Semantic Description of Location Based Web Services Using an Extensible Location Ontology

Rob Lemmens[1], Marian de Vries[2]

[1]Department of Geoinformation Processing
International Institute for Geo-Information Science
and Earth Observation (ITC) Enschede
lemmens@itc.nl

[2]OTB Research Institute for Housing, Urban and Mobility Studies
Delft University of Technology (TU Delft)
m.e.devries@geo.tudelft.nl

Abstract

A growing number of Web services, including Location Based Services (LBS) is becoming available to the public, but it is difficult to find them and to judge whether they could be used in combination with other services. This is partly caused by the fact that conventional service descriptions fall short in capturing the semantics of services. In this paper we present alternative ways to enrich service descriptions with semantic information, focussing on the domain of LBS. These descriptions refer to concepts in an extensible location ontology that is expressed in the Web Ontology Language (OWL). We demonstrate the adequacy of a reasoner to perform matchmaking between the descriptions of a required service and advertised services. Despite the current limitations, the presented method allows for much more expressive descriptions than WSDL and can be used to semantically enable OpenGIS GetCapabilities in discovery and service chaining mechanisms.

Introduction

There is an increasing number of Location Based Services available on the Web for the general public. However it is often not easy for a mobile user to find the right service, i.e. with the right functionality and possibilities for his or her specific purpose. In order for Location Based Services to be discovered (by humans or by intelligent 'search' agents or brokers) they will have to be described in a way that makes reasoning (matching of user requirements with service capabilities) possible.

A Location Based Service can be characterized in a number of ways: by name, 'free text' description or service classification, but also by specifying input(s) and output(s) of the service.

In the research presented in this paper we take the last approach. We will show how input and output of Web service operations can be described with reference to concepts in an ontology. In this paper we present a (still limited) location ontology, where concepts like 'address', 'feature type' and 'area' are defined. As language to construct the ontology we use the OWL (Web Ontology Language) DL subset (Dean and Schreiber 2004).

GEO WEB SERVICES DESCRIBED SEMANTICALLY

The discovery of the ‘right’ Web service for a specific purpose basically involves the matching of the properties of the required service with the properties of advertised services. In many cases services already carry standardised labels. For example, ‘OGC WMS’ is a known service type name, agreed in text-documented OGC specifications. One may assume that two services with equal labels do match by implicit agreement (but only as far as the specifications, indicated with that label, reach; possibly as far as the implicit references to other specifications). In contrast, the match between (1) a validity region ‘ITALY’ as property of a *requested* service and (2) a country polygon-geometry of an *advertised* service cannot be compared by name but will involve a spatial comparison mechanism of some kind. In line with this discussion Lutz et al. (2003) distinguishes implicit and explicit semantics in metadata.

We argue that web based ontologies can play an important role in service discovery by overcoming the limitations of agreements by textual specifications (deployed in many of today’s standards). The power of web based ontologies lies in the provision of links to alternative comparison mechanisms (e.g. gazetteers, in case of validity regions), their interoperable (XML based) representations, the use of unique namespaces and the fact that they allow for automated reasoning. The niche for web based ontologies typically is not in the area of data formats (which are covered by many standards) but rather at the level of conceptual data models where current standards fall short.

The representation of conceptual data and process models has seen a recent technology push by the Semantic Web. An example is OWL-S[15]: an extension of OWL with constructs to describe the semantics of Web services and their operations more adequately than is possible with the combination of UDDI and WSDL.

[15] http://www.daml.org/services/owl-s/1.0/

In a service discovery process, ontology based descriptions are used by reasoners that basically perform a matchmaking by inferencing between the concepts in a *requested* description and the concepts in an *advertised* description. There is a distinction between concept reasoning (Tbox reasoning) and inferencing with concept instances (Abox reasoning) (Baader et al. 2003). Instance querying can be done through the OWL-QL query language (Fikes et al. 2003).

In a distributed heterogeneous environment such as the Web we expect an organic development of ontologies in which each ontology developer structures his own concepts. If we want to make our Web resources, including Web services, semantically interoperable in such environment, we need to reference to multiple interoperable ontologies through mappings and reuse (Wache et al. 2001). Ontologies tend to have a specific scope and likewise determine the content of the descriptions that reference to them.

LOCATION BASED WEB SERVICES, A CHARACTERIZATION

An important interoperability initiative in the field of LBS is the OpenLS specification of the OpenGIS Consortium (OGC 2004). In this specification five Core LBS service types are defined together with the request and response parameters for each service type.

