

# GrOWL: A tool for visualization and editing of OWL ontologies

Sergey Krivov<sup>a,c,\*</sup>, Richard Williams<sup>d,1</sup>, Ferdinando Villa<sup>b,c</sup>

<sup>a</sup> Department of Computer Science, The University of Vermont, Burlington, VT 05405-0088, USA

<sup>b</sup> The Botany Department, The University of Vermont, Burlington, VT 05405-0088, USA

<sup>c</sup> Gund Institute for Ecological Economics, The University of Vermont, 617 Main Street, Burlington, VT 05405-0088, USA

<sup>d</sup> Rocky Mountain Biological Laboratory, P.O. Box 519, Crested Butte, CO 81224, USA

Received 10 January 2005; accepted 29 March 2007

Available online 7 April 2007

## Abstract

In an effort to optimize visualization and editing of OWL ontologies we have developed GrOWL—a browser and visual editor for OWL that accurately visualizes the underlying DL semantics of OWL ontologies while avoiding the difficulties of the verbose OWL syntax. In this paper, we discuss GrOWL visualization model and the essential visualization techniques implemented in GrOWL.

© 2007 Elsevier B.V. All rights reserved.

**Keywords:** Ontology visualization; OWL; Semantic networks

## 1. Introduction

In an effort to optimize visualization and editing of OWL ontologies we have developed GrOWL a new OWL visualization tool [1] based around the underlying DL semantics of OWL ontologies. GrOWL has advanced browsing and navigation tools not available in other OWL visualization frameworks and also supports visual editing.

### 1.1. Implementation

The current prototype of GrOWL is based on the Prefuse library [2,3]. GrOWL is implemented as a Java applet, as a Protégé plugin and as a stand-alone Java application (see Fig. 1). The policy-based design [4] of GrOWL ensures its extensibility and the possibility to easily adapt the software to the needs of different projects. The GrOWL editor uses Jena API, and the GrOWL applet uses a trimmed version of the WonderWeb OWLAPI. The current implementation of GrOWL editor is open source and is available for download at: <http://www.uvm.edu/~skrivov/growl/>.

### 1.2. Usage

GrOWL has been used in the Ecosystem Services Database (ESD) [5,6] a data and analysis portal to assist the informed estimation of the economic values of ecosystem services. The ESD uses the applet version of GrOWL to present the annotated OWL ontologies to the user. Once the user has located a concept of interest, the right-click menu in the applet gives access to a set of queries that the database supports for the selected concept. The second important application of GrOWL is to enable the development of ontologies in a workshop or seminar context. The annotated graphical description provided by GrOWL has proved to be a valuable learning device for students in our bioinformatics classes.

In this paper, we discuss the GrOWL visualization model and related visualization techniques. In Section 2, we introduce the GrOWL visualization model. In Section 3, we describe the visualization techniques used in GrOWL. Future work is discussed in Section 4.

## 2. GrOWL visualization model

### 2.1. Graphical coding

The first principle behind GrOWL is the use of the color, shading and shape of nodes to encode properties of the basic language constructs.

\* Corresponding author at: Department of Computer Science, The University of Vermont, 617 Main Street, Burlington, VT 05405-0088, USA.  
Tel.: +1 802 656 0380.

E-mail address: [Serguei.Krivov@uvm.edu](mailto:Serguei.Krivov@uvm.edu) (S. Krivov).

<sup>1</sup> Microsoft Research Ltd., 7 J J Thomson Ave, Cambridge CB3 0FB, UK.

