

OntoGui: a Graphical User Interface for Rapid Instantiation of OWL Ontologies

Walter TERKAJ^{a,1}

^a*Institute of Industrial Technologies and Automation (ITIA-CNR), Milan, Italy*

Abstract. The efficient instantiation of an OWL ontology still represents one of the barriers towards their extensive use in industrial applications not limited to the definition of more or less complex T-boxes. This paper presents the prototype non-commercial ontology-based software tool named OntoGui that can be employed in the management and instantiation of A-box modules, mainly supporting the validation of T-box modules and the rapid generation of data sets.

Keywords. Graphical User Interface, Ontology, OWL, A-box Generation

1. Introduction

As ontology-based approaches are becoming more and more popular in the scientific and industrial community, there is an increasing need of software tools supporting the development and instantiation of an ontology. Most of the general purpose tools focus mainly on the development of a T-box, whereas the instantiation and enrichment of the A-box is poorly supported. Even Protégé² [2], one of the most adopted ontology editors, can be hardly used to manage individuals and their relations as soon as the number of axioms defined in the T-box grows. Other tools like NeOn toolkit³ and in particular TopBraid Composer⁴ better support the generation of relations between individuals by suggesting which object and datatype properties can be used according to their ranges and the restrictions assigned to OWL classes. However, even in these cases, a safe and quick instantiation of ontologies is not fully supported because it is difficult to navigate through the A-box and consistency check functionalities are missing. The proposed OntoGui tool⁵ mainly aims at supporting:

- The fast evaluation of a T-box under development by concurrently instantiating a corresponding A-box, thus implementing a kind of test-driven development approach.
- The generation of RDF data sets to be used as input for other ontology-based applications, without needing customized graphical user interfaces or data converters.

¹Corresponding Author: Institute of Industrial Technologies and Automation (ITIA-CNR), Milan, Italy; E-mail: walter.terkaj@itia.cnr.it

²<http://protege.stanford.edu/>

³http://neon-toolkit.org/wiki/Main_Page/

⁴<http://www.topquadrant.com/tools/IDE-topbraid-composer-maestro-edition/>

⁵<http://www.terkaj.com/tools.html#OntoGui>

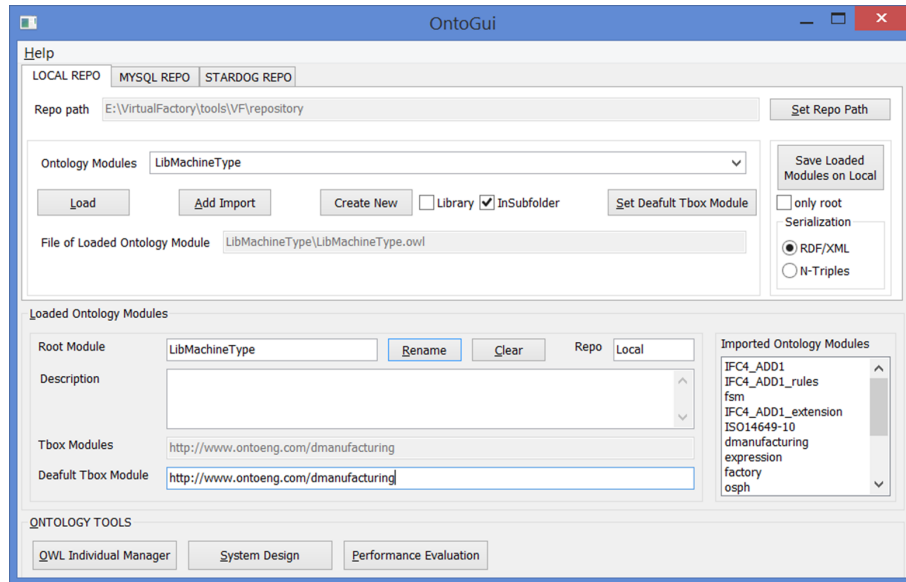


Figure 1. OntoGui: Control Panel. The ontology module `LibMachineType` is loaded as an example

