Ontology Development Portal

G. Zhao, R. Verlinden, R. Meersman
Department of Computer Science
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels
BELGIUM

Abstract: This paper presents the design and development of an ontology development portal. It is based on a two-layered approach to ontology modeling, called DOGMA. The portal is architected in the framework of DOGMA Application Knowledge Development Architecture, in a context of ontology-based application knowledge engineering. It is part of a suite of facilities supporting the ontology engineering process, mainly devoted to ontology storage, sharing and communication. It is deployed in an ontology development project to support the team effort to ontology engineering, collaborative and distributed modeling, consensus building and evaluation. The paper introduces the DOGMA representation framework and describes how the framework is translated into an ontology development portal in the architectural framework of DOGMA-AKDA. It also discusses how the major user functionalities of ontology modeling and maintenance are implemented on the top of the DOGMA ontology server.

Key-Words: ontology, knowledge engineering, portal, architecture

1 Introduction

The Ontology Development Portal (ODP) is one of the facilities supporting the DOGMA (Developing Ontology-Guided Mediation for Agents) ontology-based knowledge engineering process. It targets team knowledge development and maintenance, thus emphasizing communication, consensus building and collaborative distributed knowledge modeling by domain experts, business analysts, knowledge analysts and ontology engineers.

Domain and application semantics are always multi-variate, configured in multiple perspectives, abstraction and delimitation with respect to contexts. Natural languages are characterised by semantic under-specification and synonymous or alternative means of expressing the same idea with different semantic and referential emphases. This shows the multi-dimensional nature of semantics of the subject world. It will be naïve to assume that a reductionistic model of semantics will support and scale well in face of the multiple perspective real-life ontology-based applications. The ODP does not deal with a small set of consistent mathematical formulae, but with a large population of conceptualization potential, which is to be selected and committed to by applications.

The ODP is based on DOGMA, an ontology representation framework. It is part of a knowledge development architecture, AKDA (Application Development Architecture), deployed for ontology repository, sharing and publication in an EU research & technology project, FF POIROT\(^1\).

In the following sections, we shall introduce two main underpinnings of the ODP: the DOGMA ontology framework and development architecture and then describe the ODP’s software architecture and functionalities.

2 Ontology representation framework

Ontology is an approximate semiotic representation of conceptualization of a subject world. The DOGMA paradigm of ontology representation defines an ontology base as a set of lexons. These are defined as 5-tuples of the form \( \langle \gamma, t_1, r_1, r_2, t_2 \rangle \), where \( \gamma \) is a context identifier, \( t_1 \) and \( t_2 \) are terms and \( r_1 \) and \( r_2 \) are roles. Intuitively, a lexon stands for a fact expressing a possible (binary) relationship between two concepts, referred to by the terms involved, and within the identified context. The two roles express both sides of the relationship between \( r_1 \) and \( r_2 \).

\(^1\) FF POIROT is a research & technology project of the EU’s IST 5th framework program. Its theme is the use of ontology for financial fraud prevention and detection.

www.ffpoirot.org
refers to resource documents in which the respective fact is "explained" to the satisfaction of ontology users and developers [3].

While lexons capture application-independent ontological conceptions, the commitment to them grounds them to a particular application and task context [5]. It not only maps application specific symbols to the lexons in the ontology base, but also specifies the constraints and rules to define the specific behavior of the application on the ontological conceptualisation. On the commitment layer, application/task independent conceptions are constrained, instantiated, composed and connected into semantically consistent and unambiguous networks in given application contexts and task perspectives. The commitment layer acts as interface between generic conceptions and their specific inferences and interpretations in the application system.

The DOGMA paradigm factors out the complexity of modeling and processing into two basic layers: ontology base and commitment layer. This facilitates design scalability and flexibility, and model versatility and reusability across applications. It stresses support for consensus building and communication in ontology development.

3 Knowledge Development Architecture

The distinction of lexon base and commitment layer provides the perspective in which knowledge modeling is to take place. It is translated into software architecture by means of DOGMA-AKDA.

AKDA [7] is derived from DOGMA. The architecture builds the principle of ontology engineering into an application-oriented knowledge engineering context.

It is stratified into three layers: lexons, commitments and uses. Each architectural layer reflects three major dimensions of knowledge engineering. The lexon layer presents ontology models of domain-dependent or independent concepts and relationships. The commitment layer composes and connects lexons into commitments in the light of application domain specific tasks. The use layer takes the view of application systems and packages commitments into knowledge components and assembles them into knowledge solutions.

The architecture features 6 viewpoints from 4 main categories of knowledge system developers. These viewpoints are adopted in the corresponding knowledge development processes. Each process targets a particular work deliverables.