The OpenLS specification is an effort to standardize the interfaces between the various types of LBS services and LBS clients. The text of the specification defines the operations that must be supported (and the ones that are optional), the input parameters and the media-types and other characteristics of the output of each operation. 'Input' and 'output' have to do with the syntax of the interface: the name-value pairs of parameters and/or the XML Schema structure of XML data streams that go from client to server and vice versa.

An important distinction to be made here is the difference between the interface of the operations (the generic side of an OpenLS service), and the specific properties of a particular (instance of an) OpenLS service. E.g. Presentation Service Y has both generic properties (as a consequence of it being an implementation of the OpenLS standard) and specific properties, which have to do with the content of the service: the features types, the layers, the spatial reference system, the bounding box, etc.

For pure 'syntactic' matching between requested and advertised services it would be enough to know that Presentation Service Y is an implementation of an OpenLS Presentation Service, but because of the specific data content of individual Web services also individual service descriptions are necessary.

As an illustration, we present a use case in which a mobile user travels from a start location S to destination D and wishes to know whether he will meet any obstacles or irregularities along his way. In an advanced scenario the user engages an agent service that takes the start location (if it is the current position then it could be read from a GPS) and the destination location and searches the Web for relevant 'obstacle' services and presents the best route. In a less advanced scenario, the mobile user is going to search for 'obstacle' services which are in the neighbourhood of S and D and visually deduces the best route on a map showing S, D and the obstacle(s). If a service is not able to act on the required data input, e.g. because it takes as input a coordinate pair instead of an address, than the user may need another service that provides a conversion (in this case for example the Placefinder service of ESRI[16]). A typical service chain would include the following services, depending on their I/O: Positioning - Geocoding - Bounding box calculation – Coordinate transformation – WMS or OpenLS Presentation Service that shows road network and obstacles for that area.

Location ontology

Our approach follows partially the OWL-S Profile design. For the common reference of locations we have created a location ontology in OWL and opted for a flexible setup in which we (1) reuse parts of the ISO 191** family of standards (ISO 2004) and (2) allow for plugging-in existing Web based ontologies or models such as feature type classifications of national mapping agencies (e.g. the Dutch Top10NL) and GML object geometries types that we expect to be modelled in the OWL family of languages soon. The scope of our ontology is limited to the aspects of geo-locations which are considered to be basic entities for Location Based Services. Appendix A and B (together forming one diagram) show the core part of our ontology, exposing the concepts that are essentially used for differentiating the operations' data inputs and outputs with service matchmaking as our primary goal. The diagram is a graphic representation of the OWL code in the Protégé software environment[17]. Central in this ontology is the *feature* concept. The feature concept can have a coordinate identifier (e.g. lat, lon) or a geographic identifier (e.g. a postal code number) as its location identifier. This is essential to distinguish between, for instance, LBSs that take either coordinates or addresses as inputs. Further, the ontology contains concepts that support specific foci

[16] http://arcweb.esri.com/arcwebonline/index.htm
[17] http://protege.stanford.edu/

with respect to the service matching process (see table 1). Example 1 (geocoding): In case we want to match the description of a service taking a full address and one supporting only postal code areas, then the ontology concept *Geo-id elements* is used in the matching. Example 2 (geometry): A service that expects a town[18] to be a polygon will not take a point as input. Example 3 (theme): The type of thematic queries depends on the attribute model of the data embedded in the service. If the matching focus is of such thematic character, we have to include the *feature type classification* concept in the matching process.

Tab. 1: Feature characterizations: the appearance of concepts, as part of the geodata ontology design, follows a specific service matching focus. In general, all the matching foci together form the *scope* of the ontology. This table shows the key matching foci of our ontology.

Feature characterization	**Ontology concept**	**Sub concept examples**	**Service matching focus**
Geocoding	Geo-id elements	Postal_code, Street_name, Town_name	(only for geographic identifiers) Type & Accuracy of the location identifier
	Not yet implemented		(only for coordinate identifiers) Type & Accuracy of the location identifier
Geometry	Object geometry type	GML_point, GML_linestring, GML_polygon	(only for coordinate identifiers) - Level of detail of feature - Operation capabilities
	Real world geometry type (conceptual geometry type, linked to the geocoding by the service)	Point, Line, Area	(only for geographic identifiers) Level of detail of feature
Theme	Feature type classification	Building, Road, Town	Meaning of data content

Obviously, the ISO 19115 (GI-metadata) standard was not designed for the representation of functional aspects of geo locations and was therefore not suitable to serve as our starting point. Parts of ISO 19111 (spatial ref-

[18] Only in case the thematic character is relevant (e.g. the essence of the feature being a *town* is a condition for the service match), the feature type classification is needed, otherwise the thematic type of the feature is generic.

erence by coordinates) and ISO 19112 (spatial reference by geographic identifiers) provided more useful constructs, and although they were not sufficient for our purpose, we borrowed some of their concepts.