2. Design and Functionalities

OntoGui is a graphical user interface developed as a desktop application in C++ making use of `wxWidgets Cross-Platform GUI Library`⁶ for the creation of graphical elements, and the `RdfCpp` library. `RdfCpp` is a C++ library, based on `Boost Library`⁷ and `Redland RDF`⁸ (enabling the parsing and generation of RDF triples), that provides classes and functions to manage a network of RDF graphs, parse and generate OWL individuals, parse OWL axioms of a T-box (i.e., equivalent classes; subclasses; restrictions of any degree involving universal quantifier, existential quantifier, or cardinality constraints; domain and range of properties). Moreover, `RdfCpp` supports the connection with three RDF store solutions: (a) file-based; (b) `MySQL`⁹ relational database; (c) `Stardog`¹⁰ triplestore.

The main window of OntoGui is a Control Panel (Figure 1) that can manage (networks of) ontologies in a file-based repository or other more scalable RDF stores (i.e., `MySQL`-based repository and `Stardog`) by selecting different *repo* tabs. The Control Panel allows to load an existing ontology module and its dependencies, create a new A-box module, define new import relations between modules, and save the modules in any of the available repository. The Control Panel provides also access to a set of *ontology tools*: 1) `OWL Individual Manager`, 2) `System Design`, and 3) `Performance Evaluation`.

OWL Individual Manager is a general purpose tool for the management of OWL individuals. The main window of the tool (Figure 2) is dynamically reconfigured every time an OWL class belonging to the available T-box is selected (in the top left cor-

⁶<https://www.wxwidgets.org/>

⁷<http://www.boost.org/>

⁸<http://librdf.org/>

⁹<https://www.mysql.com/>

¹⁰<http://www.stardog.com/>

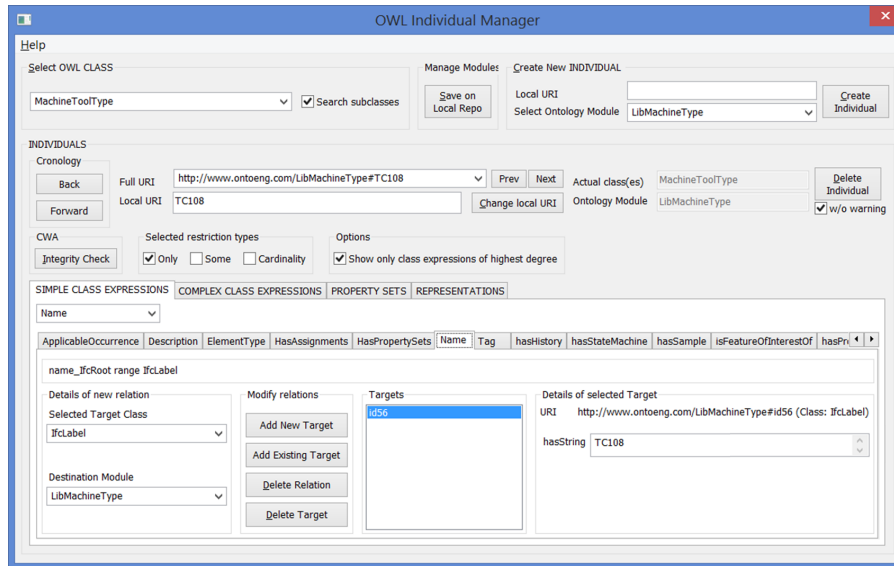


Figure 2. OntoGui: OWL Individual Manager

ner). After loading an ontology module in the control panel and selecting an OWL class in OWL Individual Manager, the characterization of the OWL classes provided by the RdfCpp library enables the following functionalities:

- Generation and listing of individuals belonging to the selected class.
- Listing the properties that can have the selected individual as a subject. Exploring and setting the target value of a property for the selected individual.
- Checking the integrity of the selected individual by interpreting the OWL axioms as Integrity Constraints according to the Closed World Assumption (CWA) [1].