<table>
<thead>
<tr>
<th>Users</th>
<th>Viewpoints</th>
<th>Processes</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain expert, Business</td>
<td>Applicational</td>
<td>Scoping</td>
<td>Knowledge constituent model</td>
</tr>
<tr>
<td>analyst, Knowledge analyst</td>
<td>Ideational</td>
<td>Signifying</td>
<td>Lexons</td>
</tr>
<tr>
<td>Ontology developer</td>
<td>Structural</td>
<td>Grouping</td>
<td>Models</td>
</tr>
<tr>
<td>Application knowledge</td>
<td>Task</td>
<td>Grounding</td>
<td>Commitments</td>
</tr>
<tr>
<td>developer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application system developer</td>
<td>Components</td>
<td>Modularizing</td>
<td>Assembles</td>
</tr>
<tr>
<td></td>
<td>System</td>
<td>Configuring</td>
<td>Solutions</td>
</tr>
</tbody>
</table>

Table 1 Architectural viewpoints

The applicational viewpoint adopted by domain experts, business or knowledge analysts seeks to present a global view of the structure of business logic or domain/application semantics in a knowledge constituent model. In the ideational and structural viewpoint, the ontology developer takes a microscopic view on each knowledge constituent, extracts and abstracts lexons, and organizes or aligns them into logical groups. The application knowledge developer grounds the lexons in the viewpoint of tasks. Organized around a taxonomy of application tasks, lexons are used (constrained, instantiated, composed, and connected) to create basic pieces of business logic. The commitments are packaged into knowledge components in view of system contexts to build a component library, these components are then configured into solutions.
Fig. 2 DOGMA-AKDA

The knowledge development framework (see Fig. 2) represents application semantics derived in three layers from the knowledge resources. They are referenced and deployed as knowledge artifacts. The knowledge development facility supports two tracks of knowledge system development: development of knowledge components and knowledge processors such as ontology interpreters, business process managers, inferencers. It features software development framework, ontology libraries and development portal.

4 Ontology development portal

In this section, we shall use a deployment of the ODP in the project FF POIROT to illustrate its design and implementation. One of the tasks is building and validating lexons from these knowledge resources collected and compiled in the earlier phase of development. One of the facilities to facilitate the task is the FF POIROT Ontology Development Portal.

4.1 Use case summary

ODP is intended as ontology content management platform used by a consortium of ontology developers and users, geographically distributed. Its users are mainly:

- Application domain experts, who have expertise on the application logic of the application domain.
- Knowledge analysts, who elicit knowledge and structure that knowledge to serve the purpose of ontology development.
- Domain ontology modeler, who extracts and abstracts lexons from the collected knowledge resources, creating an application- and task-independent conceptual model of the subject domain.
- Application ontology developer, who specifies commitments for the application rules on top of the lexon base.
- Knowledge application developer, who transforms or programs ontology-based knowledge specification (lexons and commitments) into the expected format and deploys it in the knowledge application engine.

The ODP serves a medium of publishing, sharing and validating ontology contents in a process of consensus building within the consortium. It requires

- Ontology server functionalities, such as storing ontology and its meta-data, indexing, updating, retrieving, access control, etc.
- Ontology browsing
- Ontology querying
- Ontology uploading and downloading
- Ontology validation and voting

4.2 Architecture

The architecture of ODP can be described in terms of five major function blocks as the diagram in Fig. 3 shows.

It is an extension of the OntoServer API [4], and translates the principles of DOGMA ontology modeling along the lines of AKDA. The architecture also takes into account software architecture needs, such as modularity, scalability, encapsulation of changes, multi-tiered and distributed systems.

4.3 Development

The strategy to encapsulate changes and uncertainty and allow for distributed deployment used here is layering. From the back-end relational databases to the presentation, there are such layers as SQL, Data Management, Ontology Management, Session and User Management. All layers offer the possibility of adding remote procedure calls in XML-RPC or SOAP. And within each layer, all packages,
representing major functionalities, can be deployed individually in a distributed fashion.

Fig. 3 ODP functional architecture

The Presentation package consists of JSP pages and JavaBeans. The ClientCommControl package is concerned with user management and session management. Since the current needs are largely search and retrieval, the transaction oriented session management is excluded. The LexonsManagement package consists of a lexon management facade and lexon managers, facilitating data operations on lexons. It is only one of the packages at the Application Logic Layer. Other packages will be CommitmentsManagement, and KnowledgeResourcesManagement. The DataManagement serves to encapsulate particular database management systems and database schemas from data services exposed to clients. The current implementation uses MySQL and a relational database schema. The foundation layer consists of packages concerned with facilities and manipulation of processes and data in the other ODP packages.

4.4 User Interface

Together with the available functionalities and the underlying methodology, the user interface is the most important package of the ODP from the end user point of view. It reflects the user requirements summarised in Section 1. In the following, we shall describe some major user functionalities of the ODP.

