

# From ontology design to ontology implementation: A web tool for building geographic ontologies

Eleni Tomai Maria Spanaki  
PhD Students-National Technical University of Athens  
Zografos Campus, Athens, Greece  
etomai@central.ntua.gr, spanaki@mail.ntua.gr

## SUMMARY

*In this paper, we provide a web-based tool for building geographic ontologies. This work draws from a previously presented framework for building geographic ontologies utilizing Natural Language Understanding Semantics. Several special characteristics of geographic concepts have been tackled therein making it an appropriate framework for providing a generic ontology of geographic ontology design. The implementation process consists of two steps. The first step is to determine the generic schema of building a geographic ontology, while the second is to develop the interface where the users can build their own ontology based however on the predefined generic scheme. This tool has additional functions such as testing the ontology's consistency, assigning values to properties of geographic concepts and saving locally the ontology in OWL Full format. The current approach applies the specifications of W3C for ontology languages and tools.*

**KEYWORDS:** *Implementation of Geographic Ontologies, OWL, Web Tool*

## INTRODUCTION

Even though the term ontology has been widely used by the GIScience community for up a decade now, building a geographic ontology has not proven an easy task. The most crucial intricacy that has emerged is that no shared understanding of the underlined conceptualization has guided the intended specification. This absence of consensus among GIScientists originates in the difficulty of tackling all the special characteristics of the given domain, and in an indecisive view in terms of where domain semantics can be grasped and formalized from. In this setting, an attempt to provide useful guidelines for the generation of robust geographic ontologies by taking a linguistic view of the semantics of geographic space has been presented (Tomai & Kavouras, 2004). Based on this framework, we provide a web-based tool for building geographic ontologies conformant to the proposed requirements of ontology design. This tool enables the users to build their own geographic ontology, test its consistency, and assign values to properties, while providing additional functions explained later on.

## THE IMPLEMENTATION PROCESS

The steps for implementing this tool comprise of the following:

- Determine the generic geographic ontology design.
- Develop the user interface where the users can generate their own ontology based on the predefined scheme.

### **The Requirements of Ontology Design in an Ontology Language**

In this step, the framework proposed by Tomai & Kavouras (2004) was written in Ontology Web Language (OWL) (2004). A few lines about the language are in order at this point to justify its use.

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDFS) by providing additional vocabulary along with a formal semantics. The OWL Web Ontology

Language is intended to provide a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications.

The OWL language is used to:

- Formalize a domain by defining classes and properties of those classes,
- Define individuals and assert properties about them, and
- Reason about these classes and individuals to the degree permitted by the formal semantics of the OWL language (McGuinness & van Harmelen, 2004).

The OWL Web Ontology Language is a language for defining and instantiating Web ontologies. An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL Formal Semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.

### **OWL and XML Schema Standards**

One question that comes up when describing another XML/Web standard is why there is the need to use OWL language instead of the well known XML and XML Schema standards. As the W3C states at the OWL Web Ontology Language Guide (Smith, et al., 2004), there are two answers to this question.

- An ontology differs from an XML schema in that it is a knowledge representation, not a message format.
- One advantage of OWL ontologies will be the availability of tools that can reason about them. Tools will provide generic support that is not specific to the particular subject domain, which would be the case if one were to build a system to reason about a specific industry-standard XML schema.

### **The Species of OWL**

The OWL language provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

- OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. Owl Lite has a lower formal complexity than other type of OWL.
- OWL DL supports those users who want the maximum expressiveness while retaining computational completeness and decidability. OWL DL is so named due to its correspondence with Description Logics, a field of research that has studied the logics that form the formal foundation of OWL.
- OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary.

The choice between OWL Lite and OWL DL depends on the extent to which users require the more expressive restriction constructs provided by OWL DL. The choice between OWL DL and OWL Full mainly depends on the extent to which users require the meta-modeling facilities of RDF Schema (i.e. defining classes of classes). When using OWL Full as compared to OWL DL, reasoning support is less predictable. In the present methodology OWL Full was chose over OWL DL for making use of its special characteristics.

