Ontological Context Visualization

Julia Dmitrieva, Yun Bei, and Fons J. Verbeek

LIACS, Universiteit Leiden,
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
{jdmitrie,ybei,fverbeek}@liacs.nl
http://bio-imaging.liacs.nl

Abstract. Ontologies contain information about concepts and their relations. Certain concepts may occur in different ontologies at the same time and these concepts can be used as glue to integrate information from related domains. In support of our efforts to represent information on interesting concepts, we have developed a visualization engine "ContextVis". Meanwhile, this tool can also be used for a visual knowledge mining and inspection of ontology integration. In this paper we present the Ontology Context Visualization with a case study on Alzheimer disease; a typical example in which contextual information on concepts allows to collect knowledge from different resources.

Key words: Ontology Visualization, OWL, Description Logic, Ontology Integration

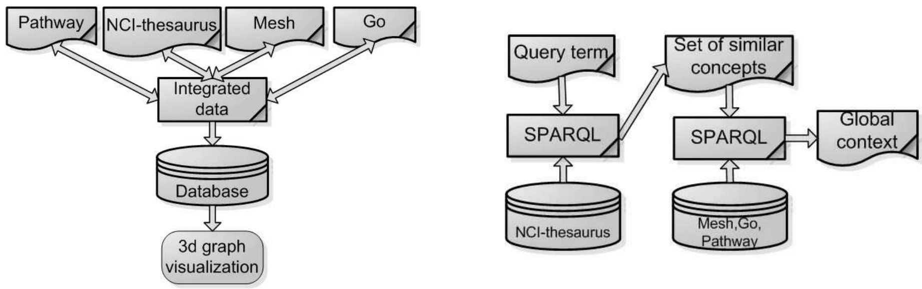
1 Introduction

Formally, ontologies existed as controlled vocabularies shared by domain experts and focused on describing the concepts and relations. In the life sciences much effort is put in accumulating all existing knowledge and creation of a domain model by means of expressive language, such as OWL-DL [1]. This has resulted in producing huge knowledge bases, that are difficult to process; operations such as making SPARQL [2] queries and reasoning [3] requires a lot of computation time. The motivation for the tool presented here is to provide a straightforward visualization of context that is created from different ontologies. Such can not be realized with existing tools since the integration [4] is only possible for those ontologies that are very much related in their concept vocabularies as well as in their structures. Moreover, visualization tools are dedicated to visualization of the just one ontology. In "ContextVis" tool we use some *interesting concept* (e.g. Alzheimer disease) as glue to connect concepts from different ontologies and to relate knowledge from different domains.

2 Tool description

The process of ontological context visualization consists of two principle parts (see Fig. 1), i.e.:

- *Knowledge Acquisition.* Here we have made use of Jena API [5] and SPARQL [2]. At this point, the ontological context is built. After that the ontological context is saved in special purpose database (MySql [6]).
- *Visualization.* During the visualization stage the data is retrieved from the database and represented as a $3D$ graph. For the implementation of graph visualization we have used the Java3D API [7].



(a) Architecture of the "ContextVis" (b) Creation of the Global Ontological Context

Fig. 1: Architecture and Global Context

3 Knowledge Acquisition

Data about the *interesting concept* are collected from different ontologies, such as PATHWAY, MESH, NCI-THESAURUS and GO. These ontologies are obtained from the OBO Download Matrix [8]. We are, however, not restricted to just these knowledge bases. The only constraint is that the ontologies must be represented in OWL format.

3.1 Global Ontological Context

The process of data acquisition starts with a search of concepts which, in their definitions, comments or labels, have words that are equal to the query term. To find these concepts we make use of SPARQL [2] queries in Jena [5]. First, the NCI-THESAURUS ontology is searched for relevant information. All retrieved concepts are saved in a temporary data structure and later will be used to trigger a similar query process in other ontologies. Subsequently, MESH ontology is searched and the concepts that contain the specified query term plus terms resulting from the search in NCI-THESAURUS are collected. Finally, in the similar fashion, the PATHWAY and GO knowledge bases are processed. According to this strategy, we have obtained a set of concepts from different ontologies (see Fig. 1). This set is referred to as the global ontological context.

3.2 Local Ontological Context

Around each concept found in different ontologies a local ontological context is created from its native ontology; this implies that we collect the information about related concepts using all kind of relationships, not just subclass/superclass. For example *part_of* from PATHWAY or *Chemical_Or_Drug_Affects_Gene_Product* from NCI-THESAURUS. The context can be created with different level of depth. Although, the inferred model is preferable in order to get relationships between concepts, we were not, however, successful in classifying NCI-THESAURUS ontology, because it is simply too big.

