

A Quick Guide to Knowledge Reuse with the WATSON Plugin for the NeOn Toolkit

Mathieu d'Aquin and Mari Carmen Suárez-Figueroa

The WATSON plugin is a tool which aims to facilitate large scale knowledge reuse by extending an ontology editor with the features of the WATSON Semantic Web search engine¹. With this plugin, it is possible to discover, inspect and reuse ontology statements originating from various online ontologies directly in the ontology engineering environment. This document quickly describes the usage of the WATSON plugin to build or extend an ontology within the NeOn toolkit ontology editor².

The two demo videos available at http://watson.kmi.open.ac.uk/editor_plugins.html and http://videlectures.net/iswc07_daquin_watson/ provide useful additions to this guide.

Basics

The NeOn toolkit is able to consider ontologies in two different formats, F-Logic and OWL. A version of the WATSON plugin exists for both formats, but as it is more stable for F-Logic at the moment, we assume here that an F-Logic ontology is being developed.

The WATSON plugin works in the context of an existing ontology. First, a new F-Logic project has to be created (“New Project”), and a new ontology should be added to the project (“New Ontology” or import an existing one).

Essentially, the WATSON plugin is a feature integrated within the NeOn toolkit that can be asked for information about a particular entity (statements) from any other ontology of the semantic web. It can be triggered through the “right-click” menu of a particular entity.

For example, in figure 1, a class *Researcher* has been created. Clicking on the “Watson Search” item of its right-click menu will trigger a search for any statement on the semantic web concerning a class named *researcher*.

The result of the search for a particular entity is displayed in a separate view. The list of entities that have been found is shown, with for each entity, the statements they are associated with.

In the example of figure 1, several classes *Researcher* have been found in various semantic web ontologies. Among these classes, one is a subclass of *Person*, has for subclass *Computer.Science.Researcher*, has for label *Researcher*, is the domain of the property *listed_author_in*, etc.

