ONTOLOGYJAM
A Tool for Ontology Reuse

Luis Fernando Piasseski, Cesar Augusto Tacla, Milton Borsato
Department of Information Technology, Federal University of Technology in Paraná,
Av. 7 de setembro 3165, Curitiba-PR, Brazil
lui_fp@yahoo.com.br, tacla@utfpr.edu.br, borsato@utfpr.edu.br

Keywords: Ontology, Research, Knowledge Management, Re-utilization.

Abstract: There has been notable growth in the use of ontologies in knowledge management. This occurs because, with the use of ontologies, knowledge is shared and re-utilized efficiently and clearly among all the resources, whether a person or an application. However, for ontologies to establish confidence within an extremely competitive and flexible market, which is currently the case, they must be created in a way that is swift and with high credibility, portability and scalability. However, there is a noted lack of tools to aid knowledge specialists for the construction of a new ontology. For this purpose, this article presents a tool that enables the research of concepts in the knowledge entities represented in a given ontology, by importing multiple ontologies. As a result, knowledge can be exported and modeled into a new ontology. This project is based on the re-utilization of knowledge, with the aim of extending an ontology so as to make it adequate to its application.

1 INTRODUCTION

Methodologies for the construction of ontologies are used in various studies and the fact that a standardized methodology does not exist contributes to this diversity. As examples of such diversity, we mention Methontology (GOMEZ-PÉREZ et al., 1997), On-To-Knowledge Methontology (OTKM) (SURE et al., 2001), by Uschold & King (Uschold and Gruninger, 1996), by Gruninger & Fox (Gruninger and Fox, 1995). Methontology and On-To-Knowledge Methodology recommend that knowledge specialists perform research of concepts in other ontologies with the purpose of reutilizing those that are in accordance with the domain researched. This avoids re-work and prioritizes the sharing of knowledge.

For OTKM (SURE et al., 2001), the “Viability Study” and the “Kickoff” are the phases during which the knowledge specialist should do research into other ontologies. In the first, the research is conducted to evaluate the feasibility of the ontology to be created. For the second, the specialist should evaluate other ontologies with the purpose of reutilizing concepts already defined for the creation of a list of concepts that make up the new ontology. These concepts are utilized in the “Refining” phase.

In the Methontology (GOMEZ-PÉREZ et al., 1997), “Specification” and “Integration” are phases in which the re-utilization of concepts should take place. “Specification” can be compared to the “Kickoff” phase of OTKM. “Integration” is the phase that explicitly deals with the reutilization of concepts; this occurs through a semantic and contextual analysis of concepts already defined in other ontologies in order to reutilize them.

The goal of this paper is to describe a tool that can aid knowledge specialists in activities of construction of ontologies, and this tool is called OntologyJam. Figure 1 shows the life cycle for the development of each methodology, the phases that reuse is very important in a given methodology are highlighted. On the other hand, OntologyJam can be used in all phases that are underlined.

The need to re-utilize knowledge results in a laborious activity for the knowledge engineer, as currently this is done manually, that is, the ontology must be scanned and analyzed concept by concept using tools such as Protégé, that are not specific to the activity. Thus, in many cases, the reuse phase is not performed.

There are several techniques to reuse resources of ontology, for example, ontology matching techniques are based on semantic and syntactic
analysis to identify similar concepts and properties. However, matching techniques are not enough when the knowledge specialist want to include a new concept into an ontology. Matching is not recommended because the goal is to include a new concept and not to match a concept under definition.

2 METHODOLOGY

This section presents the methodology applied to build OntologyJam.

2.1 Stages

A technical-theoretical approach has been applied to carry out the project, that is, in the first place, a theoretical basis was sought with regards to ontologies, OWL, Jena, Protégé, as well as a survey of requirements. Subsequently, development of the tool was initiated. The project was divided into stages as described below.

2.1.1 Search for Information

This stage had the objective of understanding how the “tags” available in the OWL language manage the information contained in an ontology, as well as which methods and classes Jena offers to manipulate this information. Protégé was used to aid in this stage.

2.1.2 Prototype

This phase was the first deliverable of the project, and took place with a prototype presented to those interested in the project. Its performance was evaluated for each functionality defined in the requirement specification phase. The improvements to the tool were also enumerated.

2.1.3 Development

This stage was divided into two sub-stages. The first had the objective of analysing the best way to develop the user interface, as well as an analysis of the system. The second consisted of programming the source code.