## IMPLEMENTING THE “GEOGRAPHIC ONTOLOGY REQUIREMENTS” ONTOLOGY

We made use of the Protégé OWL plug-in (2004), for implementing the guidelines for building a geographic ontology (figure 1 illustrates an excerpt of the OWL code<sup>1</sup>, while figure 2 portrays the visualization of the ontology<sup>2</sup>). The result is an “ontology” of the requirements for the design of a geographic ontology.

In what follows we explain the steps of the ontology.

- All elements of a Geographic ontology were conceptualized (i.e., *Geographic Concept*, *Relation*, *Axiom* and *Lexicon*) as classes.
- The class *Geographic Concept* was further distinguished into two subclasses the *Manmade Geo-Concept* and the *Natural Geo-Concept*.
- Then in the *Relation* class two subclasses were created, that is, *Semantic Relation* and *Semantic Property*. Under the *Semantic Relation* class the subclasses of *Taxonomy*, *Synonymy* and *Partonymy* were created. The *Taxonomy* class was further divided into *Hyponymy* and *Hypernymy*, while *Partonymy* into *Holonymy* and *Meronymy*. On the other hand, under the class *Semantic Property*, the subclasses of *Spatiality*, *Temporality*, *Activity*, *Nature*, *Purpose*, and *Material/cover* were created. Some of the semantic properties were enriched with subclasses; subsequently we ended up with a total of 20 classes of semantic properties.
- In addition, the *Semantic/syntactic Structure* class was created. Due to the fact that all semantic relations and properties are depicted, in our approach, from linguistic patterns, such as the syntactic structure of a sentence that describes a geographic concept, this class is essential in the ontology design process, and was therefore, introduced. The diverse nature of the properties and relations, as well as, the richness of Natural Language in denoting them, obliged us to provide the *Semantic-Syntactic Structure* class with 17 distinct subclasses. Examples of those are: *Activity denoting*, *Location denoting*, *Purpose denoting* etc, each one corresponding to a specific semantic relation or property. What is more, in order for the geographic ontology to be robust, we decided to include the syntactic and semantic structures that help us identify semantic properties and relations in the requirements. For that reason, we used the OWL possibility of creating enumerated classes that is provide their members, so that no other individuals can be declared to belong to the specific class<sup>3</sup>. To give an example, the *Topology denoting* class is enumerated with individuals (instances) such as *Adjacent to*, *Connected to*, *Separated from ... by*, and *Surrounded by*.
- Afterwards, ten object properties were created. These are: *Constrains*, *Describes*, *Documents*, *Entails*, *Involves*, *Has duration*, *Has element*, *Is antonym of*, *Is lexicalized by*, and *Takes value from*, each of which is explained hereinafter.
  1. The *Constrains* property was assigned to *Axiom* class with the restriction to be applied for the classes *Geographic concept* and *Relation*.
  2. The *Describes* property was assigned to class *Relation* and *Semantic property* and restricted to class *Geographic concept*.
  3. The *Documents* property was assigned *Lexicon* for the classes *Geographic concept*, *Axiom*, and *Relation*.
  4. The *Entails* property was assigned to several instances of *Purpose denoting* and restricted to *Manmade geo-concept*, subclass of *Geographic concept*.

---

<sup>1</sup> The complete code will be available at: <http://ontogeo.ntua.gr/>

<sup>2</sup> For reasons of readability, the visualization of the ontology can also be found in the previous web page.

<sup>3</sup> For the time being, a small amount of instances (patterns of semantic/ syntactic structures) have been included as members of the classes. The list is far from exhaustive nevertheless indicative of what elements in language help us identify properties of geographic concepts.