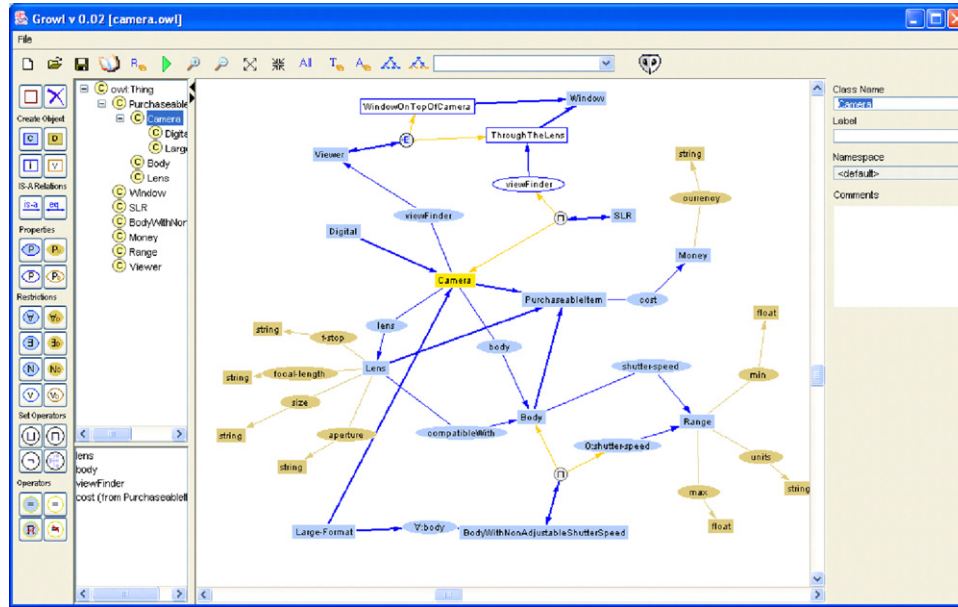


Fig. 1. GrOWL in editing mode.

## 2.2. ABox mapping

We presume that the reader is familiar with DL semantics and Abstract Syntax of OWL [7]. We also presume familiarity with basic DL terminology.

Consider the following ABox:

```
Individual (JohnSmith
  type (AssociateProfessor)
  value (teaches Java)
  value (teaches ArtificialIntelligence)
  value (age "32"^^xsd:integer))
```

The graphical mapping of these ABox is shown in Fig. 2.

Figs. 3 and 4 provide simple examples that illustrate idioms for axioms SameIndividual and DifferentIndividuals.

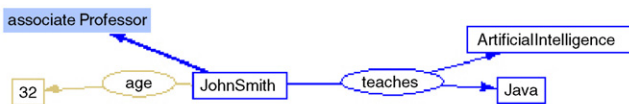


Fig. 2. Graphic idioms for assertions about individuals.



Fig. 3. Representation of axiom SameIndividual (AI ArtificialIntelligence).



Fig. 4. Representation of axiom DifferentIndividuals (JohnWSmith JohnSmith).

## 2.3. TBox mapping

GrOWL TBox mapping was inspired by the Domain Maps introduced in Ref. [8]. The mapping is defined by two mutually recursive functions. The first function is the structural mapping  $G$  that maps every definition of class  $C$  from the OWL knowledge base into a graph  $G(C)$ . Only one node of the graph  $G(C)$  represents the mapped class  $C$ . Thus, the second function is base node mapping  $BN$  that draws correspondence between a class definition and the single node  $BN(C)$  of graph  $G(C)$ . The node  $BN(C)$  (base node of  $C$ ) represents class  $C$  in the sense that every arrow representing subclass of and equal-to relations of a class is attached to the base node of this class. Thus, for any pair of either named or anonymous classes  $C1$  and  $C2$  the following holds:

- $G$  maps  $C1 \sqsubseteq C2$  ( $C1$  is subclass of  $C2$ ) to the diagram in Fig. 5.
- $G$  maps  $C1 = C2$  (which is equivalent to  $C1 \sqsubseteq C2$  and  $C2 \sqsubseteq C1$ ) to the diagram in Fig. 6.
- $G$  maps ABox expression  $a: C1$  ( $a$  is instance of  $C1$ ) to the diagram in Fig. 7.

In Figs. 5–7 and in Table 1, the node labeled  $BN(C1)$  is a base node of class  $C1$  and the node labeled  $BN(C2)$  is a base node of class  $C2$ . They do not have to be shaded squares, but may be of

Fig. 5. Graph  $G(C1 \sqsubseteq C2)$ .Fig. 6. Graph  $G(C1 = C2)$ .



Fig. 7. Graph  $G(a: C_1)$ .