For the first phase, a navigation flow was created based on Use Cases, as presented in figure 2. Based on this flow, how many and which screens should be used in the system were defined.

With the objective of developing a scalable system, it was determined that each screen should be treated individually, but integrated to the overall interface. For this to be possible, each screen was treated as an “Internal Frame”, and each frame was inserted into the OntologyJam work area; through activation of the corresponding buttons, or links, they can be treated independently by the user.

2.2 Tests

After implementing OntologyJam and performing unit tests, usability tests and functional tests were executed with some selected users. For this, two test cases were formulated, one for the usability test and the other for the functional test.

First of all, the tool was introduced to the certification team by presenting its functionalities, with the purpose of explaining the navigation between screens, how to import and how to create a new ontology. Manuals were delivered, containing a
description of functionalities exemplified by the respective screens, the test cases and an evaluation questionnaire. The certification team was composed of four people and had two weeks to perform the tests and fill in the questionnaire. The questionnaire has open questions about system’s functionalities that could be scored between zero (bad) and five (excellent).

3 ONTOLOGYJAM

3.1 Architecture

In order to develop the software, the Java programming language was adopted, mainly because it offers portability and scalability, and also due to its integration with the Jena API. From the point of view of performance of the application, development in Java Swing was chosen for the interface, thus, processing occurs locally. In order to create the interface, framework JSR-296 was used.

OntologyJam supports ontologies in OWL language only. Therefore, for research in Swoogle a prior check is made as to the type of ontology, which is possible through the field hasFiletype, returned by Swoogle. Ontologies that are not in conformity with the reading standard for OntologyJam are informed to the user.

For the research functionality, a research method was used based on simple text comparisons, only differentiating between upper-case and lower-case letters, without taking into account the semantics of the terms. A general view of the tool is given in Figure 3.

3.1.1 Use Case ‘Import’

This Use Case covers the import of ontologies in OWL. This functionality serves to import ontologies in OWL that are posted on the internet, by their URI address, or locally. These ontologies serve as basis for conducting research on concepts.

3.1.2 Use Case ‘Research’

This Use Case covers the research of entities in the ontologies given in the list of ontologies imported by the system. This functionality serves to specify the way in which the research will be conducted, which techniques will be used by the system, as well as the interaction with the user for the success of the research.

3.1.3 Use Case ‘Export’

This Use Case covers the export of the newly built ontology re-utilizing the concepts, priorities and characteristics of the imported ontologies. This Use Case serves to specify the functionalities to be displayed in the option to export the constructed ontology.

3.1.4 Use Case ‘Swoogle Research’

This Use Case covers the execution of a survey using the Swoogle ontology database, which, in turn, uses semantic techniques in its research. This functionality serves to conduct research on terms, entities or key words in multiple ontologies posted in the Swoogle database.

3.2 System Interface

OntologyJam was developed with an interface that is easy to understand, with links that aid in navigability between screens, making the process more agile for creating new ontologies, which makes OntologyJam an agile and user-friendly application.

Figure 3: Overview – Use Case of OntologyJam.

Figure 4: Initial Frame.
Figure 4 presents the initial screen, including the work area, where other system screens will be opened. The tool bar contains icons for the tool functionalities, but it can be deactivated using the menu view, to expand the work area.

The Swoogle research screen, presented in figure 5, enables the user to conduct semantic research in ontologies using the Swoogle search mechanism, with the enormous advantage that, with one click, the user can import the desired ontology.

The ontology import screen, presented in figure 6, manages the ontologies imported by the user, either through Swoogle, or remotely, with the URI address, or located on the local Hard Disk. This screen shows the URI address, Label, Comment and Version of the imported ontology.

The search screen, presented in figure 7, allows the user to conduct research in the previously imported ontologies. The result is divided by ontologies, classes and properties in which the term was found. When a concept, property or ontology is selected, several details of the selected item are presented. Activating the option “Add to Favorites”, the concept is added to the folder “Favorite Classes”. This way, the user can select the main concepts that he/she considers necessary.

The export screen, depicted in figure 8, is activated using the ontology link on the search screen. On this screen, the user can set up the taxonomic tree of the new ontology based on the taxonomic tree of the selected concept.