5. The *Involves* property was assigned to *Activity denoting* subclass of *Semantic-syntactic structure* and restricted to Semantic property *Role*.
6. The *Has duration* property was assigned to *Process*, subclass of semantic property *Activity*.
7. The *Has element* property was assigned to Geographic Ontology over the classes, *Lexicon*, *Geographic concept*, *Axiom*, and *Relation*.
8. The *Is antonym of* property was assigned to Natural and Artificial subclasses of Nature denoting subclass of *Semantic-syntactic structure*.
9. The *Is lexicalized by* property was assigned to several semantic properties and relations such as *Topology*, *Agent*, *Location*, *Hyponymy*, *Hypernymy*, *Process*, *Transition*, and others.
10. The *Takes value from* property was assigned to all other semantic properties and relations, to which the *Is lexicalized by* property could not been assigned, such as *Nature*, *Size*, *Form/Morphology*, *Shape*, *Time*, *Condition/Status*.

```

</owl:Class>
<owl:Class rdf:ID="GeographicOntology">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty>
<owl:ObjectProperty rdf:ID="HasElement"/>
</owl:onProperty>
<owl:allValuesFrom>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="file:/C:/Program%20Files/Protege_2.1/2211#GeographicConcept"/>
<owl:Class rdf:about="file:/C:/Program%20Files/Protege_2.1/2211#Relation"/>
<owl:Class rdf:ID="Lexicon"/>
<owl:Class rdf:about="file:/C:/Program%20Files/Protege_2.1/2211#Axiom"/>
</owl:unionOf>
</owl:Class>
</owl:allValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="TopologyDenoting">
<rdfs:subClassOf>
<owl:Class>
<owl:oneOf rdf:parseType="Collection">
<TopologyDenoting rdf:ID="adjacent_to"/>
<TopologyDenoting rdf:ID="connected_to"/>
<TopologyDenoting rdf:ID="seperated_from..._by"/>
<TopologyDenoting rdf:ID="surrounded_by"/>
</owl:oneOf>
</owl:Class>

