Reusing Knowledge from the Semantic Web with the Watson Plugin

Mathieu d'Aquin, Marta Sabou and Enrico Motta
Knowledge Media Institute (KMi)
The Open University, United Kingdom
{m.daquin,r.m.sabou,e.motta}@open.ac.uk

1. INTRODUCTION

Ontology reuse is a complex process involving activities such as searching for relevant ontologies for reuse, assessing the quality of the knowledge to reuse, selecting parts of it and, finally, integrating it in the current ontology project. As the Semantic Web provides more and more ontologies to reuse, there is an increasing need for tools supporting these activities. Semantic Web search engines like WATSON [1] provide useful features to find and select relevant, reusable ontologies, while ontology editors such as the NeOn Toolkit[2] are becoming better at handling multiple and heterogeneous sources of knowledge within the life cycle of an ontology project. However, despite the direct availability of a large amount of knowledge and improved tools, ontology reuse remains a complex task, often performed manually, by combining the output of several different tools.

The WATSON plugin aims to overcome this situation by integrating the search capabilities of the WATSON Semantic Web gateway within the environment of an ontology editor (the NeOn Toolkit). The resulting infrastructure allows the user to perform all the steps necessary for large scale reuse of online knowledge within the same environment where this knowledge is processed and engineered. More specifically, the plugin leverages the features provided by WATSON to find, in online ontologies, statements that are relevant to extend the description of a particular ontology entity, selected by the user during the editing process. These potentially relevant statements are loaded and displayed in the editor so that the user can inspect and evaluate them. Statements selected by the user are then integrated in the edited ontology.

2. CONTEXT

The WATSON plugin[3] is developed as part of the NeOn Project[4], whose goal is to provide effective methodological and tool support for developing applications able to effectively exploit the large amounts of semantic data that are now available on the Web. It integrates the functionalities of the WATSON Semantic Web gateway within the environment of the NeOn toolkit ontology engineering tool.

The Watson Semantic Web Gateway. The goal of the WATSON Semantic Web gateway is to support the development of applications able to dynamically find, select and exploit semantic information made available on the Web [1]. To achieve this, WATSON automatically collects and indexes online semantic documents, to provide a variety of access mechanisms, both for human users and software programs. The combination of mechanisms for searching semantic documents (keyword search), retrieving metadata about these documents and querying their content (e.g., through SPARQL) provides all the necessary elements for applications to select and exploit online semantic resources in a lightweight fashion, without having to download the corresponding ontologies.

The NeOn Toolkit. The NeOn Toolkit is a novel ontology engineering environment that combines industrial-strength robustness with a comprehensive support for the ontology engineering life-cycle. The NeOn Toolkit is designed around an open and modular architecture. Strong emphasis is put on the management of networked ontology. Built on the Eclipse platform[5], the Toolkit defines an open framework for developing additional features in the form of plugins.

3. THE WATSON PLUGIN: OVERVIEW

The WATSON plugin relies on the Watson infrastructure to facilitate the large scale reuse of knowledge available on the open Web. In practice, it allows the ontology developer to find, in existing online ontologies, descriptions of the entities present in the currently edited ontology (i.e., the *base* ontology), to inspect these descriptions (the statements attached to the entities) and to integrate these statements into the base ontology. For example, when extending the base ontology with statements about the class *Researcher*, the WATSON plugin identifies, through WATSON, existing ontologies that contain relevant statements such as:

- *Researcher is a subclass of AcademicStaff*
- *PhDStudent is a subclass of Researcher*
- *Researcher is the domain of the property isAuthorOf*

These statements can be used to extend the edited ontology, integrating them to ensure, for example, that the class

[1] `http://watson.kmi.open.ac.uk`
[2] `http://neon-toolkit.org`
[3] `http://watson.kmi.open.ac.uk/editor_plugins.html`
[4] `http://neon-project.org`
[5] `http://www.eclipse.org/`

Researcher becomes a subclass of a newly integrated class *AcademicStaff*.

In this way, the WATSON plugin literally brings knowledge from the Semantic Web to the "fingertips" of the ontology engineer and seamlessly integrates the reuse of this knowledge within the editing process. While traditionally ontologies are built as local, monolithic models in a controlled environment where only carefully selected external resources are integrated, the WATSON plugin supports the construction of *patchwork ontologies*: ontologies that combine knowledge elements drawn from heterogeneous, distributed ontologies that are automatically and dynamically discovered.