When the option “Save in OWL Export Tree” is activated using the corresponding icon in the previous screen, the export option screen is shown, as presented in figure 9. From this screen it is possible to select the options for the export of the created ontology.
In addition to the functionalities described above, OntologyJam also offers the option of saving the state of the work area. When this option is activated, the links of the imported ontologies and the current state of the taxonomic trees on the export screen are stored. This structure is saved in an extension of OntologyJam itself.

Figure 10 shows a general view and the advantages of using “Internal Frames” can be observed. Thus, the user navigates between OntologyJam screens, or frames, in the manner deemed most convenient.

4 RESULTS

4.1 Evaluation of functionalities

From data collected of questionnaires answered by 4 users with good knowledge in ontologies, we have obtained the following results for the system’s functionalities evaluation.

Swoogle search: it scored 4.5 and users suggested that visualizing searched ontologies would improve the functionality;

Ontology importing: it scored 4.5 and users suggested to include an option to cancel the importing when an ontology is too big and network communication rate is low.

Research: it scored 3.75 and users suggest filtering searches by ontology when there are a lot of already imported ontologies.

Export: it scored 3.88 and users suggest verifying if a concept to be exported already exists in the “export tree” avoiding duplicated concepts. Users believe the system should show graphically one or more ontology hierarchical structures allowing simultaneous handling. Drag and drop interface commands to transfer concepts from one tree to another were also suggested.

System in general: it scored 4.38 and users, aside interface modification suggestions, considered functionalities are simple to learn and to use. They believe OntologyJam can reduce time when executing the reuse phase.

4.1 Discussion

Several methodologies have been created with the purpose of aiding and standardizing the development of ontologies. Each one has its techniques and principles. However, it should be noted that, no matter which methodology is adopted, knowledge specialists should always think about re-utilizing concepts from other ontologies.

The evaluation of OntologyJam, made during its certification phase by final users has left important insights that will be considered for the future enhancements of the software tool, as they represent indications improvements that the tool needs to perform its functionalities more appropriately.

In relation to functionalities, the search function was the one that had the lowest evaluation, which is due to the way in which the search is done. OntologyJam conducts the search based on the comparison of texts, with simple differentiation between higher and lower case letters, without taking into account the semantics of the terms, for example, radicals, synonyms and hyper-synonyms. Ideally, this search should be conducted using adequate algorithms. This way, the final result will be more explicit and efficient.
The development of this project was based on the re-utilization of the concepts already defined in other ontologies with the objective of aiding knowledge specialists in the first steps of creating an ontology. The result was the development of a tool that aids not only in re-utilization, but also in sharing, management and developing common understanding.

A very simple prototype was developed at the beginning of the project, with a few options for importing, searching and exporting, with minimum resources implemented within each one of these functionalities. This prototype was of great importance to understand the needs of knowledge specialists.

During the project, enough knowledge was acquired on ontology building, as well as working with tools such as the Jena framework. Thus, it was noticed that the project could go far beyond the initial idea. Several functions were included as further ideas came about. And so it can be concluded that the tool exceeded the initial expectations, even though it must be emphasized that software will only be useful if it has the approval of users.

OntologyJam was tested by people with knowledge of ontologies. Beyond testing, a questionnaire was requested with the purpose of obtaining an evaluation of all functionalities, the icons, the navigation and the results presented by the tool through a critical view of the users.

The results of the questionnaire were very positive, again exceeding expectations. Therefore, the conclusion that can be reached is that OntologyJam will be well accepted by the majority of users, as a useful everyday tool for knowledge specialists. Thus, all the objectives established at the beginning of this project were achieved successfully.

The main contribution of this project is a specific tool for re-utilizing concepts from other ontologies; a need not hitherto attended to by IT. However, the conclusion of this project is far from meeting all the needs of the ontology creation process.

With the experience acquired during this project, new opportunities will be opened for the implementation of the new tools that can meet all the needs of knowledge specialists. The integration of these tools based on existing methodologies will enable new domains of knowledge to be mapped in new ontologies in a much easier way.
Recommendation, 2004, available in: 
<http://www.w3.org/TR/owl-features/>.
MORENTE Manuel Garcia, “Fundamentos de Filosofia, Lições Preliminares”, 2008, capítulo 20. available in: 
SMITH Michael; WELTY Chris; McGUINNESS, “OWL Web Ontology Language Guide”, W3C Recommendation, 2004, available in: 
SURE York; STUDER Rudi; “On-To-Knowledge Methodology - Final Version”, University of Karlsruhe, 2002