Tools for Intelligent Search in Collections of Digitized Manuscripts

Maria Nisheva-Pavlova, Pavel Pavlov

Faculty of Mathematics and Computer Science, "St. Kl. Ohridski" University of Sofia
5 James Bourchier Blvd., Sofia 1164, Bulgaria
{marian | pavlovp}@fmi.uni-sofia.bg

Abstract

The paper presents a methodology for development of tools for semantics oriented search in repositories of digitized manuscripts. This methodology is based on the application of some Semantic Web technologies to existing manuscript collections that may include: electronic catalogue containing marked-up manuscript descriptions, full texts of manuscripts, digital images of manuscript pages. It is directed to the development of software environments that will be able to deal with complex user queries and answer questions using ontological knowledge and dictionaries of synonyms.

1 Introduction

Current search technology is mostly keyword-based. Usually the user provides the search engine with a phrase or combination of words which s/he expects to find in a set of documents. There is no straightforward, reasonable interpretation of these words as denoting a concept. At the same time many users may prefer to formulate queries in terms of high-level semantic concepts that are more relevant to their professional knowledge and experience. In these cases the search engine is provided with a phrase which is intended to denote an object about which the user is trying to gather information. The goal is to locate a number of documents which together will give him/her the necessary information. We believe that it might be useful for the search engine to have an understanding of these concepts denoted by the search phrase. Understanding the denotation can help to augment and refine the search and thus will improve the obtained results. A first step to supplying the search engine with such understanding is to develop proper ontologies defining the relationships between the concepts in the corresponding domain(s). Thus the search may be enriched with the addition of a set of synonyms and phrases denoting more specific concepts than the one given by the user.

Here we present a methodology for development of tools for ontology-driven search in repositories of digitized manuscripts. Our methodology is designated to assist the search activities in collections that may enlist:

- electronic catalogue containing manuscript descriptions compatible with the document type definition structure suggested by the project MASTER (Manuscript Access through Standards for Electronic Records, http://www.cta.dmu.ac.uk/projects/master/) and adopted by TEI (the Text Encoding Initiative, http://www.tei-c.org/);
- marked-up full texts of manuscripts that may be written in different languages;
- digital images of manuscript pages.

It is directed to the development of software environments that will be able to deal with complex user queries and answer questions such as "When (or where) are written manuscripts in which natural calamities or irregularities are mentioned?".

In this work we lay aside the problems connected with the processing of questions formulated in natural language and concentrate on queries containing phrases like "natural calamities or irregularities". More precisely, the queries may contain conjunctions and disjunctions of key words and phrases. As a result of the processing of a user query, a set of documents (manuscript descriptions and/or texts of manuscripts) and images of manuscript pages containing words and phrases semantically related to these used in the query should be retrieved and properly visualized. The scope of the queries should not be predefined, but it is necessary to have a clear idea about their area(s) in order to provide and describe the corresponding domain knowledge.

2 Main Characteristics of the Suggested Approach

The emphasis in the suggested methodology falls on the following main topics:

- Development of proper ontologies describing the conceptual knowledge relevant to the chosen domain(s). These ontologies define sets of concepts with their basic properties and the relationships (mainly hierarchical in our case) between them. The concepts should be defined in many languages.
- Definition of an ontology providing the vocabulary for describing the content of digital images of manuscript pages. This ontology should describe the main characteristics of the digital images and their content in accordance with the provided conceptual knowledge.
- Development of proper intelligent agents for search and processing purposes that are able to retrieve and filter documents and images by their semantic properties.

In the next sections we discuss some advisable implementation details concerning the corresponding software components.

3 Ontology Development

An *ontology* is an explicit specification of a conceptualization. Ontologies define domain concepts and the relationships between them, and thus provide a domain language that is meaningful to both humans and machines. They are formal theories supporting knowledge sharing and reuse and in this sense form the basis of the so-called Semantic Web - "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." (Berners-Lee, 2001).