In order to support the reuse of multiple ontologies, we have made our ontology extensible by providing empty connector concepts at specific places. The connectors are used to plugin name spaced concepts of external ontologies. One option to realize the actual connection is to assert the connector and the plug to be equivalent concepts in the same ontology (see figure 1). An alternative option is to create external mappings. In this way we are able to perform reasoning across multiple ontologies.

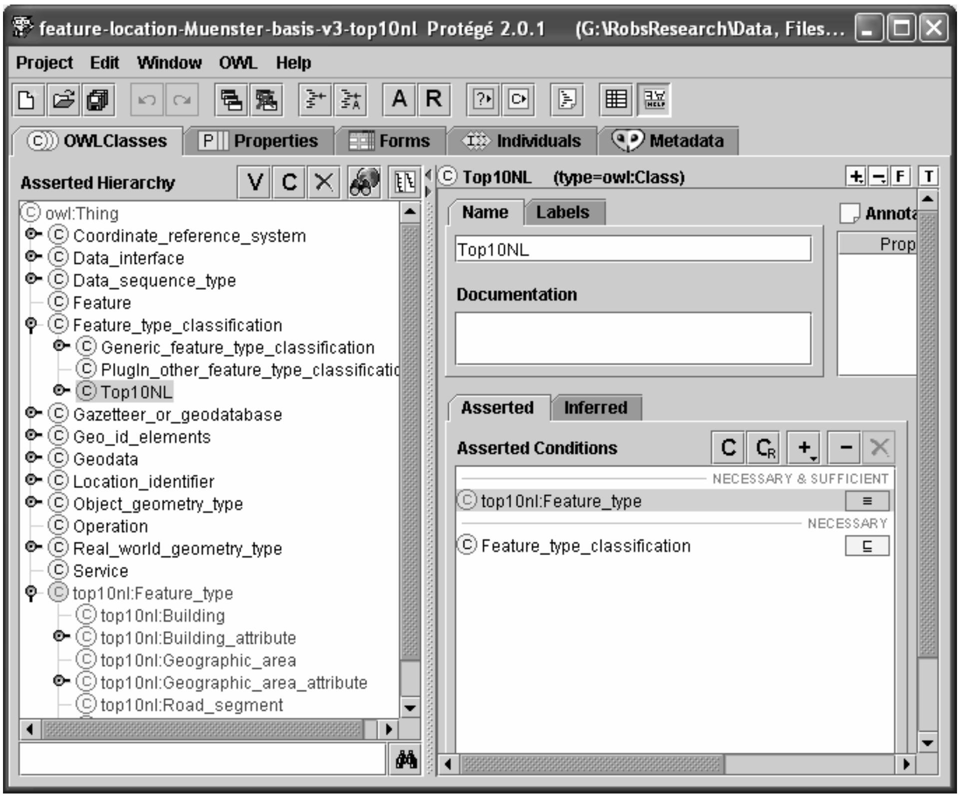


Fig. 1: Modularity of the ontology. In the asserted hierarchy a feature type ontology, based on the Dutch Topographic Service classification (Top10NL, see Knippers et al. 2002), has been plugged-in.

Further, as can be seen in appendix A, in our ontology a service is clearly distinguished from its operations and operation I/O. Despite the appearance as a concept, we have not yet implemented service type classifications at this stage (e.g. based on the service taxonomy in ISO 19119).

Ontology based descriptions

We will now elaborate on two alternatives to capture (parts) of the semantics of a service in DL statements. These statements directly translate into OWL-DL documents that can be used to enrich WSDL and OpenGIS GetCapabilities documents.

Modelling with concepts

The input/output of an operation is characterized by so-called atomic concept conditions, each pointing to specific concepts in the ontology, following a path from the top concept down into the ontology.

