Преглед НЦД 9 (2006), 13–17

Maria M. Nisheva-Pavlova
Pavel I. Pavlov
(Faculty of Mathematics and Computer Science,
Sofia University “St. Kliment Ohridski”)

ON THE APPLICABILITY OF PROTÉGÉ/OWL IN BUILDING SOFTWARE TOOLS FOR INTELLIGENT SEARCH IN DIGITIZED COLLECTIONS OF MANUSCRIPTS

Abstract: We present some considerations related to the implementation of a methodology for development of software tools for intelligent (more precisely, semantics oriented) search in repositories of digitized manuscripts. This methodology is oriented to the construction of proper ontologies relevant to the domain(s) of the user queries and some intelligent agents for search and query processing purposes. The main features of the Web Ontology Language (OWL) and the open ontology development environment Protégé are analyzed from the point of view of their applicability in building software tools for intelligent search. The conclusion is that the OWL Plugin of Protégé is an excellent instrument for the development of tools for semantics oriented search in digitized collections of manuscripts.

Keywords: Manuscript Digitization, Semantic Web, XML, OWL, Protégé

1. Introduction

Information technologies play a significant role in lots of successful projects directed to digital preservation of cultural and scientific heritage. The growth of the number of digitized collections of manuscripts and printed editions gives rise to the elaboration of proper software tools assisting the access to these collections and making the best use of them.

A consequence in this direction is the growth in the development of proper search methods and tools. Instead of the facilities supported by the traditional keyword-based search engines, many users prefer to formulate queries in terms of high-level semantic concepts that are more relevant to their professional needs. In these cases the search engine is provided with a phrase which is intended to denote an object or an event about which the user is trying to gather information. The aim is to find a suitable set of documents which together will give him the necessary information.

In [5] we suggest a methodology for development of tools for semantics oriented search in repositories of digitized manuscripts. This methodology is designated to assist the search activities in collections that may enlist XML documents which should be:

- catalogue descriptions of manuscripts compatible with the document type definition structure suggested by the project MASTER and adopted by TEI;
- marked-up full texts of manuscripts that may be written in different languages.

It is directed to the development of software environments that will be able to deal with user queries containing words or phrases that are considered as domain concepts.

The emphasis in our methodology falls on two main types of activities:

- 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.
- Development of proper intelligent agents for search and processing purposes that are able to retrieve and filter documents by their semantic properties.

The paper discusses some considerations related to the implementation of this methodology. The main features of the Web Ontology Language (OWL) and the ontology development environment Protégé are analyzed from the point of view of their applicability in building software tools for intelligent search. We draw the conclusion that the OWL Plugin of Protégé is a quite proper instrument for the development of tools for semantics oriented search in digitized collections of manuscripts.

2. Building Ontologies with Protégé/OWL

As it was discussed above, our methodology is oriented to the construction of (1) proper ontologies relevant to the domain(s) of the user queries and (2) some intelligent agents for search and query processing purposes. An *ontology* is an explicit specification of a conceptualization. Ontologies define domain concepts, their properties 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 which play a significant role in the development 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" [1].

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.

There are lots of ontology description languages and ontology development software tools available nowadays.

The Web Ontology Language (OWL) [6] is widely accepted as the standard language for ontology construction and sharing Semantic Web contents. It is based on a description logic model that makes possible to define and describe concepts and to make reasoning about them.

Protégé [2] is an open ontology development environment with a large community of active users. Protégé's model (the internal representation mechanism for ontologies and knowledge bases) is based on a flexible metamodel, which is comparable to object-oriented and frame-based systems. It basically can represent ontologies consisting of classes, properties (slots), property characteristics (facets and constraints), and instances. Recently Protégé has been extended with support for OWL, and has become one of the leading OWL tools.

Protégé provides functionality for editing classes, 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.

The OWL Plugin [4] 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 tools based on description logic. The OWL Plugin's user interface provides various default tabs for editing OWL classes, properties, individuals, and ontology metadata. Protégé can save ontology descriptions in various formats (OWL, RDF, CLIPS etc.). As a starting point for the implementation of intelligent search agents we prefer the standard DIG code generated by the OWL Plugin because of its simple and clear structure.

