# ONTOXPL* – Intelligent Exploration of OWL Ontologies

Volker Haarslev and Ying Lu and Nematollah Shiri
Computer Science Department, Concordia University, Montreal, Canada
{haarslev, ying_lu, shiri}@cs.concordia.ca

## Abstract

*The OWL ontology explorer* ONTOXPL *is based on the web server tomcat. Standard HTML browsers can be used to interact with* ONTOXPL. *It is intended to complement existing ontology editors and does not offer any editing support.* ONTOXPL *uses the OWL DL reasoner* RACER *via its extensive query interface in order to support the intelligent exploration of OWL ontologies.*

## 1. Introduction

The semantic web initiative is based on description logics (DLs) and defines important challenges for current DL system implementations. Recently, one of the main standards for the semantic web has been proposed: the Ontology Web Language (OWL) [12]. Practical description logic systems play an ever-growing role in the context of the semantic web due to the increasing popularity of OWL. State-of-the-art description logic inference systems such as RACER allow for interpreting OWL ontology documents as T-boxes and A-boxes [4]. Racer accepts the OWL DL subset [12] (with the additional restriction of approximated reasoning for so-called nominals). Descriptions of individuals are represented as A-boxes by the RACER System. Viewing the "individuals part" of OWL DL documents as A-boxes provides for query languages supported by DL systems.

User interfaces are very important for practical work with description logic inference systems. An increasing number of graphical interfaces are available for existing DL systems. One class of interfaces consists of ontology editors such as OILED [1] and PROTÉGÉ [9]. With these editors ontologies can be interactively built and stored, for example, as OWL documents. In addition, the editors can be used for describing information about individuals with respect to OWL ontologies. Applications using these OWL documents require an inference engine that supports reasoning about individuals. Indeed, OILED and PROTÉGÉ can be configured to use RACER [3] as an inference engine for classifying ontologies and for answering simple queries about individuals.

Another class of interfaces offers browsing and visualization capabilities. RICE [8] supports the input of textual queries and displays the concept/class hierarchy of T-boxes as outline views as well as the relational structure of A-boxes as directed graphs. The outline view of classes is usually also supported by ontology editors but RICE additionally supports the visualization of A-boxes. Other OWL/RDF visualization tools or editors with visualization capabilities are, e.g., KAON [10], OntoEdit [11], and OntoTrack [6].

The OWL ontology explorer ONTOXPL presented in this paper is intended to complement existing ontology editors and visualization tools. It is completely based on OWL and offers a large variety of information queries. Three potential user groups are targeted by ONTOXPL's design: (i) users with a limited background of ontologies and OWL; (ii) ontology developers that are OWL experts; (iii) users interested in understanding and reusing existing ontologies. ONTOXPL is available as a web server based on the tomcat architecture. Standard HTML browsers can be used to interact with ONTOXPL. Its interface makes heavy use of RACER's extensive query interface in order to support users when exploring OWL ontologies. The following sections give a brief tour on using ONTOXPL and explain its rationale in more detail. Afterwards ONTOXPL is compared with related work. This paper concludes with an outlook to possible future work.

## 2. ONTOXPL's main user interface

ONTOXPL's design is influenced by OWL (and its foundation on DLs). Therefore, it focuses on the three main language elements of OWL, classes/concepts, properties/roles, and individuals/nominals (the OWL/DL vocabulary is used interchangeably in this paper).