4. USING THE WATSON PLUGIN

In practice, the WATSON plugin interface allows the ontology engineer to select an entity to be *searched* (e.g. the existing class *Researcher*) through a right-click menu on the entity in the ontology hierarchy. A set of queries is then sent to WATSON to retrieve relevant statements for this entity, which are displayed in a specific panel. This takes the form of a list of relevant entities (e.g. existing classes *Researcher* in online ontologies), together with their semantic descriptions (i.e. statements that are attached to them in their original ontologies). Buttons labeled *"Add..."* are placed next to each of the statements in the result panel and allow the ontology engineer to simply integrate the selected statements in the base ontology, potentially creating new entities (e.g. *Academic Staff*).

It is important to notice that knowledge reuse with the WATSON plugin is meant as an interactive and iterative process: newly integrated entities should be edited (using the features of the ontology editor) to fit the requirements of the ontology engineer and the WATSON plugin be used to find statements to extend their initially empty descriptions, hence integrating pieces of knowledge from multiple, and potentially heterogeneous ontologies.

5. CREATING LINKS

An important *side effect* of the usage of the WATSON plugin is that the activity of reusing statements from multiple, online ontologies generates links between the created knowledge and the integrated one, indirectly relating the reused ontologies with each other. Indeed, the WATSON plugin can keep track of the provenance of the reused elements such that each time an entity is created or *is involved in a reused statement*, a link is declared between this entity and the one in the external ontology (using the *owl:sameAs* property). In this case, for example, when integrating the statement, *Researcher is a subclass of AcademicStaff*, two links will be created from the *Researcher* and *AcademicStaff* classes in the base ontology to their corresponding entities in the ontology where the statement originates.

Therefore, if broadly used, the WATSON plugin could produce ontologies that are already linked, or mapped, to related semantic resources, providing additional semantic context for the knowledge they contain. More importantly, such an ontology not only references the reused resources but it also provides a *semantic bridge* connecting their elements with each other. Once published on the Semantic Web, the ontologies created thanks to the WATSON plugin become the central pillars of a cluster of ontologies, interrelating previously unrelated resources. In this way, the notion of networked ontologies, which is being researched within the NeOn project, emerges in practice.

6. DISCUSSION

The WATSON plugin is a tool that facilitates the reuse of online ontologies by extending an ontology editor with the functionalities of the WATSON Semantic Web gateway. As a result, the reuse process becomes closely related to the editing process, allowing the user to reuse online knowledge for each of the edited ontology elements. Besides its impact on the way reuse is performed, the usage of the WATSON plugin also leads to a change in the result of the process: the created ontologies act as *networked ontologies*, interrelating the reused semantic resources, hence contributing to the structure of online knowledge.

The use of the WATSON plugin has been tested on 2 different populations –novices and experts in ontology engineering– realizing two different tasks –building an ontology from scratch and extending one. These experiments allowed us to conclude on the good usability of the tool, as even novices where able to quickly build ontologies integrating knowledge from the Semantic Web (which is impressive considering the effort usually required for such tasks). Moreover, testing our tool on real users shed the light on the potential impact the WATSON plugin can have on knowledge engineering. Indeed, the tool makes possible the fine-grained reuse of knowledge coming from multiple ontologies, automatically selected from the wealth of online ontologies. Hence, it is interesting to note how the role of ontologies changes within this process: They are not anymore the object of the process but rather provide semantic context for the statements that are being reused.

Of course, our tests also highlighted a number of technical and research issues, which are currently being studied to improve the WATSON plugin. In particular, assessing the statements to reuse is still a complex task, requiring to consider the semantic context of the original ontologies, as well as the coherence and compatibility with the requirements of the ontology project, with the current base ontology and with the knowledge reused from other ontologies. Building proper tool support and methodological guidelines to tackle these issues is an important part of our ongoing work concerning the WATSON plugin.

Acknowledgement

This work was funded by the EC IST-FF6-027595 NeOn Project. We thank Fouad Zablith for his help in implementing some features of the WATSON NeOn Toolkit plugin and Mari Carmen Suárez-Figueroa for her involvement in testing the WATSON plugin.

7. REFERENCES

[1] M. d'Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, and E. Motta. Watson: Supporting Next Generation Semantic Web Applications. In *Proc. of the WWW/Internet conference*, 2007.