Each ontology should adequately represent a specific domain and allow some kind of formal reasoning. Ontologies should be both understandable by humans and processable by software agents. They can be used in particular to annotate Web resources. Furthermore, since ontologies will evolve over time, they need to be maintainable. This means that the proper ontology modeling tools should provide a user-friendly view on the ontology and support an iterative working style with rapid turn-around times. Tools should also provide intelligent services that reveal inconsistencies and hidden dependencies among definitions.

All these requirements are satisfied by Protégé (Gennari et al., 2003; Knublauch, 2003) and especially by its OWL Plugin (Knublauch, 2004a) and we strongly recommend it as a development environment.

The Web Ontology Language (OWL) (Smith et al., 2004) is widely accepted as the standard language for ontology construction and sharing Semantic Web contents. Protégé is an open ontology development environment with a large community of active users. Recently Protégé has been extended with support for OWL, and has become one of the leading OWL tools.

Protégé provides functionality for editing classes, slots (properties), and instances. Its user interface consists of several screens, called *tabs*, which display different aspects of the ontology in different views. Each of the tabs can be filled with arbitrary components. Most of the existing tabs provide a tree-browser view of the model, with a tree on the left and details of the selected node on the right hand side. The details of the selected object are typically displayed by means of *forms*. The forms consist of configurable components, called *widgets*. Typically, each widget displays one property of the selected object. There are standard widgets for the most common property types, but ontology developers are free to replace the default widgets with specialized components.

The OWL Plugin is a complex Protégé plugin with support for OWL. It can be used to load and save OWL files in various formats, to edit OWL ontologies and to provide access to reasoning based on description logic. As shown on Figure 1, the OWL Plugin's user interface provides various default tabs for editing OWL classes, properties, forms, individuals, and ontology metadata.

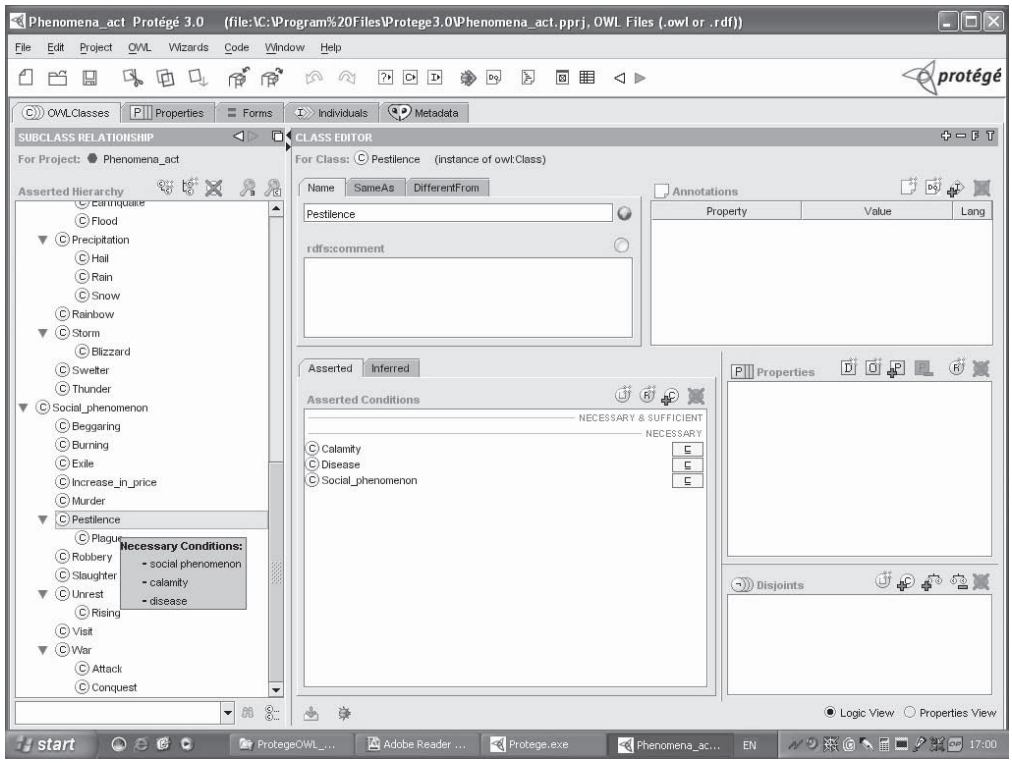

Figure 1. The class editor of the Protégé OWL Plugin

The OWL Plugin can interact with any description logic based reasoner that supports the standard DIG interface, such as Racer. In our context, *reasoning* means to infer new knowledge from the statements asserted by an ontology designer. *Reasoners* are tools that take an ontology and perform reasoning (including classification and consistency checking) with it. Classification is used to infer specialization relationships between classes from their formal definitions. Basically, a classifier takes a class hierarchy including the logical expressions, and then returns a new class hierarchy, which is logically equivalent to the input hierarchy. Protégé can display the classification results graphically. After the user has clicked the classify button, the system displays both the asserted and the inferred hierarchies, and highlights the differences between them. Using OWL, ontology designers could just add a new concept by describing its logical characteristics, and the classifier would automatically place it in its correct position. Furthermore, it would report the side-effects of adding a new class.

The class hierarchies in an OWL ontology can be viewed and navigated conveniently by an extension of the Protégé OWL Plugin called OWLViz (Figure 2). OWLViz allows the user to compare the *asserted* class hierarchy and the *inferred* class hierarchy. It has the facility to save both the asserted and inferred views of the class hierarchy to various graphics formats.

4 Intelligent Software Agents

The approach we suggest for intelligent search in electronic collections of manuscript descriptions and marked-up full texts of manuscripts is directed to the use of a set of general-purpose and specialized ontologies that can be accessed by the user, or "his" agents, in formulating and processing of relatively complex queries.

The main actions related to the execution of user queries have been performed by four agents: the Query Formulation Agent, the Ontology Agent, the Search Service Agent and the Integration Agent.

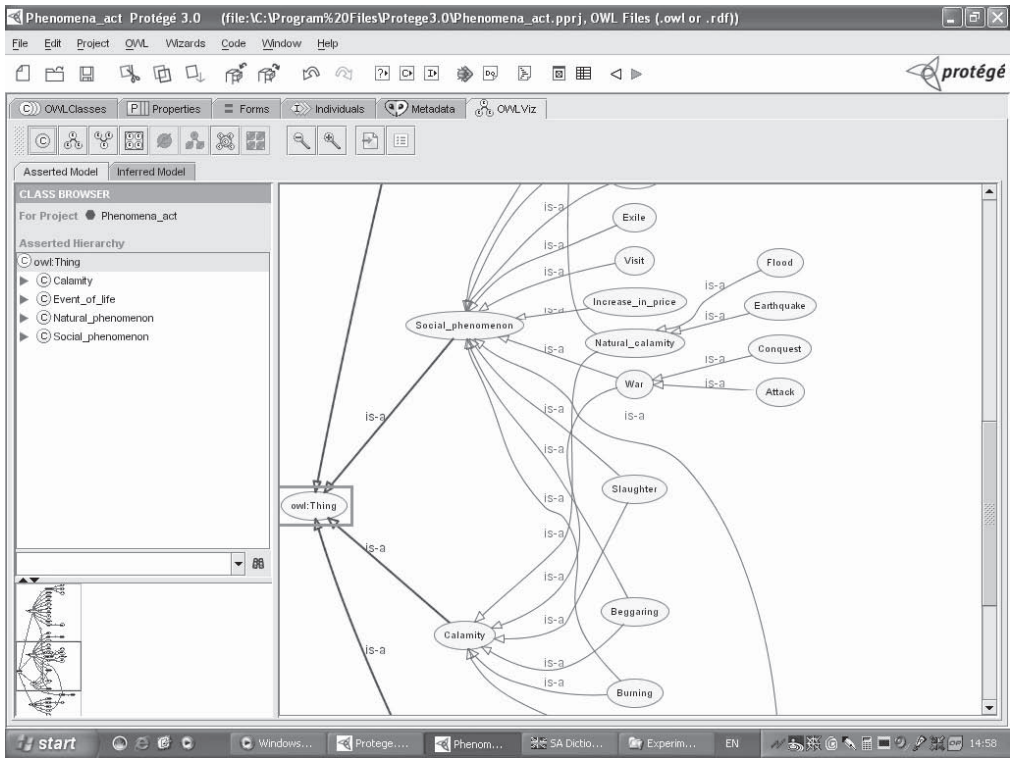

Figure 2. Visualization of the class hierarchy with OWLViz

The Query Formulation Agent interacts with the user to receive his original (initial) query. Once a query has been specified, the Query Formulation Agent decomposes it into subqueries (that do not contain conjunctions or disjunctions of phrases) and sends these subqueries to the Ontology Agent for augmentation and refinement. This agent uses the available ontology resources as a vocabulary for the representation of domain knowledge.

More precisely, the Ontology Agent considers the subqueries as domain concepts and adds to each of them the corresponding more specific concepts from the ontologies and some synonyms of the main terms from an appropriate dictionary. In this way the Ontology Agent can augment the original search query. Then it sends the new set of subqueries to the Search Service Agent for further processing in a standard way.

The Search Service Agent expects that the collection of digitized manuscripts of interest consists of XML documents. In particular, the catalogue descriptions are supposed to be compatible with the document type definition structure suggested by the project MASTER. The search should be performed in all elements of each single XML document or in a specific element indicated by the user.

At last the Integration Agent “compiles” and visualizes the subquery processing results from the various available resources.

As a starting point for the implementation of the Ontology Agent one should use most of the formats in which Protégé can save ontology descriptions (OWL, RDF, CLIPS). We prefer to use the standard DIG code generated by the Protégé OWL Plugin (Figure 3). For the implementation of the Search Service Agent we recommend to choose a proper scripting language, e.g. VBScript. XQuery (Katz, 2003) should be a good alternative but nowadays it does not support the use of dynamic contexts where the concepts of current collection and document set are described by variables. This restriction may cause some painful problems.