The main command pane of ONTOXPL is shown in Figure 1. The filename of the OWL ontology currently loaded into ONTOXPL and RACER is shown with a summary of the

* ONTOXPL's URL: http://www.cs.concordia.ca/ying_lu/

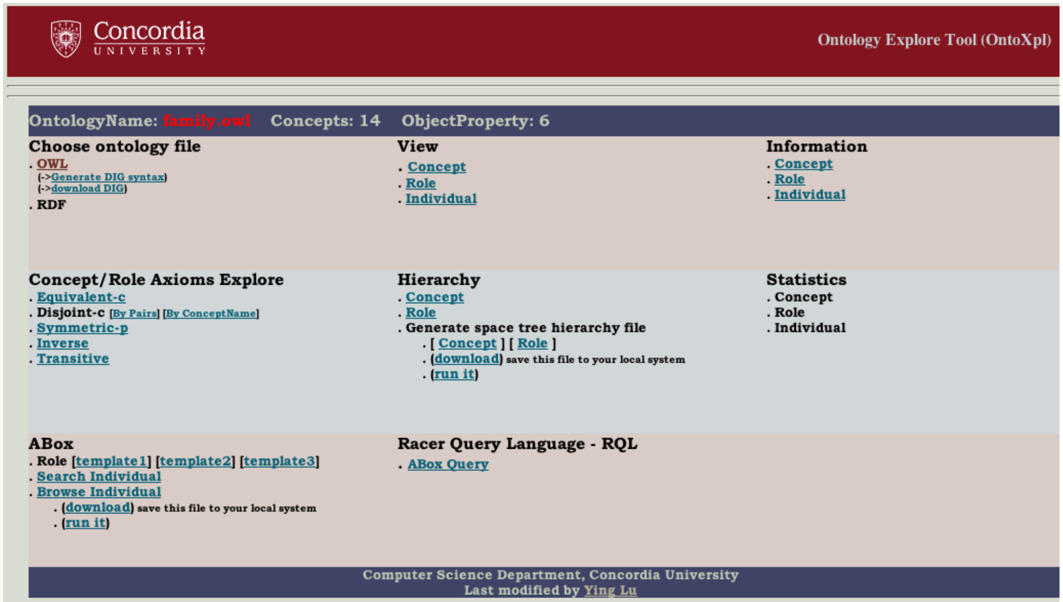



**Figure 1. ONTOXPL's main command pane.**

number of contained concept and role names (see also Section 3 for an explanation of the example knowledge base). ONTOXPL's interface offers eight principal browsing categories (from left to right and top to bottom): file selector, "natural language" description, structural information, exploration of concept/property axioms, inspection of concept and role hierarchies, view of statistical information (not yet implemented), inspection of A-box graph structures, and the interactive use of RACER's query language nRQL. In the following the seven implemented categories are described.

The first horizontal command pane (see Figure 1) offers three groups of commands. The left group of commands is used to load an OWL file and generate a DIG representation of the loaded OWL file. The middle group of commands applies to concepts, roles, and individuals. These commands result in displaying the OWL source code (e.g., see Figure 4) together with a "natural language" description (e.g., see Figure 3). The "natural language" (NL) description is based on the DL notation and tries to describe the selected item w.r.t. this notation. These NL descriptions are intended for users with a limited background on DL and OWL. The information views of concepts (e.g., see Figure 5), roles (e.g., see Figure 7), and individuals (e.g., see Figure 6) use RACER's query interface to display their (inferred) characteristics. Concepts are described by (i) their relative position in the classification hierarchy (e.g., parent, children), (ii) the roles occurring in the concept declarations, and (iii) the individuals that are instances of this concept. By analogy, a role is similarly described but in addition to its position in the role hierarchy, the concepts are listed that use this role. An individual is described by (i) its most specific concept names (so-called types) of which it is an instance, (ii) other individuals that are instances of concepts (parents, children, etc) related to its types.

The command group on the left of the middle pane allows one to query about equivalent or disjoint concept names and symmetric, inverse, and transitive roles. The other group is concerned with concept and role hierarchies. There exist two principal services: one can browse the concept or roles hierarchies in an outline view or a data file for the SpaceTree tool [2] is generated such that graphs of taxonomies can be interactively inspected. Taxonomies are displayed as a pure trees due to a restriction of the SpaceTree tool, i.e., edges to more than one superclass are ignored. The two command groups in the bottom pane are dedicated to explore A-boxes. The first command group has several search forms to retrieve individuals and their known relationships with other individuals, to browse relationships in an outline view or inspect the A-box structure with SpaceTree. The second command group allows users to query A-boxes with RACER's query language nRQL [5].