¹<http://watson.kmi.open.ac.uk>

²<http://neon-toolkit.org>

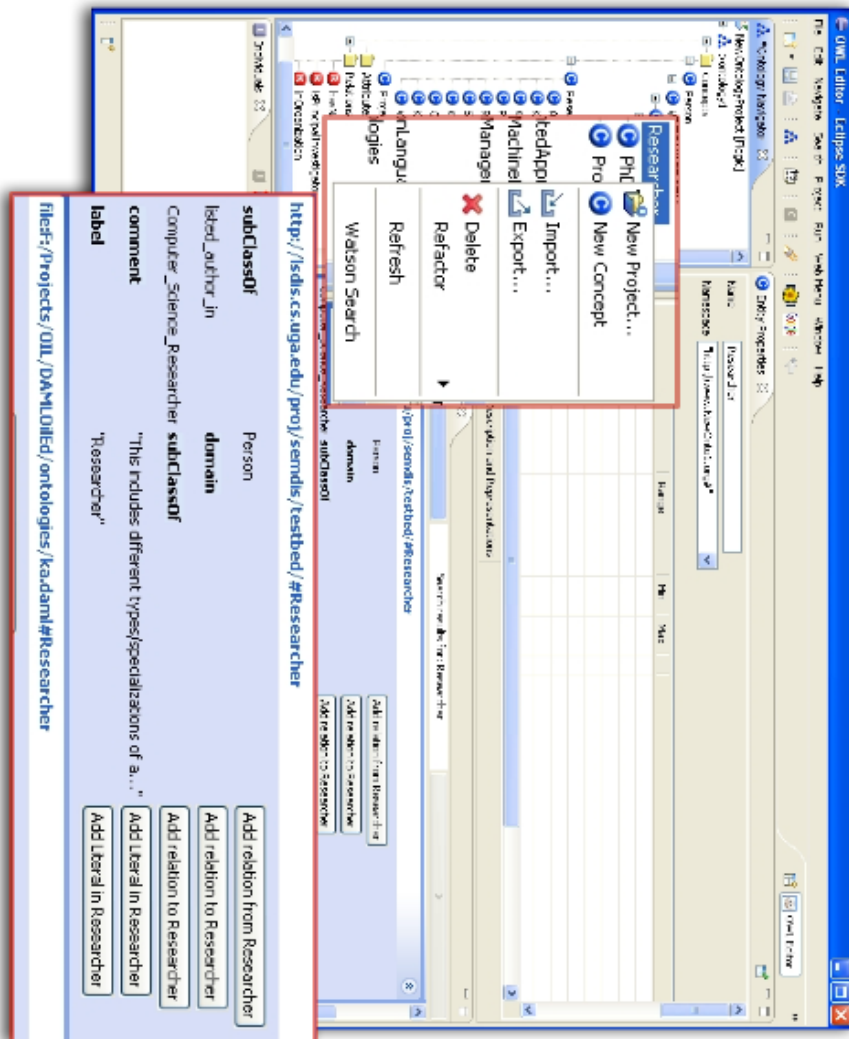


Figure 1: Screenshot of the Watson NeOn Toolkit plugin.

Each of the statements retrieved thanks to WATSON can be imported into the ontology using the button next to it. The statement will then be attached to the original entity (the one that triggered the search) in the currently built ontology.

In our example figure 1, clicking on the button “Add relation from Researcher” next to the statement “**subClassOf** Person” will: 1- create the class *Person* in the current ontology if it does not already exist and 2- make *Researcher* a subclass of *Person*. In the same way, importing the statement “**listed_author_in domain**” will create the relation *listed_author_in* with *Researcher* as domain.

Choosing the Statements to Import

Note: This section is based on the section 7.5. *Proposed Guidelines for Reusing Ontology Statements* of the deliverable *D5.4.1: NeOn Methodology for Building Contextualized Ontology Networks* of the NeOn project. See <http://neon-project.org> to obtain this document.

The WATSON plugin is a tool supporting the task of reusing ontology statements from the semantic web. It can be applied in two different situations:

1. When building an ontology from scratch. In this scenario, it is useful to have a clear idea of what the ontology requirements are, as well as a good understanding of the ontology domain, the ontology purpose, etc.
2. When extending or improving an existing ontology.

It is important to mention that the use of the WATSON plugin to retrieve ontology statements can serve not only for reuse, but also as a way to better understand the domain and the different ways to model it.

Searching for Statements

As explained before, the search with the WATSON plugin is triggered by a base entity in the current ontology. Therefore, when building an ontology from scratch, it is necessary to first create a few core entities in the ontology, choosing for example the most frequent terms in the *Ontology Requirements Specification Document* (ORSD).

When extending an ontology, the existing entities can be used as a base for reuse, but new ones can also be identified in the ORSD.

Selecting the Statements for Reuse

Given the heterogeneity of online available ontologies from a quality perspective, assessing whether a statement satisfies the needs of the current ontology is not trivial. We provide below a set of common sense criteria useful to assess statements:

- Check if the ontology statement belongs to an ontology with the same or similar scope to the ontology being developed. Here, the WATSON web interface (see <http://watson.kmi.open.ac.uk>) can be useful to explore the original ontology.

- Check the purpose of the ontology statements found and the purpose of the ontology being developed to know if they are similar or not.
- Check the clarity of the ontology statement. You should be able to understand precisely the information encoded by the statement.
- Check the information content of the statement. The statement should be informative and a useful addition to the ontology being developed.
- Assess the correctness of the statement from a (formal) modeling perspective. You should agree with the information encoded by the statement and be sure that the statement is not the result of a modeling error (e.g. use of the subclass relation to model a part-of relation).
- Check that the naming of entities in the ontology statement reflect the intended meaning of the statement given its ontological context. The statements may refer to entities that are named in a way that makes sense in there original ontology, but not in yours.

Not the End of the Story

Some statements can be reused as they are, without requiring any effort from the ontology engineers, but, honestly, it is rarely the case. In many cases, newly created entities will have to be re-named, added restrictions, moved at the right place in the class hierarchy, etc. This manual re-engineering of the reused knowledge is necessary to obtain a coherent, homogeneous and well structure ontology.