For example, to characterize our 'obstacle' WMS, the *advertisement* of its operation output is formalized by the concept A, containing atomic concept conditions A_1 through A_6 in the following (not complete) Description Logic (DL) statements[19]:

$A_1 \sqsubseteq Operation$

$A_2 \sqsubseteq \exists has\_output\ (\exists has\_feature\ (\exists refers\_to\_feature\_type\ Obstacle))$

$A_3 \sqsubseteq \exists has\_output\ (\exists has\_feature\ (\exists has\_location\_identifier\ (\exists has\_coordinate\_reference\_system\ Dutch\_CRS)))$

$A_4 \sqsubseteq \exists has\_output\ (\exists has\_feature\ (\exists has\_location\_identifier\ (\exists represented\_by\_object\_geometry\_type\ GML\_polygon)))$

$A_5 \sqsubseteq \exists has\_output\ (\exists has\_validity\_region\ (\exists has\_location\_identifier\ (has\_geo\_id\_elements \ni THE\_NETHERLANDS)))$

$A_6 \sqsubseteq \exists has\_data\_interface\ Map\_output$

$A \equiv A_1 \sqcap A_2 \sqcap A_3 \sqcap A_4 \sqcap A_5 \sqcap A_6$

The operation's input and the I/O of a *requested* service operation is captured in a similar way.

Accordingly, the input of the geocoding ESRI_placefinder Web service, as possible part of our service chain, is advertised as follows:

$A_1 \sqsubseteq Operation$

$A_2 \sqsubseteq \exists has\_input\ (\exists has\_feature\ (\exists refers\_to\_feature\_type\ Town))$

$A_3 \sqsubseteq \exists has\_input\ (\exists has\_feature\ (\exists has\_location\_identifier\ (\exists has\_geo\_id\_elements\ Town\_name)))$

$A_4 \sqsubseteq \exists has\_input\ (\exists has\_feature\ (\exists has\_location\_identifier\ (\exists represented\_by\_real\_world\_geometry\_type\ Single\_area)))$

$A_5 \sqsubseteq \exists has\_input\ (\exists has\_validity\_region\ (\exists has\_location\_identifier\ (has\_geo\_id\_elements \ni THE\_WORLD)))$

$A_6 \sqsubseteq \exists has\_data\_interface\ Manual\_text\_input$

$A \equiv A_1 \sqcap A_2 \sqcap A_3 \sqcap A_4 \sqcap A_5 \sqcap A_6$

---

[19] The WMS also provides the road network as a background; in order to limit the length of the paper this is left out in the given DL statements.

The output of the ESRI_placefinder Web service is characterized as follows:

$A_1 \sqsubseteq Operation$

$A_2 \sqsubseteq \exists has\_output\ (\exists has\_feature\ (\exists refers\_to\_feature\_type\ Town))$

$A_3 \sqsubseteq \exists has\_output\ (\exists has\_feature\ (\exists has\_location\_identifier\ (\exists has\_coordinate\_reference\_system\ Geographic)))$

$A_4 \sqsubseteq \exists has\_output\ (\exists has\_feature\ (\exists has\_location\_identifier\ (\exists represented\_by\_object\_geometry\_type\ GML\_point)))$

$A_5 \sqsubseteq \exists has\_output\ (\exists has\_validity\_region\ (\exists has\_location\_identifier\ (has\_geo\_id\_elements \ni THE\_WORLD)))$

$A_6 \sqsubseteq \exists has\_data\_interface\ Text\_output$

$A \equiv A_1 \sqcap A_2 \sqcap A_3 \sqcap A_4 \sqcap A_5 \sqcap A_6$

Based on this output[20] we can infer that in our LBS scenario this particular geocoder is suitable in case start and destination are ‘far’ apart (i.e. in separate towns, see figure 2a), but is useless in case both start and destination are within one town (see figure 2b).