3.3 Database representation

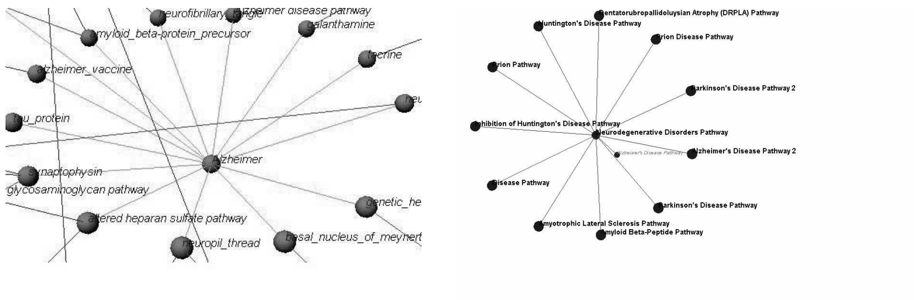
The information about the *interesting concept* with surrounding global and local context is transferred to our special purpose database. In this database, initially, only two tables are used: one for concepts and the other for relations. In order to be able to go beyond the representation of subclass and superclass relationships between concepts, a loosened OWL-DL representation of the relation is introduced. In OWL-DL the structure that connects subject with object via some predicate is written as follows:

```
<owl:Class rdf:ID="PW_0000024">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#part_of"/>
        </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#PW_0000234"/>
      </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

In Description Logic [9] this means: $P_1 \sqsubseteq \exists part\_of.P_2$, where P_1 is the term $PW\_0000024$ and P_2 is the term $PW\_0000234$. For our application we simplify this structure and represent it as a graph-triple with subject $PW\_0000024$, object $PW\_0000234$ and predicate $part\_of$. This allows us to see an ontology as a simple graph. We expect that users would like to expand some of the concepts, and explore them at a higher level of granularity. This implies that the ontological context also has to be defined for the global concepts that are surrounding the query term that we started from (cf. Fig. 2). This can be achieved by extending our special purpose database with extra tables. For each concept from the global ontological context we create a concept table and a relation table.

4 Visualization

The "ContextVis" is based on the Java3D API [7]. We have chosen a $3D$ instead of a simple $2D$ representation because this helps to utilize the space efficiently as well as to experience the ontological world created from the merging process in an intuitive way. Relevant visualization functionalities such as zoom, pan and rotate are made available in the "ContextVis" engine. Further interaction is provided through the mouse; i.e., if a concept definition exists in the ontology, it can be shown by clicking the "right" mouse button. The nodes for which the concept and relation tables are created can be expanded and, at each expansion, the local context of concept will be represented. An example of the local and global context visualization is depicted in Fig. 2



(a) Visualization of global ontological context created around "Alzheimer"

(b) Visualization of local ontological context created around "Alzheimer's disease pathway" from NCI-THESAURUS ontology

Fig. 2: Global and Local Context Visualization

5 Conclusion

In this paper we have described our efforts in developing a tool that can visualize the ontological context of some *interesting concept*. With the "ContextVis" tool we can explore how one and the same concept is represented in different ontologies. This brings about the possibility to connect the knowledge from one domain with the knowledge from others. All considered, our processing approach integrates the data around some *interesting concept* and produces a graph structure that can be visualized very well. This simple graph representation has the advantage that we can use graph algorithms for reasoning. We are interested especially in the searching of paths between the concepts in the graph.

References

1. OWL Web Ontology Language Overview: `http://www.w3.org/TR/owl-features/`
2. SPARQL Query Language for RDF: `http://www.w3.org/TR/rdf-sparql-query/`
3. RacerPro an OWL reasoner: `http://www.racer-systems.com/`
4. J. de Bruijn, M. Ehrig, C. Feier, F. Martín-Recuerda, F. Scharffe, M. Weiten. Ontology mediation, merging and alignment. Semantic Web Technologies. Chapter 6. John Wiley & Sons, Ltd, (2006).
5. Jena - A Semantic Web Framework for Java: `http://jena.sourceforge.net/`
6. MySql: The world's most popular open source database: `http://www.mysql.com/`
7. Java 3D API: `http://java.sun.com/products/java-media/3D/`
8. OBO Download Matrix: `http://www.berkeleybop.org/ontologies/`
9. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Pater-Schneider, editors. The Description Logic Handbook. Cambridge University Press, (2002).