The basic functionalities of OWL Individual Manager are provided for any T-box. However, it is possible to add customizations for specific T-boxes. In particular, further functionalities are available if the T-box includes the ifcOWL ontology [3], i.e., the OWL version of the Industry Foundation Classes that is a key reference standard in the Building Information Modeling (BIM) domain. The ifcOWL-based functionalities include:

- Customized windows for the characterization of an IfcProduct individual in terms of 3D placement and shape representations (i.e., a bounding box or a linked binary file).
- The aggregation structure of an IfcTypeObject individual is automatically replicated for its typed IfcObject individuals.
- Visualization and specification of values for pre-defined property sets.

System Design is a tool for manufacturing applications that enables to populate libraries with reusable information related to: part types to be produced and the process plan needed to produce it; decomposition of a process plan into process steps characterized by processing times and precedence relationships; assignment of a process step to one or more production resources; definition of a production system in terms of connected machines (with failure modes) and buffers (with capacity).

The definition of a production system is exploited by the **Performance Evaluation** tool to evaluate the system performance against a production plan that is specified with a customized interface. The evaluation can be performed via mathematical methods or discrete event simulation that are linked to OntoGui thanks to software connectors.

3. Applications

OntoGui has been employed during computer lab classes of the Master level courses Reconfigurable Manufacturing Systems and Production for Made in Italy Lab at Politecnico di Milano, and during a training course within the national project Smart Manufacturing 2020¹¹. It was demonstrated that OntoGui supports the rapid modeling of production systems even if it is used by engineering students with little or no knowledge about ontologies. Furthermore, OntoGui was exploited by several research projects in the manufacturing domain, such as Pro2Evo¹² and ProRegio¹³. OntoGui enabled the generation of ontology modules needed as input of other ontology-based tools supporting 3D layout design, discrete event simulation, factory monitoring, and process simulation.

4. Demo and Software/Hardware Requirements

OntoGui works in Windows operating systems (XP or above) endowed with MVC++2010 Redistributable Package. A minimum amount of free memory (1 GB) and disk space (1 GB) must be available. The tool is demoed¹⁴ by showing how catalogs of production plans and production resources are generated to support the design of production systems [4]. The academic version of OntoGui can be shared upon request to the author.

Acknowledgments

This work was partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 636966 (ProRegio) and by the Italian research project Smart Manufacturing 2020 within the ClusterTecnologico Nazionale Fabbrica Intelligente.

References

- [1] R. J. Brachman and H. J. Levesque. Chapter 11 - defaults. In R. J. Brachman, , and H. J. Levesque, editors, *Knowledge Representation and Reasoning*, The Morgan Kaufmann Series in Artificial Intelligence, pages 205 – 235. Morgan Kaufmann, San Francisco, 2004.
- [2] M. A. Musen. The protégé project: A look back and a look forward. *AI Matters*, 1(4):4–12, June 2015.
- [3] P. Pauwels, T. Krijnen, W. Terkaj, and J. Beetz. Enhancing the ifcOWL ontology with an alternative representation for geometric data. *Automation in Construction*, 80:77 – 94, 2017.
- [4] W. Terkaj and G. P. Viganò. Semantic GIOVE-VF: an Ontology-based Virtual Factory Tool. In *Proceedings of the Workshop on Data meets Applied Ontologies*, 2017.

¹¹<http://www.fabbricaintelligente.it/en/progetti/>

¹²<http://www.fabbricadelfuturo-fdf.it/progetti/sottoprogetto-2/progetto-pro2evo/>

¹³<http://www.h2020-proregio.eu/>

¹⁴<https://www.youtube.com/watch?v=1Kpwa79ooqE>