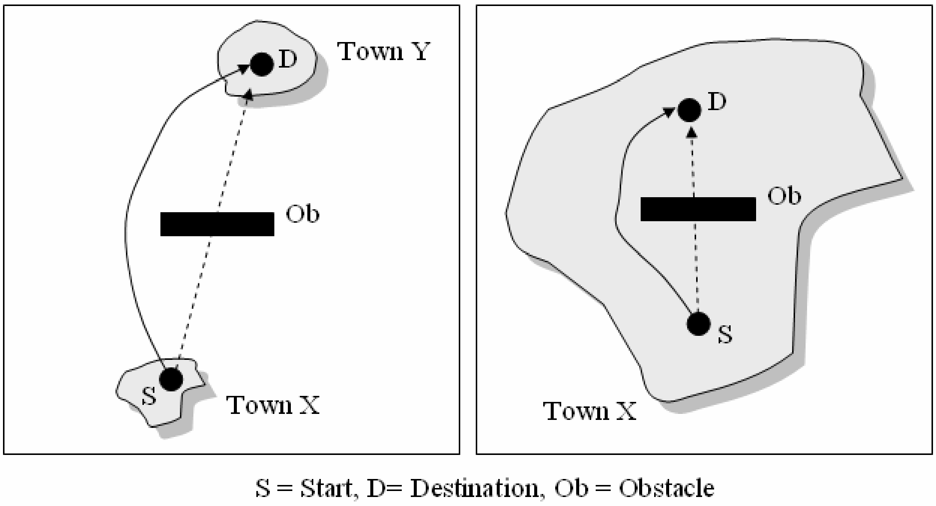


Fig. 2: Two LBS scenarios at a different scale: a. Routing locations are in different cities; b. Routing locations in one city.

All of the above DL statements are expressed in by OWL code in our software environment. Currently the OWL code for each service description is an integral part of the location ontology, but with a namespace mechanism it can easily persist in a distributed environment.

Modelling with individuals

[20] The feature type concept is in fact a service aspect that involves both input and output.

We distinguish between concept conditions that involve concepts only (A_1 to A_4 and A_6 in the above ESRI_placefinder description) and ones that involve statements about individuals (A_5). The latter type allows us, together with the actual individuals, to perform Abox reasoning. Figure 3 shows an example of the ontological model in such a case, similar to what Trastour (2001) has shown in an e-commerce setting. Note that the 'generic feature types' such as 'BUILDING' and 'TOWN' are now modelled as individuals, opposed to the original model as showed in the appendix. It is the task of the reasoner to infer whether the specific individual such as an *advertised* operation input is an instance of the *requested* input. In the figure, the 'Requested_input' concept is equivalent with the concept ' $\exists$*has_feature* ($\exists$*refers_to_feature_type Town)* '.

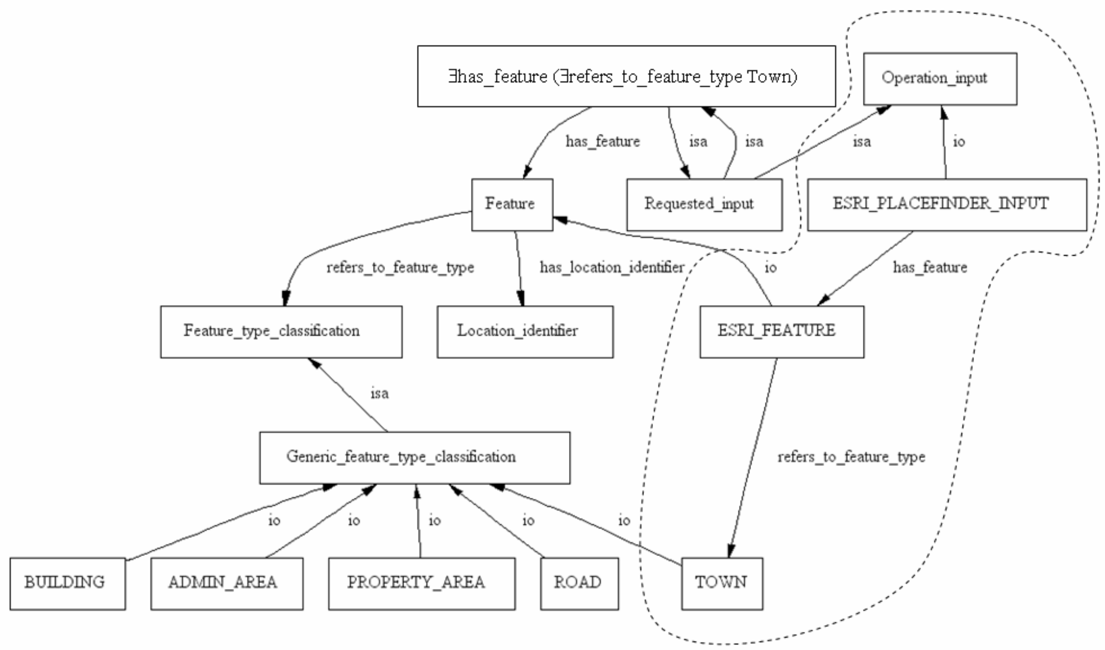


Fig. 3: A service modelled as an *aggregated* individual. The service's advertised operation input is actually represented by the interrelated instances, enclosed by the dashed boundary (only one service characterization is shown: feature type). All individuals are indicated with capitals and are related to a concept ('io' = instance of); 'isa' indicates a subsumption relation. The diagram is a graphic representation of the OWL code in the Protégé software environment[21].