```

**Figure 1:** Excerpt of the OWL code

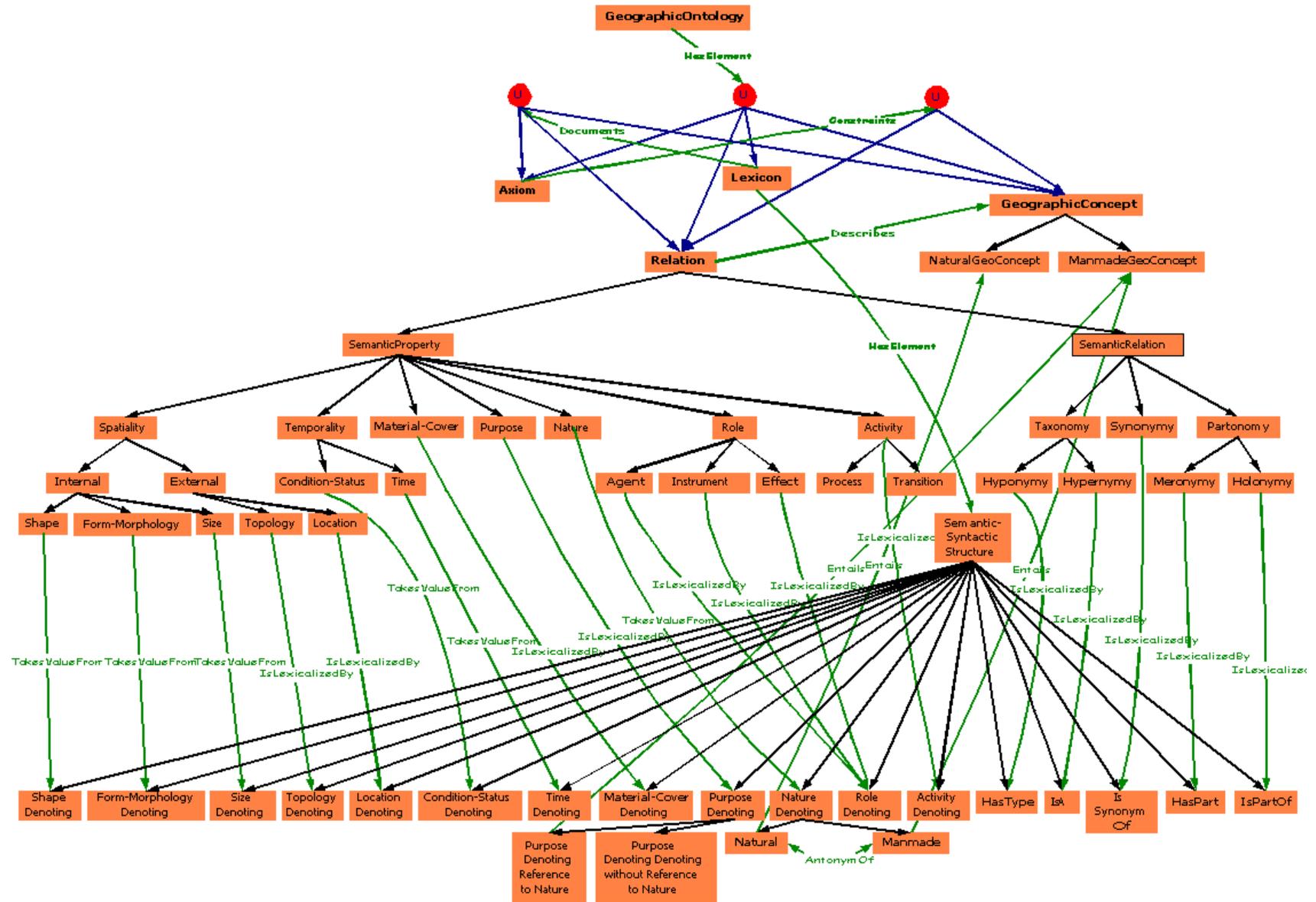


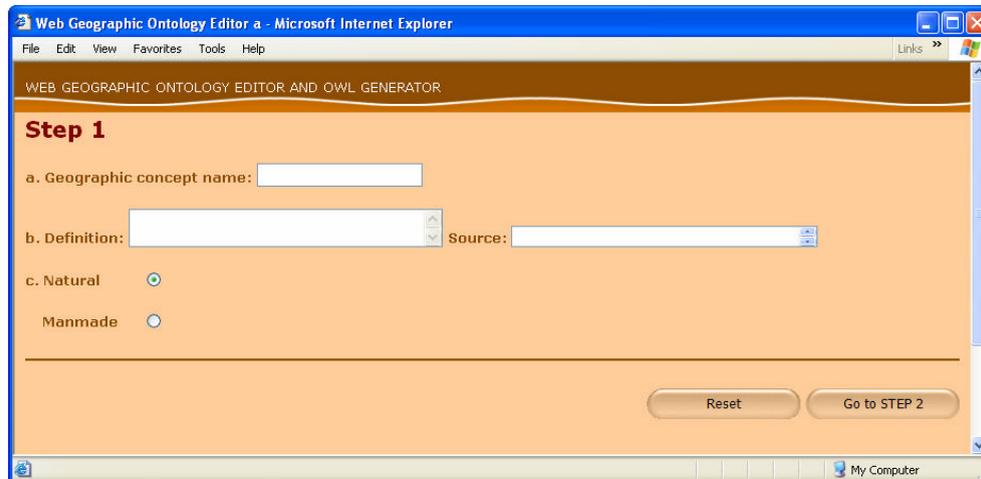
Figure 2: Visualization of the generic ontology

## WEB GEO-ONTOLOGY EDITOR AND OWL GENERATOR

When building/constructing a geo-ontology for application reasons, the importance of its reuse or dissemination among other users and implementers, is not only desirable but also a necessity. Furthermore, when this procedure takes place over the internet, the existence of a web geo-ontology editor that generates XML based files gives meaning to the so called “interoperability” on the web.

Having established a way to build a geo-ontology with a generic scheme using OWL, the second step is to enable users to have an application interface that helps them create their own ontologies. This work introduces a web geo-ontology editor that has been implemented for research and educational purposes. The editor helps new users to understand basic ontology concepts by giving them the ability to create an ontology and also see the resultant OWL Full file and save it locally.