## 3. Example scenario

The capabilities of ONTOXPL are best explored interactively. However, in this section we try to briefly illustrate some of its main features. Let us assume that ONTOXPL is used to explore an ontology file called "family.owl" describing knowledge about family members (e.g., mother, aunt) and their relationships (e.g., has-child, has-sibling). The structure of the corresponding class hierarchy is shown in Figure 2. From the T-box graph a user might be inter-

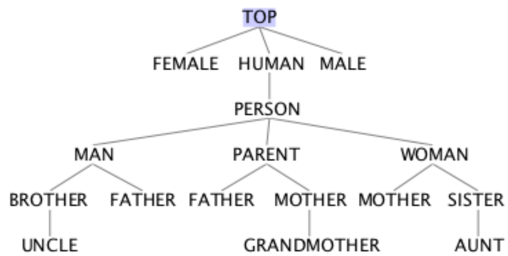


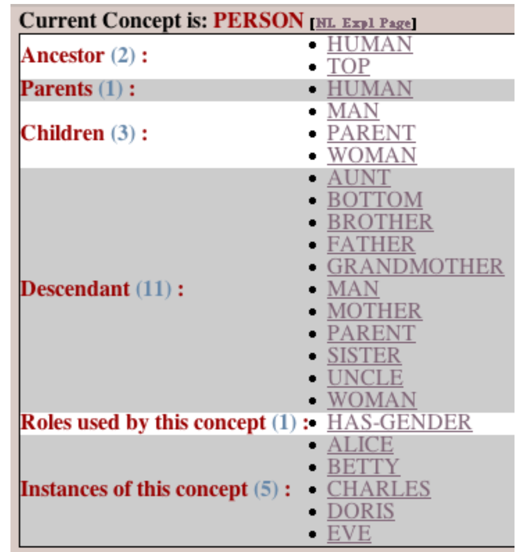

**Figure 2. Class hierarchy of the "family" KB.**

```
It is the anonymous subclass of
    A concept HUMAN
    and
    it has a filler in the role HAS-GENDER
    at least one (or more than one) of its instances is(are):
    A concept FEMALE
            or
    A concept MALE
```

**Figure 3. NL description of class PERSON.**

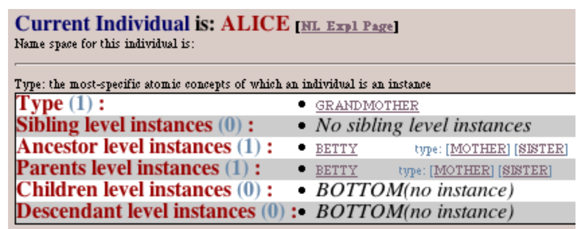
```
(rdfs:subClassOf)
  (owl:Class)
    (owl:intersectionOf rdf:parseType="Collection")
      (owl:Class rdf:about="HUMAN")
      (/owl:Class)
      (owl:Restriction)
        (owl:onProperty rdf:resource="HAS-GENDER")
        (/owl:onProperty)
        (owl:someValuesFrom)
          (owl:Class)
            (owl:unionOf rdf:parseType="Collection")
              (owl:Class rdf:about="FEMALE")
              (/owl:Class)
              (owl:Class rdf:about="MALE")
              (/owl:Class)
            (/owl:unionOf)
          (/owl:Class)
        (/owl:someValuesFrom)
      (/owl:Restriction)
    (/owl:intersectionOf)
  (/owl:Class)
(/rdfs:subClassOf)
```