The most important view in the Protégé OWL Plugin is the OWLClasses tab (Fig. 1). This tab displays the tree of the ontology's classes on the left, while the selected class is shown in a form in the center. The tree widget of the OWLClasses tab is organized according to the subclass hierarchy. Protégé users can browse, view, and edit the classes from the tree, create new subclasses, and move classes easily with drag-and-drop.

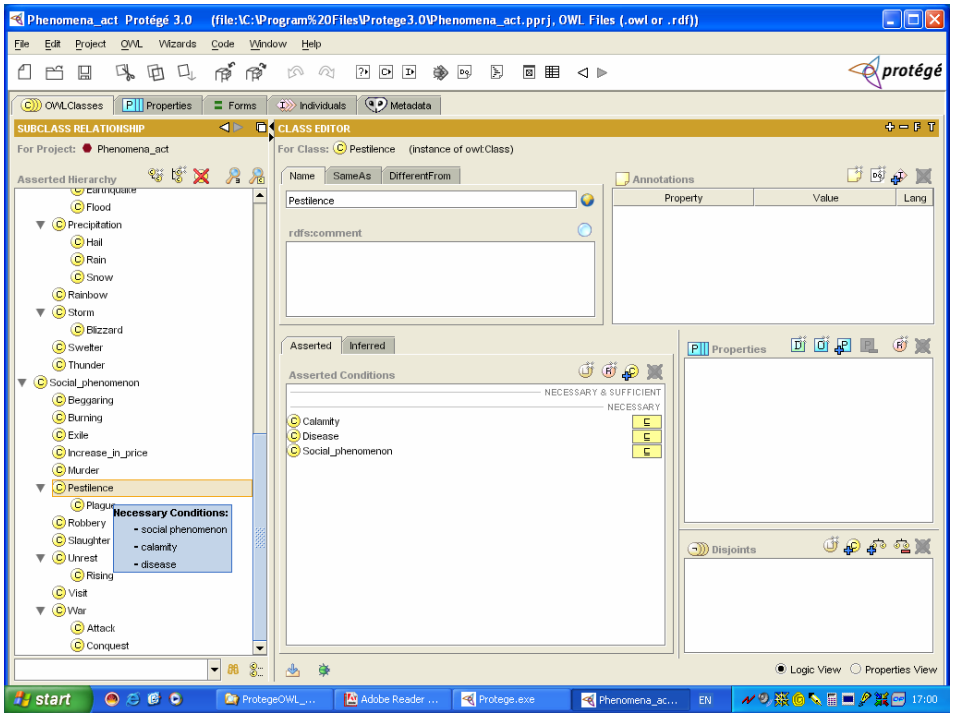

Fig. 1. The class editor of the Protégé OWL Plugin

The OWL Plugin also allows to navigate and edit ontologies according to other relationships between classes. Thus the user interface of the OWL Plugin of Protégé is very convenient for knowledge engineers and can be accessed without serious problems by other types of users (philologists, historians, librarians etc.).

3. Other Useful Features of Protégé/OWL

The OWL Plugin can interact with any reasoner that supports the standard DIG interface, such as Racer [3]. During ontology design, the most interesting reasoning capability from this type of tools is classification.

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. The class hierarchies in an OWL ontology can be viewed and navigated conveniently by an extension of the Protégé OWL Plugin called OWLViz (Fig. 2). OWLViz has the facility to save both the asserted and inferred views of the class hierarchy to various graphics formats. These visualization facilities can utilize the interaction between knowledge engineers and domain specialists and thus should increase the effectiveness of the knowledge acquisition process.