Table 1  
Recursive mapping of DL class constructors

Definition of class $C$	The diagram $G(C)$	Base node $BN(C)$
Named Class $C$		
Intersection $C_1 \sqcap C_2$		
Union $C_1 \sqcup C_2$		
Complement $\neg C_1$		
Enumeration $\{o_1, o_2\}$		
Exist Restriction $\exists R.C_1$		
For all Restriction $\forall R.C_1$		
Number Restriction $\geq nR$	Eg.	
Number Restriction $\leq nR$	Eg.	
Value Restriction $R \sqsubseteq o$		

any shape permissible for base nodes shown in the third column in Table 1. Table 1 describes recursive mapping of OWL class constructors into a graph that constitutes the core of GrOWL. This mapping is created by functions  $G$  and  $BN$  described in second and the third column of the table, respectively. Data properties and data property restrictions are not shown in the table since their mapping is identical to the mapping of the respective relations pertaining to individuals, and the difference is only in color.

Fig. 6 describes the mapping of the “subclass of” axiom; the mapping of remaining OWL class axioms is described in Fig. 8.

Structural mapping follows the recursive definitions of the OWL semantic constructs, and therefore every TBox construct receives a graphic representation. Structural mapping is one-to-one.

#### 2.4. Properties

Most of the GrOWL idioms can be obtained from the structural mapping of a representation of an OWL file in form of a DL Knowledge Base, but property declarations are an exception. A property is a binary relation. Fig. 9 shows a graphic idiom for a simple specification of an object property:

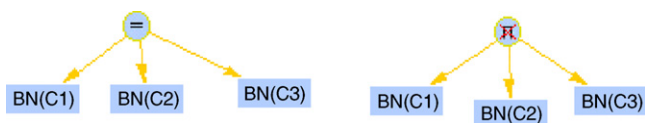


Fig. 8. Mapping of OWL class axioms EquivalentClasses ( $C_1 C_2 C_3$ ) and DisjointClasses ( $C_1 C_2 C_3$ ).

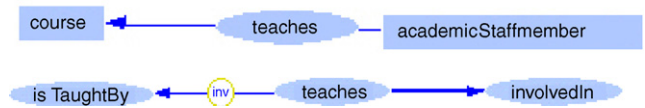


Fig. 9. The separate diagrams generated by the object property specification.

```
ObjectProperty (teaches super (involvedIn)
domain (academicStaffMember
range (course)
inverseOf (isTaughtBy) )
```

This specification of an object property will generate two separate diagrams as it is shown in Fig. 9.

The second diagram in Fig. 9 appears in the RBox view only. The separation of the TBox, ABox and RBox views is especially convenient in viewing large ontologies.

### 3. Visualization techniques

The following techniques were used in the first prototype of GrOWL:

- *Automatic and manual layout.* There are various algorithms for automatic graph layout. A very interesting one is animated force directed layout, which performs the layout interactively in combination with animation. GrOWL also supports a manual layout to allow the user to reposition individual nodes as they wish.
- *TBox, ABox, RBox and class hierarchy views.* There are several meaningful ways to filter out a part of the entire graph, including viewing only the ABox, TBox or class hierarchy. Such filtering may not require selection of a focal node.
- *Locality restricted view.* An attempt at drawing even a simple semantic network on a two-dimensional surface would likely end up with a non-planar graph – meaning that the relationships that link concepts together would be drawn crossing and overlapping each other – a very messy and potentially unusable view of the underlying structure. It is possible to provide a mechanism for restricting the set of visible nodes and edges to only those that lie in a topological neighborhood of a selected node. When the selected node is changed the set of visible nodes and edges is readjusted accordingly. A node may be set as selected by clicking on it or by obtaining it as a result of one or another search and navigation method.
- *Tree based navigation.* The list of classes and instances occurring in the ontologies may be given in a class tree familiar to the users of Protégé. Selecting any object from this tree sets the respective node as selected. This action readjusts the visible locality of the network, centering it on the selected node.