The discussion on concept vs. individual modelling is both a fundamental and a practical one. Concept modelling allows us to define subsumption relationships, disjointness, etc. and will be always at the basis of individ-

[21] http://protege.stanford.edu/

ual modelling. An essential choice has to be made whether to define the leafs of the ontological tree as concepts or individuals. Concepts have the advantage of being able to comprise accurate conditions, referring to other concept definitions. In addition, concepts *have* to be used in case of connecting plug-in concepts. On the other hand, individuals provide an easier service description entry by a service provider or service requester and they allow us to fully deploy query languages such as OWL-QL. Obviously, the description and ontology design in terms of concept vs. individual modelling is also directly linked with the reasoning method. Some remarks are made in the next section.

FINDING THE RIGHT SERVICE

The alternative OWL models, as presented, have their pro and cons concerning the reasoning. In our tests we distinguish between the following reasoning methods: (1) using concept-only conditions, (2) using concept-instance conditions and (3) using concept-instance conditions plus actual individuals. In the last case we can capture either the advertised service or the requested service as an individual. In our test environment we have used the Protégé + Racer[22] classifier for TBox reasoning (cases 1 and 2) and Rice[23] + Racer for Abox reasoning (case 3).

Tbox reasoning (reasoning without the actual individuals) was performed through a re-classification by Racer and resulted in the identification of subsumption relations between the advertised service and the requested service description in case a correct match exists between the service descriptions. Similar successful results were reported by Klien et al. (2004) for the retrieval of geographic information in a heterogenic data environment. Further, the way we implemented our service descriptions even allows us to perform relaxed matching (represented by partial matches such as plug-in, subsume and intersection (see Lemmens et al. 2004).

ABox reasoning (with the actual individuals) performed equally well. The principle of ABox reasoning can be seen in figure 3: an ABox reasoner can infer from this ontological structure that ‘ESRI_PLACEFINDER_INPUT’ is an instance of the concept ‘Requested_input’. The result is depicted in figure 4.

[22] http://www.sts.tu-harburg.de/~r.f.moeller/racer/
[23] http://www.b1g-systems.com/ronald/rice/

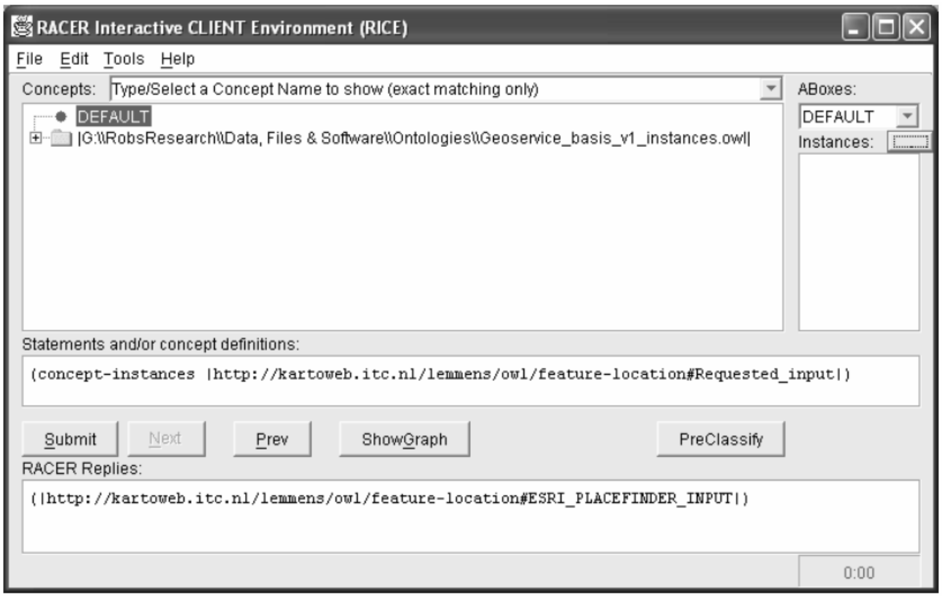


Fig. 4: On request the Racer reasoner finds the ESRI_PLACEFINDER _INPUT as an instance of the requested input. This input is modelled as a conditioned concept in the location ontology.

We found that both Tbox and Abox reasoning provide a powerful core for a discovery mechanism due to their exploitation of the semantic relations between concepts defined in an ontology and the possibility of relaxed matching. The description methods used are flexible and seem adequate to capture a wide range of Location Based Services.

The description of Web services with OWL DL (with emphasis on their input and output) makes discovery of these services easier and less error prone (less 'false hits', 'missed hits'). This is a result of the fact that the OWL DL atomic concepts can serve as input for reasoners like Racer.

Benefits of the OWL DL approach are:

- reasoning based on subsumption: if a user looks for a service with validity_region ‘FRANCE’, a service with validity_region ‘EUROPE’ is also OK.
- reasoning based on a combination of properties of services (= aggregate individuals)

The proposed way to describe Location Based Services (with OWL DL atomic concept conditions) is especially geared to the description ('publish') and discovery ('find') of services.

For the invocation of services ('bind') a combination with either WSDL or GetCapabilities still seems necessary. These provide the access points (URL's of the operations), and other 'syntactic' information about the interface. Further research will look at ways how to 'synchronize' these various levels of service descriptions.

An important limitation of the current method lies in the fact that we only model the data input and data output of a service. Modelling service *functionality* through service type classifications and process models seems to be the next logical step. In addition, we have not embarked yet on designing an end user friendly interface for creating descriptions and their reasoning based discovery. This is planned to be done with the Jena[24] OWL API in further work.

CONCLUSIONS AND FUTURE WORK

OWL service descriptions provide powerful elements for a discovery mechanism because they use a common information model reference (the ontology) in two ways: during the description process and the discovery. Although the present scope of our example ontology (locations) is limited, the extensibility of the ontology environment with external data models is an important asset for flexible service description. The proposed Description Logic based method is in principle generic. It allows for much more expressive descriptions than WSDL and can be used to semantically enable OpenGIS GetCapabilities in efficient automatic discovery mechanisms. In this paper we have discussed ontology modelling alternatives from the design and reasoning perspective. We expect our method to be applicable in environments where services are not known to be fully interoperable, because (1) they do not adhere to a common standard such as OpenGIS or (2) the standard does not cover certain service aspects, important for the matchmaking effort, such as some of the -in this paper indicated- locational, thematic and scale issues.

Future work (marking the limitation of the current methods) will focus on possible links with the OWL-S Process model, the development of a front-end for service providers and composers and the design of an efficient ontology infrastructure, providing appropriate mapping mechanisms.

---

[24] http://jena.sourceforge.net/

REFERENCES

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

Dean, M., Schreiber, G. (2004): *OWL Web Ontology Language Reference* ,W3C Recommendation 10 February 2004. Available at: http://www.w3.org/TR/owl-ref/.

Fikes, R., Hayes, P., Horrocks, I. (2003): OWL-QL - *A Language for Deductive Query Answering on the Semantic Web.* Knowledge Systems Laboratory Report KSL-03-14, Stanford University, Stanford, CA, 2003.

ISO (2004): ISO/TC 211 geographic information series of standards. Available at: http://www.isotc211.org/.

Klien, E., Einspanier, U., Lutz, M., Hübner, S. (2004): *An Architecture for Ontology-Based Discovery and Retrieval of Geographic Information*. In: Proceedings 7th Conference on Geographic Information Science, AGILE 2004, Heraklion, Greece.

Knippers, R., Kraak, M. (2002): *Top10vector Object-oriented. Design Data-model* version 1.1.2. April 2002. Available at http://kartoweb.itc.nl/top10nl/TOP10NL_eng/download/pdf/Design.pdf.

Lemmens, R., Arenas, H. (2004): *Semantic Matchmaking in Geo Service Chains: Reasoning with a Location Ontology.* 1st International Workshop on Geographic Information Management (GIM 2004), http://gim04.unizar.es, September 2-3, 2004, Zaragoza, Spain. To be published.