**Figure 4. OWL code of class PERSON.**



| Current Concept is: PERSON [NL Expl Page] | |
|---|---|
| Ancestor (2) : | • HUMAN<br>• TOP |
| Parents (1) : | • HUMAN |
| Children (3) : | • MAN<br>• PARENT<br>• WOMAN |
| Descendant (11) : | • AUNT<br>• BOTTOM<br>• BROTHER<br>• FATHER<br>• GRANDMOTHER<br>• MAN<br>• MOTHER<br>• PARENT<br>• SISTER<br>• UNCLE<br>• WOMAN |
| Roles used by this concept (1) : | • HAS-GENDER |
| Instances of this concept (5) : | • ALICE<br>• BETTY<br>• CHARLES<br>• DORIS<br>• EVE |

**Figure 5. Taxonomic info about PERSON.**

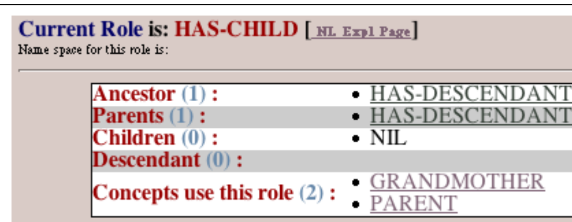


Current Individual is: ALICE [NL Expl Page]
Name space for this individual is:

Type: the most-specific atomic concepts of which an individual is an instance

| | |
|---|---|
| Type (1) : | • GRANDMOTHER |
| Sibling level instances (0) : | • *No sibling level instances* |
| Ancestor level instances (1) : | • BETTY type: [MOTHER] [SISTER] |
| Parents level instances (1) : | • BETTY type: [MOTHER] [SISTER] |
| Children level instances (0) : | • *BOTTOM(no instance)* |
| Descendant level instances (0) : | • *BOTTOM(no instance)* |

**Figure 6. Taxonomic info about ALICE.**



Current Role is: HAS-CHILD [ NL Expl Page]
Name space for this role is:

| | |
|---|---|
| Ancestor (1) : | • HAS-DESCENDANT |
| Parents (1) : | • HAS-DESCENDANT |
| Children (0) : | • NIL |
| Descendant (0) : | |
| Concepts use this role (2) : | • GRANDMOTHER<br>• PARENT |

**Figure 7. Taxonomic info about HAS-CHILD.**

ested in the class PERSON and selects this class for further inspection. Figure 3 shows a "natural language" (NL) description of this class (the underlined names link to the corresponding NL views) while Figure 4 displays its OWL specification (using the XML syntax). The NL and OWL views are directly linked with the corresponding taxonomic views. Figure 5 displays the taxonomic information about the class PERSON retrieved from RACER. It lists ancestors, parents, children, and descendants of PERSON. It also shows the role names used in this class specification and the individuals which are instances of PERSON.

A user might be interested in the individual ALICE. Its taxonomic information is shown in Figure 6, e.g., ALICE is an instance of GRANDMOTHER. This view also lists instances of concepts that are ancestors, parents, siblings, descendants, or children of ALICE's most-specific subsumers (GRANDMOTHER). For instance, BETTY is an instance of these parent classes. Figure 7 shows the taxonomic information about the role HAS-CHILD. The display of inferred information in these windows is intended to help users better understand the structure of the T-boxes and A-boxes.

In addition to the hierarchical views displayed by the SpaceTree tool, ONTOXPL also offers standard outline views for concept and role hierarchies as well as A-box structures. The disadvantage of this type of view is the repeated occurrence of nodes (or subtrees) that have more than one parent. However, the class names occurring in outline views are represented as hyperlinks allowing immediate access to more information.

## 4. Discussion

Currently there do not exist many stable and usable ontology visualization or exploration tools (and even editors). The lack of suitable tools and their shortcomings were one of the major motivations to design and implement ONTOXPL. The motivation for ONTOXPL's web server based architecture was the ease of use with standard HTML browsers and the simple adaptation to multi-user environments. To the best of our knowledge ONTOXPL is currently the only ontology exploration tool that is fully targeted to OWL and relies on RACER's deductive capabilities for offering users better exploration capabilities. A detailed description of ONTOXPL and its architecture as well as a comparison with related work can be found in [7].