These techniques have proved to be efficient in visualization of small ontologies. However, in visualization of large ontologies even *Locality Restricted View* does not guarantee clarity. Using user feedback we implemented the following visualization techniques:

- **Filtering.** GrOWL provides a mechanism for restricting the view to only class definition, the subclasses, the super-classes, or all instances associated with a selected node. This filtering mechanism is complementary to the topological neighborhood mechanism. Filters make the presentation meaningful by allowing the user to focus precisely on what is of interest. The filter that outputs the class definition for the selected class node has proven to be specially useful.
- **Prefix search.** Users can search for the nodes (classes and properties) using incremental matching. While user types the name of the node the information about the set of nodes with the given prefix is displayed. Selecting one of the nodes brings it into focus and the visible locality of the network is readjusted using the current filter.

#### 4. Conclusion

In this article we discussed visualization models for OWL and the visualization techniques implemented in the GrOWL prototype. GrOWL visualization model is optimal in terms of simplicity and completeness. GrOWL is the only tool that gets a good score in a simple test we propose:

*Provide a readable rendering of all or almost all elements of Roger L. Costello's camera ontology on 640 by 800 canvas.*

GrOWL uses advanced visualization techniques including animated force directed layout, interactive locality restricted viewing and filtering. Initially we were concerned about possible performance bottlenecks that the force directed layout algorithm and filters may impose. Yet the current GrOWL prototype can handle gracefully the visualization of the 1620 KB FungalWeb Ontology [9], where the only visible problem was with the load time. Use of a database backend for storage of the graph will further improve performance and resolve the problem with the load time. We plan to add this in future development, and the Prefuse library has support for this.

Browsing of large ontologies was problematic in the initial version of the prototype. However, the filter mechanism effectively solved the major problems. Based on our users' experience, we believe that a combination of filtering and navigation techniques is the most powerful and useful visualization techniques.

#### 4.1. Future work

In the future, we plan to extend GrOWL by implementing graphic maps—a feature that will provide domain specific rendering for instances of certain classes. The major strength of the GrOWL visualization model is its affinity with DL semantics. This affinity allows a natural extension of the model for presentation of queries over DL ontologies [1] and visualization of SWRL.

#### Acknowledgments

The development of ESD has been supported by NSF grants 9982938/0243957. The work on GrOWL has been supported by NSF sponsored project 0225676 - Science Environment for Ecological Knowledge (SEEK). GrOWL evolved through the discussions within SEEK community and we owe a lot to every member of seek-kr-sms discussion group. We are specially grateful to Mark Schildhauer who had motivated us to develop visual representation of OWL.

#### References

- [1] S. Krivov, F. Villa, Towards the paradigm of an ontology based user interface: from simple knowledge maps to graphical query language, in: B. Ludascher, L. Raschid (Eds.), Data Integration in Life Sciences, Springer, 2005, Lecture Notes in Bioinformatics 3615.
- [2] J. Heer, S.K. Card, J.A. Landay, Prefuse: a toolkit for interactive information visualization, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '05, ACM Press, Portland, OR, USA/New York, NY, USA, 2005, pp. 421–430.
- [3] Prefuse (URL: <http://prefuse.sourceforge.net/index.html>).
- [4] A. Alexandrescu, Modern C++ Design: Generic Programming and Design Patterns Applied, Addison-Wesley/Longman Publishing Co., Inc., 2001.
- [5] F. Villa, M.A. Wilson, R. DeGroot, S. Farber, R. Costanza, R. Boumans, Design of an integrated knowledge base to support ecosystem services valuation, Ecol. Econ. 41 (2002) 445–456.
- [6] Ecosystem services database (URL: <http://esd.uvm.edu>).
- [7] P.F. Patel-Schneider, P. Hayes, I. Horrocks, OWL web ontology language semantics and abstract syntax, W3C recommendation, URL: <http://www.w3.org/TR/owl-semantics/>, 2004.
- [8] B. Ludäscher, A. Gupta, M.E. Martone, Model-based mediation with domain maps, in: Proceedings of the 17th International Conference on Data Engineering ICDE, IEEE Computer Society, Heidelberg, Germany, 2001.
- [9] A. Shaban-Nejad, C.J.O. Baker, V. Haarslev, G. Butler, The FungalWeb Ontology: Semantic Web Challenges in Bioinformatics and Genomics, in: Proceedings of 4th International Semantic Web Conference (ISWC), Springer 2005, Lecture Notes in Computer Science, vol. 3729.