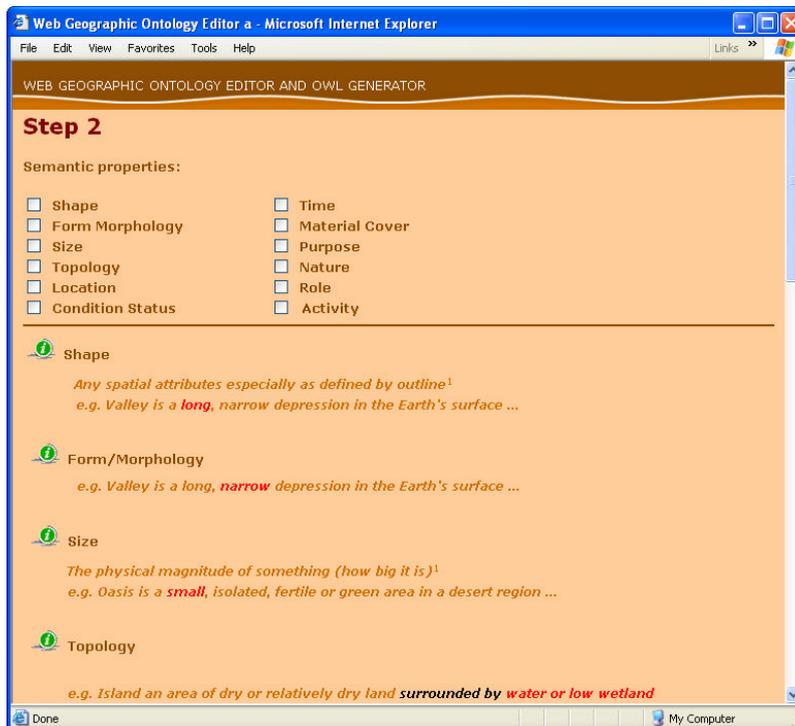
The interface is a web page implemented in .asp format and runs at the moment in a intranet environment for testing. The user fills in specific fields beginning from the name of the basic ontology concept and its definition taken from a specific source (step 1, figure 3). Then (step 2, figure 4), he/she is asked to select the semantic properties of the concept; the user is also given help tools that explain each semantic property along with an example. In the next step (step 3, figure 5), the user is asked to chose which semantic/syntactic structures exemplify or lexicalize the selected semantic properties. Steps 4 and 5 (figures 6 and 7) comprise in the same procedure, this time for the semantic relations of the geographic concept. At the last step the user can also save the resultant OWL file<sup>4</sup>. The final step generates the schema of the ontology. Needless to say, that steps1 to 5 have to be repeated for each geographic concept to be included in the ontology.

The image shows a screenshot of a web browser window titled "Web Geographic Ontology Editor a - Microsoft Internet Explorer". The browser's address bar and menu bar are visible. The main content area of the browser displays a web page with the title "WEB GEOGRAPHIC ONTOLOGY EDITOR AND OWL GENERATOR". Below the title, the page is labeled "Step 1". The form contains three main sections: "a. Geographic concept name:" with a text input field; "b. Definition:" with a text area and a "Source:" label followed by a text input field; and "c. Natural" and "Manmade" with radio buttons. At the bottom right of the form, there are two buttons: "Reset" and "Go to STEP 2". The browser's taskbar at the bottom shows "My Computer".

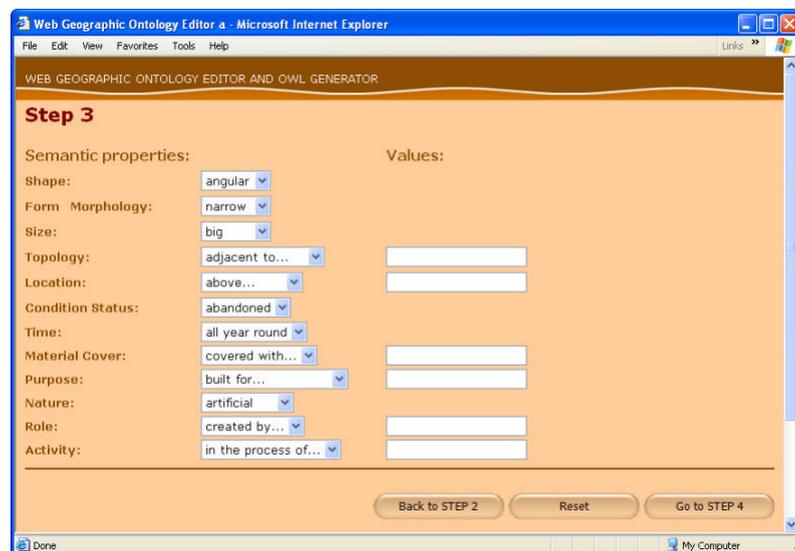
**Figure 3:** The first step in filling in the Web geo-ontology editor and OWL generator, where the user is asked to give the geographic concept’s name, its definition and where the latter came from (its source)