Various features of ONTOXPL are also (partly) supported by ontology editors such as PROTÉGÉ [9] and OILED [1] or OWL/RDF visualization tools or editors with visualization capabilities such as KAON [10], OntoEdit [11], and OntoTrack [6]. For instance, PROTÉGÉ also offers users a high-level (DL-like) description of the definition of concept names if the mouse pointer is moved over these names. However, this does not seem to be very suitable for longer concept definitions and does not support the inspection of the occurring OWL elements via hyperlinks. ONTOXPL's "NL description" seems to be more readable and carefully supports the inspection of mentioned entities via hyperlinks. RICE [8] offers visualization facilities for A-boxes where the complete graph structure of A-boxes is displayed. ONTOXPL is restricted to a tree-like approximation due to the underlying SpaceTree tool [2] but it works better for larger A-boxes.

In our experience, ONTOXPL's cross-referencing capabilities for hyperlinked concept, role, and individual names help users comprehend unknown ontologies faster than with the support offered by traditional editors.

## 5. Conclusion

In this paper we briefly introduced ONTOXPL, a first step toward an OWL ontology exploration tool. ONTOXPL is intended to complement ontology editors or other ontology visualization tools. A recently conducted informal experiment, where about 40 students had to design and implement 15 different OWL ontologies with a size of several hundred concept names, demonstrated that ONTOXPL provides helpful information about ontologies that is otherwise not as easily available in ontology editors such as PROTÉGÉ or OILED. The implementation of the statistics command group is underway. It is also planned to integrate query results from nRQL such that individuals names are recognized as hyperlinks. Another important issue is the optimization of ONTOXPL performance for larger ontologies containing thousands of concept names.

## References


[1] S. Bechhofer, I. Horrocks, and C. Goble. OilEd: a reasonable ontology editor for the semantic web. In *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna*. LNAI Vol. 2174, Springer-Verlag, 2001.

[2] J. Grosjean, C. Plaisant, and B. Bederson. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Procedings of IEEE Symposium on Information Visualization*, pages 57–64, Boston, USA, October 2002.

[3] V. Haarslev and R. Möller. RACER system description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, 2001.

[4] V. Haarslev and R. Möller. Optimization techniques for retrieving resources described in OWL/RDF documents: First results. In *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*, June 2004.

[5] V. Haarslev, R. Möller, R. V. D. Straeten, and M. Wessel. Extended query facilities for Racer and an application to software-engineering problems. In *Proceedings of the International Workshop on Description Logics (DL-2004), Whistler, BC, Canada*, June 2004.

[6] T. Liebig and O. Noppens. OntoTrack: Fast browsing and easy editing of large ontologies. In *Proceedings of The Second International Workshop on Evaluation of Ontology-based Tools (EON2003), located at ISWC03*, Sanibel Island, USA, October 2003.

[7] Y. Lu. Exploration of OWL ontologies. Master's thesis, Department of Computer Science, Concordia University, Montreal, 2004 (in preparation).

[8] R. Möller, R. Cornet, and V. Haarslev. Graphical interfaces for Racer: querying DAML+OIL and RDF documents. In *Proc. International Workshop on Description Logics – DL'03*, 2003.

[9] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with Protege-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[10] D. Oberle, R. Volz, B. Motik, and S. Staab. An extensible ontology software environment. In *Handbook on Ontologies*, International Handbooks on Information Systems, chapter III, pages 311–333. Steffen Staab and Rudi Studer, Eds., Springer, 2004.

[11] Y. Sure, J. Angele, and S. Staab. OntoEdit: multifaceted inferencing for ontology engineering. *Journal on Data Semantics*, 2800/2003:128–152, 2003.

[12] F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL web ontology language reference, http://www.w3.org/tr/owl-guide/, 2003.