Reasoning with spatial plans on the semantic web

Hoekstra, R.; Winkels, R.; Hupkes, E.

Publication date
2009

Document Version
Final published version

Published in
BNAIC 2009: Benelux Conference on Artificial Intelligence

Citation for published version (APA):
Reasoning with Spatial Plans
on the Semantic Web

Rinke Hoekstra\textsuperscript{ab} Radboud Winkels\textsuperscript{a} Erik Hupkes\textsuperscript{a}

\textsuperscript{a} Leibniz Center for Law, University of Amsterdam
\textsuperscript{b} AI Department, Vrije Universiteit Amsterdam

1 Abstract

Land use regulations are an important but often underrated legal domain. Especially in densely populated regions such as the Netherlands, spatial plans have a profound impact on both (local) governments and citizens alike. Given the prominence of maps in spatial plans, in order to improve access to these regulations a combination of existing technology for disclosing legal texts with that currently available in geographical information systems seems inevitable.

This paper presents an approach for specifying spatial norms using Semantic Web technology that enables an intuitive way of visualising their effects: map based legal case assessment. Users can see what is allowed and what not in specific areas on the map, they can represent a (simple) case by selecting or drawing an area on the map. Given a designation for that area, they can have the system assess whether this is allowed or not. The same solution also enables the comparison of two or more sets of spatial norms that govern the same region, e.g. coming from a municipality and the province it is part of. We demonstrate a practical use of the case assessment method specified in [3] using OWL 2 DL, and present a prototype system that provides a partial implementation of the approach. The system relies on two web services: a SPARQL endpoint, allowing the querying of our Sesame RDF repository, and a WFS service (Web Feature Service), that allows us to retrieve geospatial information from a GeoServer installation. GeoServer is specifically designed to store and reason on polygons, and supports the determination of certain spatial relations between features, allowing us to represent relations such as e.g. \texttt{la:overlaps}, and \texttt{la:next_to} as queries on the WFS service. The result of these queries can be added as relations to the RDF repository. Reasoning services are provided through the SwiftOWLIM inference layer on the Sesame repository.\textsuperscript{1} Results are shown on top of a map using the Google Maps API.

Arguably, the use of a restricted language as OWL DL may pose some problems for the representation of regulations. Firstly, certain aspects of legal reasoning may be hard or impossible to represent using a language that depends on monotonic reasoning. But our earlier work has shown that a significant portion of exceptions between norms, such as \textit{lex specialis}, can be dealt with without resorting to defeasible representations. A second problem is the complexity of the \textit{world} that is governed by law. To retain decidability, OWL 2 DL is restricted to models that have the ‘tree property’: situations that describe complex configurations of multiple objects can be only approximately defined in DL [1]. However, the domain itself is otherwise at a relatively high level of abstraction that excludes the complex structures found in domains such as biology and engineering, and complex spatial reasoning can be delegated to a geoserver.

The normative content of spatial plans is represented by specifying OWL descriptions of those situations, e.g. regions, that are allowed or disallowed by a spatial plan. Suppose we have two types of land use: \texttt{ex:Industry} and \texttt{ex:Nature}, both of which are subclasses of the general class of \texttt{la:Land Use}:

\begin{verbatim}
        ex:Industry ⊑ la:Land Use
        ex:Nature ⊑ la:Land Use
\end{verbatim}

\textsuperscript{1}SwiftOWLIM provides a limited subset of OWL DL inferences.
Correspondingly we define two regions, `ex:IndustryRegion` and `ex:NatureRegion`, both of which are subclasses of `la:Region`:

\[
\begin{align*}
ex:IndustryRegion & \equiv (la:land\_use \ some \ ex:Industry) \cap la:Region \\
ex:NatureRegion & \equiv (la:land\_use \ some \ ex:Nature) \cap la:Region
\end{align*}
\]

We map the corresponding land use categorisations from the Dutch national standard IMRO by assigning them as individuals belonging to the respective classes. This representation has a number of benefits, the most important being that our representation of land use is independent of the categorisation scheme adopted. For instance, we can map `gemet:industry` to `ex:Industry` in the same way, allowing us to specify norms on both IMRO and GEMET (European thesaurus for environmental information) encoded maps.

`norm:NoOverlapIN` is a norm that states that an overlap between a nature and an industry region is not allowed. In simplification of the approach presented in [3], this norm is represented as a subclass of `ex:IndustryRegion` with the restriction that the region overlaps with a region of type `ex:NatureRegion`. This norm is simultaneously defined as a subclass of `norm:ConflictRegion`:

\[
\begin{align*}
norm:NoOverlapIN & \equiv ex:IndustryRegion \cap la:overlaps \ some \ ex:NatureRegion \\
\subseteq norm:ConflictRegion
\end{align*}
\]

Should a user specify a region with intended land use ‘industry’ and the geoserver infer this overlaps with a region with land use ‘nature’ from a spatial plan, the OWL DL reasoner will infer that the user’s region is an `ex:IndustryRegion` and a `norm:NoOverlapIN`. Finally, the system will gather all individuals of the class `norm:ConflictRegion` using a simple SPARQL query:

\[
\text{SELECT ?region} \\
\text{WHERE { ?region rdf:type norm:ConflictRegion .}}
\]

Because `norm:NoOverlapIN` is a subclass of `norm:ConflictRegion`, the user’s region will be bound to the `?region` variable. The system will bring this fact to the attention of the user by highlighting his region on the map.

A drawback of the current representation is that since conflicts are represented at the class level, they cannot be queried at the instance level. For instance, this means that we currently cannot query the system for all regions that have a land use which excludes that of a hypothetical new region: the exclusion relations do not hold between the `la:Land\_Use` individuals directly, but are only inferred on the fly, for concrete situations. One option is to add the hypothetical region to the knowledge base, add `la:overlaps` relations to all existing regions, and retrieve the detected conflicts. This is not very efficient, to say the least. A second option is to explicitly assert actual `la:excludes_overlap` relations between the categories of land use in e.g. IMRO and GEMET. The main drawback of this approach is that it reintroduces a dependence on these schemas. Ideally, one would therefore like the exclusions between types of land use specified at the class level, to propagate to all instances of these classes. One way to achieve this is by introducing a complex combination of OWL 2 DL role chains, self restrictions and the universal property [2]. A similar approach has been described in [1] in the context of processes and actions, and we are currently investigating its use for the problems described here.

1.1 Acknowledgement


References