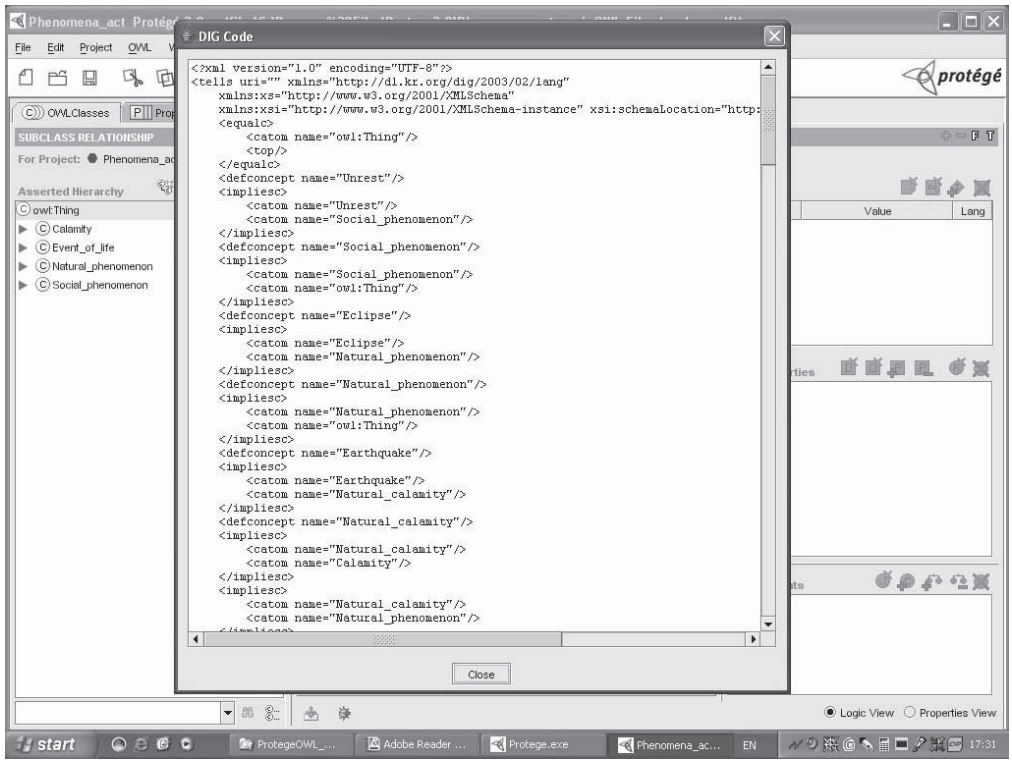

Figure 3. DIG code generated by the Protégé OWL Plugin

5 Intelligent Search in Collections of Images

For the purposes of search in collections of digital images of manuscript pages we recommend a methodology that is very similar to the one suggested in (Knublauch et al., 2004b).

First, a domain ontology providing the vocabulary for describing the contents of images (particular pages of manuscripts) should be developed.

In order to describe the page images, an image ontology containing a single class (e.g. `Image`) has to be defined. It should give an account of the main properties of the considered type of images: the dimensions of the image, its location and a link to a specific OWL class from the domain ontology.

A new ontology has to import the above two ontologies. It should contain instances of the exemplary `Image` class and should use the classes from the domain ontology as contents values.

An instance in this new ontology ought to describe a particular image of manuscript page. After the instances have been created, they can be exported to a Web server in order to be found and processed by the software agents. Supplied with a search concept, an agent may retrieve and filter images using a proper reasoner as e.g. the OWL reasoner from Jena library (Reynolds, 2004).

6 Conclusions

We are working now on some possible extensions of the suggested approach. One of them is connected with the provision of proper metadata describing the semantics of the collection as a whole and the use of these metadata to improve the search effectiveness (e.g. to reduce the set of elements in the catalogue descriptions that should be searched).

The proposed methodology could be applied to other areas. For example, it should be properly used for linking scientific and educational resources (papers, reports, lecture notes, electronic books, images etc.) into a Semantic Web and improving the access to them in order to make human learning and extraction of knowledge more effective.

Acknowledgements. This work has been funded by the EC FP6 Project "Knowledge Transfer for Digitisation of Cultural and Scientific Heritage in Bulgaria" (KT-DigiCULT-BG).

References

Berners-Lee, T. et al. (2001). The Semantic Web. *Scientific American*, May, 35-43.

Gennari, J. et al. (2003). The Evolution of Protégé-2000: An Environment for Knowledge-based Systems Development. *International Journal of Human-Computer Studies*, 58(1), 89–123.

Katz, H. (2003). An introduction to XQuery. http://www-106.ibm.com/developerworks/xml/library/x-xquery.html, last accessed on April 8, 2005.

Knublauch, H. (2003). An AI Tool for the Real World: Knowledge Modeling with Protégé. *JavaWorld*, June 20.

Knublauch, H. et al. (2004a). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. *Third International Semantic Web Conference*, Hiroshima, Japan.

Knublauch, H. et al. (2004b). Weaving the Biomedical Semantic Web with the Protégé OWL Plugin. *First International Workshop on Formal Biomedical Knowledge Representation*, Whistler, BC, Canada.

Reynolds, D. (2004). Jena 2 Inference Support. http://jena.sourceforge.net/inference/index. html, last accessed on April 8, 2005.

Smith, M. et al. (2004). OWL Web Ontology Language Guide (W3C Recommendation). http://www.w3.org/TR/owl-guide/, last accessed on April 8, 2005.