---

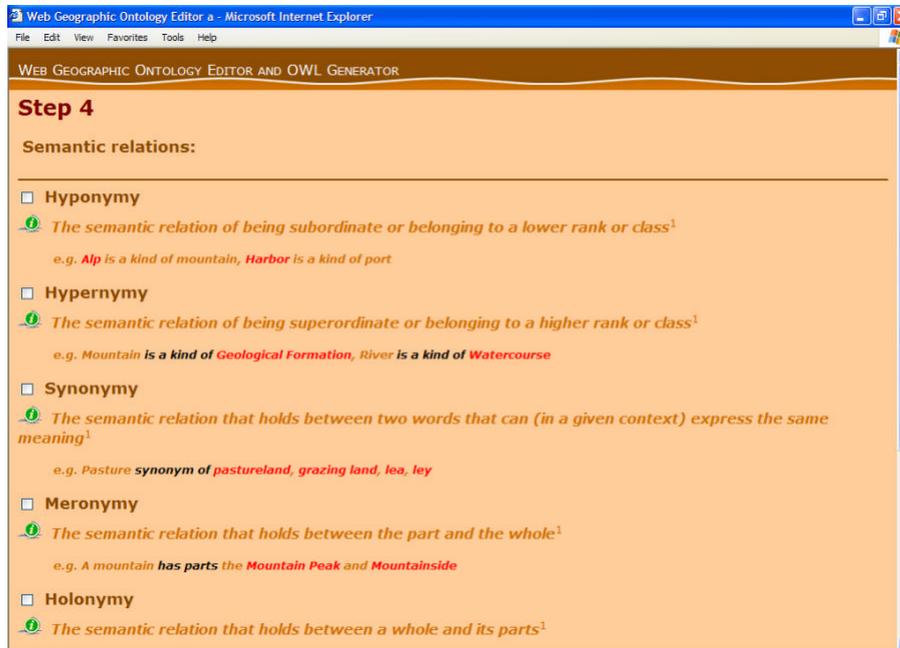
<sup>4</sup> The complete editor will be shortly available and accessible at the previously mentioned webpage.



**Figure 4:** The second screenshot of the editor where the user defines which semantic properties of the geographic concept can be determined and also they are given an info tool that explains each semantic property along with an example



**Figure 5:** The third step in filling in the editor where the user defines which linguistic or syntactic patterns exemplify the semantic properties and assigns their values



**Figure 6:** A screenshot of the fourth step of the editor where the user defines which semantic relations of the geographic concept can be determined and also they are given an info tool that explains each semantic relation along with an example



**Figure 7:** The fifth step in filling in the Web geo-ontology editor and OWL generator where the user defines which linguistic or syntactic patterns exemplify the semantic relations and assigns their values. This last step also saves the OWL file

## FUTURE DEVELOPMENT

In this paper we demonstrated the steps for developing a web tool for ontology generation using a generic “ontology” of designing a geographic ontology in the background. The guidelines for geo-

ontology generation must incorporate matters that have not been yet examined such as spatial semantics and boundaries as well as the notion of vagueness.

On the other hand, future development of the tool includes the creation of a full blown editor for geo-ontologies accessed from the web, combined with a graphical representation of the OWL Full file like many desktop ontology editors do. The automated creation of a graphical view will be an SVG file that can easily zoomed in/out, panned and searched for its content by the user. SVG is an xml based format and has obvious similarities to the xml structure of owl files.

### **ACKNOWLEDGMENTS**

This work has been partially supported by the Heraclitus Research Grant Program 2.2.3.b of the Hellenic Ministry of National Education. Special thanks to Spyros Papoutsis for helping us programming the Web Geographic Ontology Editor and OWL Generator. We would also like to thank Professor Marinos Kavouras for his very constructive comments.

### **BIBLIOGRAPHY**

McGuinness Deborah L., and van Harmelen Frank, eds. (2004). OWL Web Ontology Language Overview, W3C Recommendation <http://www.w3c.org/TR/owl-guide/>, last access: 2004-11-30

Protégé Ontology Editor (2004) <http://protege.stanford.edu/>, release 2.1.2.

Tomai Eleni and Kavouras Marinos (2004), From "Onto-GeoNoesis" to "Onto-Genesis" The Design of Geographic Ontologies, *Geoinformatica* 8(3), 285-302,.

Smith Michael K., Welty Cris, McGuinness Deborah L, eds. (2004). OWL Web Ontology Language Guide, W3C, Recommendation <http://www.w3c.org/TR/owl-guide/>, last access: 2004-11-30