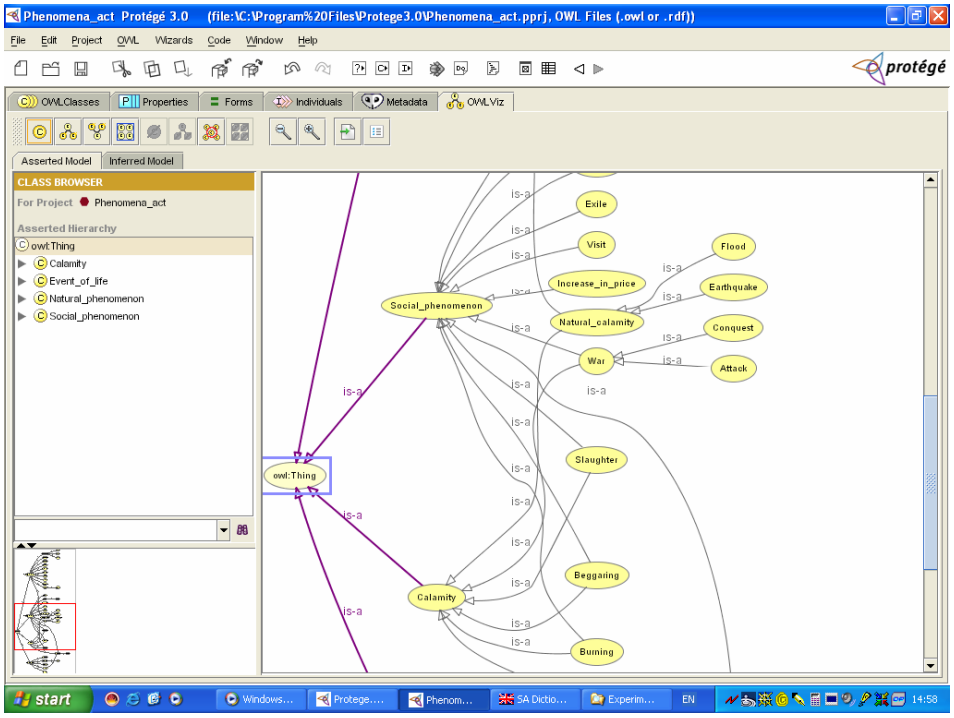

Fig. 2. Visualization of the class hierarchy with OWLViz

The mentioned reasoning capability associated with description logic is of particular importance because it allows the user to provide intensional definitions for the classes. 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. In this context Protégé/OWL seems to be a proper tool for building software environments intended to

perform some kinds of clustering of manuscript collections in accordance with the features of the authors or scribes of manuscripts.

The OWL Plugin provides a mechanism to execute small test cases. The user can press an ontology test button, and then the system will execute a configurable list of tests. These tests are small Java programs that basically take a class, property, individual, or ontology as its input, verify arbitrary conditions on them, and in case of failure, return an error message.

Protégé/OWL offers a standard set of tests for various best ontology design practices. The list of standard ontology tests can be easily extended by programmers, so that the system should execute additional user-defined tests uniformly. These additional tests could for example ensure the application of project-specific design patterns, naming conventions and other useful practices.

4. Conclusion

The considerations discussed in Section 2 and Section 3 can serve as arguments for the conclusion that Protégé and especially its OWL Plugin is a perfect instrument for the development of software tools for intelligent search in digitized collections of manuscripts.

Therefore we decided to use Protégé/OWL as a basis for the implementation of an experimental software tool for intelligent (more precisely, semantics oriented) search in collections containing XML documents that could be catalogue descriptions or marked-up full texts of manuscripts. The results of the first experiments with this tool may be evaluated as promising.

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) coordinated by the Institute of Mathematics and Informatics of the Bulgarian Academy of Sciences.

References

[1] Berners-Lee, T., J. Hendler, O. Lassila. *The Semantic Web*. Scientific American, May 2001, pp. 35–43.

[2] Gennari, J., M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, S. Tu. *The Evolution of Protégé-2000: An Environment for Knowledge-based Systems Development*. International Journal of Human-Computer Studies, Vol. 58(1), pp. 89–123, 2003.

[3] Haarslev, V., R. Moeller. RACER user's guider and reference manual. http://www.cs.concordia.ca/~faculty/haarslev/racer/, 2003.

[4] Knublauch, H., R. Fergerson, N. Noy, M. Musen. *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*. Third International Semantic Web Conference, Hiroshima, Japan, 2004.

[5] Nisheva-Pavlova, M., P. Pavlov. *Tools for Intelligent Search in Collections of Digitized Manuscripts*. In: M. Dobreva, J. Engelen (Eds.), "*From Author to Reader: Challenges for the Digital Content Chain. Proceedings of the 2005 ELPUB Conference*". Peeters Publishing Leuven, 2005, pp. 145-150.

[6] Smith, M., C. Welty, D. McGuinness. *OWL Web Ontology Language Guide (W3C Recommendation)*. http://www.w3.org/TR/owl-guide/, 2004.

marian@fmi.uni-sofia.bg
pavlovp@fmi.uni-sofia.bg