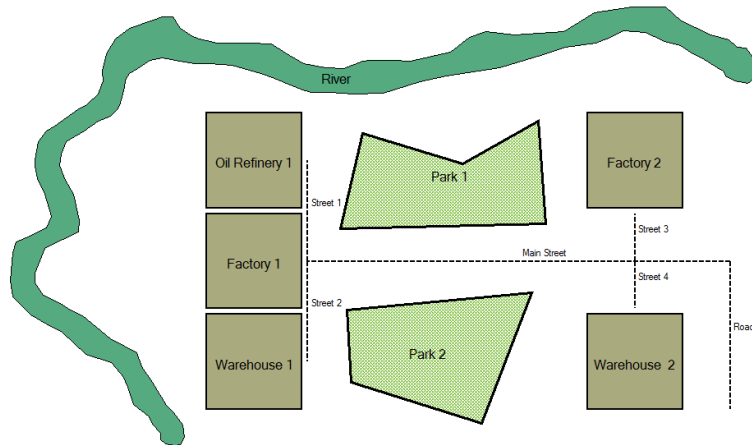


Fig. 1. This figure is a representation of the findings concerning an archaeological site which is composed of a river, different buildings, two parks. For this, the GIS Quantum user graphical interface is used.

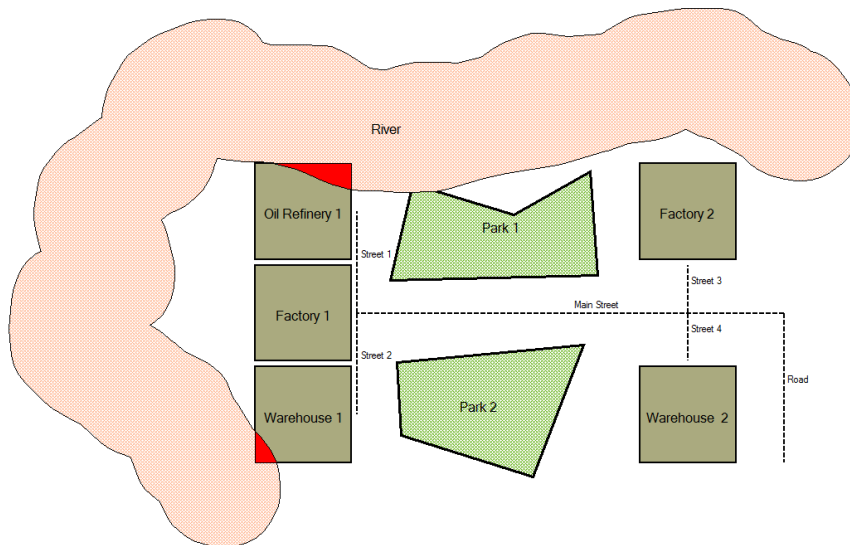


Fig. 2. This figure is the result of the spatial functions described in the example (1). Actually the building “Oil Refinery 1” and the building “Warehouse 1” are liable to flooding by the river “River”.