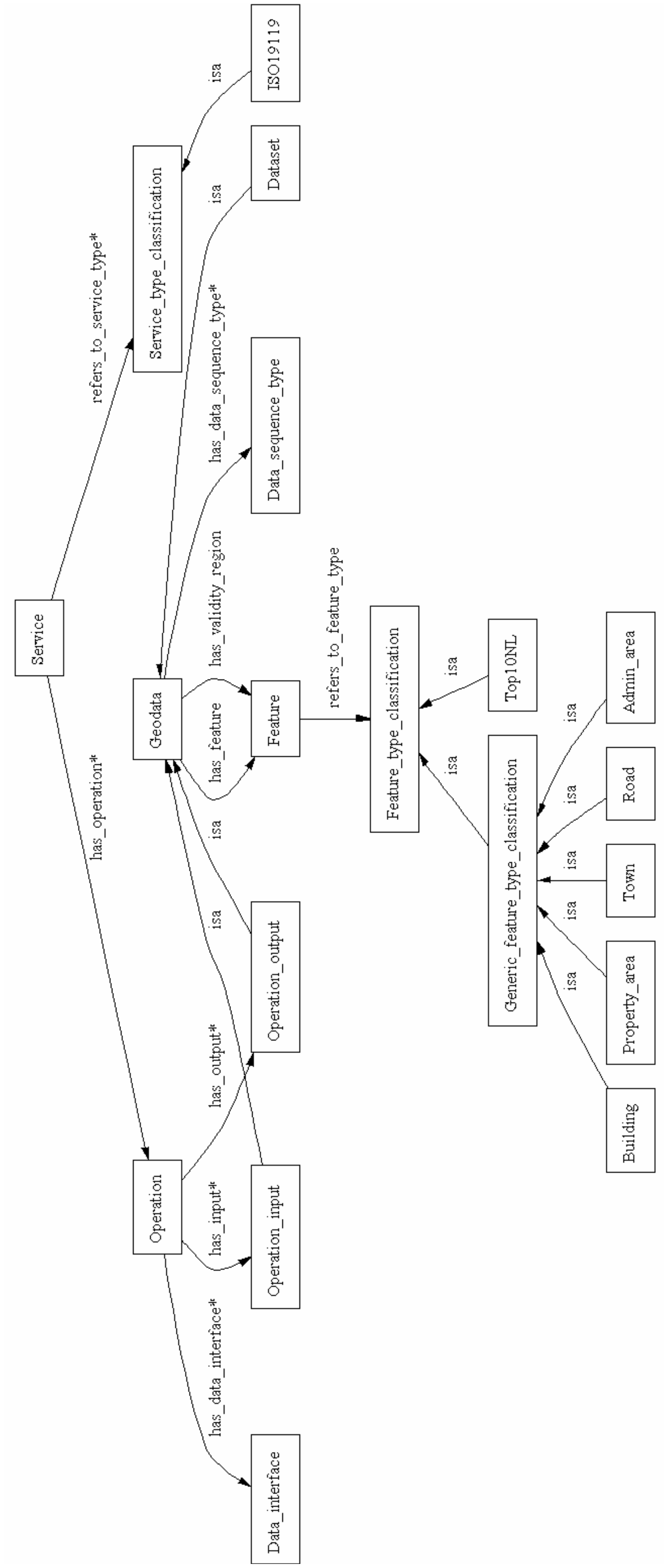
Lutz, M., Riedemann, C., Probst, F. (2003): *A Classification Framework for Approaches to Achieving Semantic Interoperability Between GI Web Services.* In: Conference on Spatial Information Theory (COSIT), 24-28 September, 2003, Ittingen, Switzerland.

OGC (2004). *OpenGIS Location Services (OpenLS) specification: Core Services [Parts 1-5] (OLS Core)* Version 1.0 document 03-006r3, January 2004.

Trastour, D., Bartolini, C., Gonzalez-Castillo, J. (2001): *A Semantic Web Approach to Service Description for Matchmaking of Services.* In: Proceedings of SWWS'01, The first Semantic Web Working Symposium, Stanford University, California, USA, July 30 - August 1: 447-461

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S. (2001): *Ontology-based integration of information - a survey of existing approaches.* In Stuckenschmidt, H., ed., IJCAI-01 Workshop: Ontologies and Information Sharing, 2001: 108-117.

Appendix A: Excerpt of Location ontology, used for TBox reasoning, part A. Appendices A and B link through the *feature* concept.



Appendix B: Excerpt of Location ontology, used for TBox reasoning, part B. Appendices A and B link through the *feature* concept.