4.4.1 Browsing lexon base

ODP features a set of views on the lexon base and meta data of the ontologies stored. In addition to the ODP main menu header, the browsing view presents the navigation and contents panes (see Fig. 5).

Fig. 5 ODP browser

The content pane presents the result of queries and browsing parameters. The primary data view is a sortable table of 5 columns, representing the basic lexon structure (Context, Term, Role, Term, Co-Role). When the heading of a column is selected the entire table is sorted according to the content of that column. The experience in FF POIROT [1] shows that the sortable table together with a data filtering mechanism is an effective means of working with a large (a few hundredth to several thousands of lexons) population of lexons in a 2-dimensional view.

- The tabular representation of lexons is a user-familiar means of representing and communicating data.
- It is a productive instrument, saving operational clicks and taps and it doesn’t distract from the analytical work.
- It is a more compact way of presenting and overviewing a large volume of information than common tree or diagram visualizations.
- It is capable of presenting multiple dimensions in columns and in particular of lexons by their specific dimensions
- Together with the basic functionality of reordering table contents by alphabetically or indexing by a particular column, it is effective in browsing and checking particular groups or individual lexons.
Besides the table view of data, a graphic visualization of the lexon population, providing a more general overview, is being developed.

The navigation pane on the left represents an hierarchical structure of the lexon base, according to the meta-data of the different ontologies. By navigating and using the search refinement functionalities the selections of lexons rendered in the data view is changed.

The navigation pane features four navigation schemes in support of the main user browsing actions. The first navigation scheme (see Fig. 5) starts with ontologies in the lexon base at the root of the navigation tree. Each respective ontology node is expanded into version nodes and further into context nodes. This enables the ODP user to quickly navigate to a functional subset of lexons.

The second navigation scheme (see Fig. 6) starts with user defined lexon groups, which enable the grouping of a functional set of lexons in a context transcending manner.

The usage of the third and fourth navigation scheme supports the collaborative and management aspects of the ODP. The navigation by Contributor (see Fig. 7) enables the peer reviewer to quickly locate lexons added to the repository by a certain organization or individual. While the navigation by Date can be used to get an insight on the temporal evolution of the different ontologies.

In addition to the different navigation trees, which offer a straight forward approach for the most common combinations of browsing parameters, the user can choose to open the search refinement view. This extra navigation control pane will be opened underneath the sortable table in the data view pane.

Fig. 6 Navigation by lexon group

The usage of the third and fourth navigation scheme supports the collaborative and management aspects of the ODP. The navigation by Contributor (see Fig. 7) enables the peer reviewer to quickly locate lexons added to the repository by a certain organization or individual. While the navigation by Date can be used to get an insight on the temporal evolution of the different ontologies.

4.4.2 Uploading and downloading

The service of uploading and downloading is to enable the user to work locally on lexons with an online or offline editor of his choice, e.g. DogmaModeler [2]. Currently two import formats are implemented: XML and Excel sheet. The intended extension will cover ontology data in other formats, such as ORMML [2], OWL[6], etc.

Fig. 9 Ontology uploading

4.4.3 Voting

The ODP is intended as a collaborative and distributed ontology modeling tool. Ontology modeling is a conceptualization process, which differentiates itself from other conceptualization, with the strong requisite of shared agreement.

The ODP supports, through the Contributor and Date navigation trees, an easy mechanism for reviewing the modeling work done by specific organizations or individuals and viewing the most recent ontology updates. In addition to that the ODP provides a
voting mechanism, enabling the different reviewers to start a vote on a certain lexon or group of lexons.

Fig. 10 Ontology validation

The voting window (see Fig. 10) shows lexon meta-data, voting statistics and comments. When voting, comments can be attached to the vote. These comments or motivations accompanying the votes can be used as guidance by future voters. As the result of a vote, the lexon or group of lexons can be excluded from a future version of the ontology.

4.4.4 Context reader

Consensus building is an important task in ontology modelling. Experience shows that, among humans, rich semantics are most effectively conveyed in natural language. The meaning or semantics of lexons is intuitively interpreted with reference to their context, \(\gamma\) (see Section 2). The ODP context viewer (see Fig. 11) provides a link to the resource documents pointed to by \(\gamma\), enabling the user to refer back to the semantically rich natural language.

5 Conclusion

The ODP is designed by the DOGMA principle of ontology engineering, implemented in the architectural framework of DOGMA-AKDA and developed in a context of ontology-based knowledge engineering.

The ODP stresses the needs of ontology engineering as a team effort, such as collaborative modeling, consensus building, communication, publishing, etc.

The current development focuses on functionalities surrounding the modeling and maintenance of the lexon base, containing the application- and task independent domain models. Future effort will be devoted to the management of knowledge resources used to create commitments on top of the lexon base.

Acknowledgement:
This study is partially funded by the EU 5th framework program, IST 2001-38248.

References: