RULES FOR GEOSPATIAL SEMANTIC WEB APPLICATIONS

Harry Chen, Stephane Fellah, Yaser Bishr
{harryc, stephanef, yaserb}@imagemattersllc.com
Image Matters LLC
Leesburg, VA 20175

I. INTRODUCTION

The use of geospatial data is ubiquitous in many real-world applications. For example, geospatial data is useful for planning bus routes for a city, for finding suitable helicopter landing areas for a military operation, and for analyzing the nuclear weapon production capability of a hostile country. In a typical geospatial application, the semantics of data (e.g., vector data, sensor data, and satellite images) and the procedures for processing the data (e.g., functions for coordinate transformation and image processing) are usually tightly coupled within the application implementations. This kind of "hard-wired" approach often hinders the flexibility and portability of the applications. With the emergence of semantic web languages (i.e., RDF/RDFS and OWL), we believe that future geospatial application will be able to exploit ontologies for geospatial knowledge sharing and computation.

RDF/RDFS and OWL allow the semantics of geospatial data to be explicitly defined using ontologies. However, the existing languages do not provide direct support for representing the semantics of the procedures for processing geospatial data (or rules).

The benefit of using rules in a geospatial application is twofold. First, rules can help to decouple the low-level implementation for processing geospatial data from the high-level computational logics that guide the execution of the low-level procedures. Second, when rules are used in conjunction with rule engines, they can help to enable different kinds of logical inference support in addition to OWL/RDF ontology reasoning – e.g., default reasoning and fuzzy reasoning.

In this paper, we describe the use of rules in geospatial semantic web applications and the implementation requirements for supporting rules. We assume that in geospatial semantic web applications, the semantics of all geospatial data will be explicitly defined by a set of shared ontologies. These ontologies will enable applications to acquire information from various sources and share it with each others.

II. AN EXAMPLE USE OF RULES

Let's consider a bus route planning application. When planning a bus route for a particular city, an analyst must determine the specific locations to place bus stops. In order to do so, the analyst must consider various kinds of geospatial data about the city, e.g., the traffic flow information (the directions in which different traffic flows), road networks (how roads are interconnected), historic traffic data (how roads are being used by the commuters), and "ridership" information (estimates of the number of bus riders at different locations).

Let's assume that all relevant geospatial data is expressed as ontologies using the OWL language. Because the bus route planning application understands the shared ontologies, it is able to acquire and process relevant geospatial data. However, in order to suggest the specific locations where the bus stops should be placed, rules are required. For example,

```
RULE1: IF distanceFrom(?locA, ?locB, ?dist) AND
          lessThan(?dist, 100, "meters") AND
          isTypeOf(?locA, fea:RoadIntersection) AND
          isTypeOf(?locB, fea:ShoppingMall)
       THEN
          busStopCandidate(?locA)

RULE2: IF busStopCandiate(?locA) AND
          existingBusStop(?locB) AND
          distanceFrom(?locA, ?locB, ?dist) AND
          lessThan(?dist, 700, "meters")
       THEN
          not(busStopCandidate(?locA))
```

In the above example, "RULE1" expresses that a road intersection is a good bus stop candidate if it is less than 100 meters away from a shopping center. "RULE2" expresses that a location is not a good bus stop candidate if it is too close (less than 700 meters) to an existing bus stop.

III. IMPLEMENTATION REQUIREMENTS

In order to integrate rules into geospatial semantic web applications, we must define a language for expressing rules and a framework for processing rules. We believe that the implementations should meet the following requirements:

1. The language should allow the expression of a predicate term with multiple arguments. For example, its common for geospatial relations to have more than two arguments (e.g, `distanceFrom(PointA, PointB, 5)` or `directionFrom(FeatureX, FeatureY, "north")`).

2. The framework for processing rules should support procedural attachments for built-ins. In geospatial applications, functions for computing various geometric attributes of a feature (e.g., a road or a building) are of great importance. Geospatial functions such as "locatedIn", "distanceFrom", "overlaps", and "contains" can only computed when the topology, geometry, and CRS (Coordinate Reference System) of the features are known. For example, the distance between two roads can only be calculated when the system knows the specific CRS that is used in describing the geometric coordinates of the roads.

3. The framework should support the serialization of rules. For example, if the underlying system uses Prolog to perform rule-based inference, the framework should allow applications to serialize rules from the Prolog native representation to the supported semantic web rule languages (e.g., SWRL), and vice versa. This requirement is essential because external rule-based inferences are usually defined in the native representation of the associated rule engines, as opposed to RDF.

4. The framework should provide a mechanism for detecting and resolving rule conflicts. In the previous example, it possible that the properties of a particular location satisfies both "RULE1" and "RULE2" – i.e., `busStopCandiate(someLocA)` and `not(busStopCandiate(someLocA))`.

5. The framework should define suitable interfaces for editing and managing rules. There are differences between editing ontologies and editing rules. For example, rules may be expressed using ontology languages such as OWL. However, the classic class-subclass UI view is inadequate for rendering rule structures and displaying the relations between different rule statements to a user.

IV. CONCLUDING REMARKS

Rules will play an important role in the future geospatial semantic web applications. An explicit representation of rule semantics can help to develop flexible and portable applications by decoupling the low-level geospatial procedures from the high-level computational logics that guide the execution of the low-level procedures.

At present, there is a number standards-proposing semantic web rule languages. Some languages are built on or extend the OWL language foundation, and some others are built on different kinds of logical languages. We believe that in order to effectively integrate rules into the future geospatial semantic web applications, semantic web rule languages should be grounded in the same language that is used for describing geospatial data. At present, we are exploring the use of SWRL in our applications.

In this paper, we have highlighted some requirements for building a rule-based framework for geospatial applications. We are currently prototyping a new geospatial semantic web application for intelligence analysis. As our research work progresses, we expect to gain additional insights on the use of semantic web rules in geospatial applications.