MOAW – URI's Everywhere

Laurian Gridinoc and Mathieu d'Aquin

Knowledge Media Institute (KMi)
The Open University, Milton Keynes, United Kingdom
{l.gridinoc, m.daquin}@open.ac.uk

The manipulation of Semantic Web data is a relatively difficult and unfriendly task, generally carried out only by experts or technology enthusiasts. URIs of semantic resources for example are not really easy to manipulate, as rich and friendly interfaces allowing to understand and reuse them are still required. Therefore, the current status of the integration of such technologies with more traditional Web interfaces hampers their adoption by a broader public. Of course, many initiatives are currently focusing either on building systems using Semantic Web technologies in a way transparent to the user (e.g. Revyu.com[1]) or in integrating semantic data in the traditional activities of the Web 2.0 user (e.g. MOAT[2]).

MOAW (pronounce like mauve: ‖mõv‖) intends to provide a simple and lightweight contribution to such initiatives. MOAW can be seen as a *URI suggestion tool*, building on the "auto-completion" feature made popular by Web 2.0 websites and Google keyword suggestion. Basically, MOAW can be attached to any text field (HTML `input` element) so that, while typing, suggestions of URIs would appear that can be selected to replace the corresponding word (see Figure 1). The URIs suggested by MOAW are discovered thanks to Watson. Watson[3] can be seen as a search engine for the Semantic Web, crawling, indexing and providing access to Semantic Web resources for applications.

One important strength of MOAW is that, thanks to the use of flexible technologies, it can be easily applied to any online Web form without having to edit the corresponding webpage: URIs coming from virtually anywhere on the Web can be reused virtually anywhere on the Web. In addition, as can be seen in Figure 1, MOAW not only displays the suggested URIs, but also provides, thanks to Watson, a rich description of each URI so that it can be properly understood and selected by the user.

Underlying technologies: In practice MOAW takes the form of a *bookmarklet*: a link is dragged onto the bookmarks of the client browser and, whenever this bookmark is clicked, the Javascript code of MOAW is "injected" into the current webpage. This procedure for "installing" MOAW is therefore relatively simple and provides an homogeneous way to extend the capability of any web form, in potentially any browser.[4]

[1] http://revyu.com
[2] http://moat-project.org/
[3] http://watson.kmi.open.ac.uk
[4] Note that the current prototype version only works with Firefox and Safari 3.

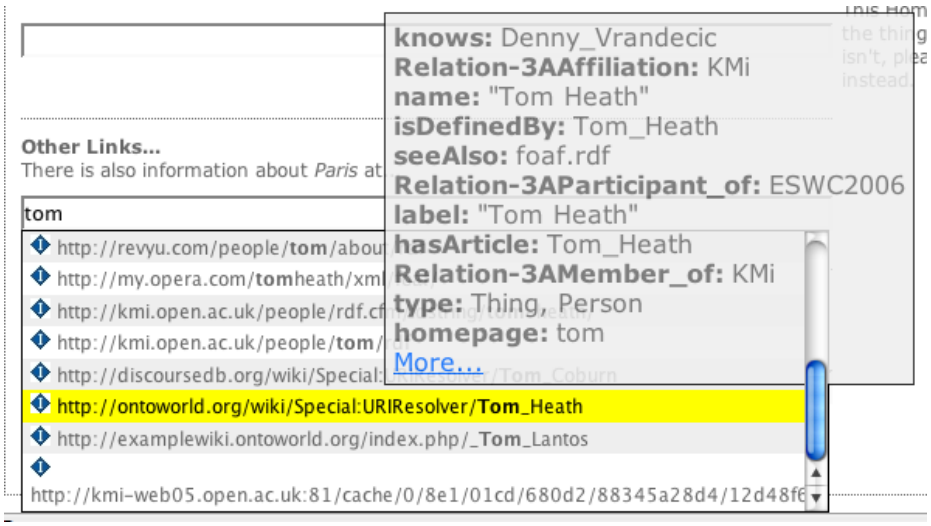

Fig. 1. MOAW suggesting URIs for "tom" on the Revyu.com website.

Once activated, MOAW loads the necessary scripts and libraries, and attach the auto-completion feature to any "input text" in the current webpage. This feature is based on the JQuery auto-complete plugin[5], that has been customized to search and display URIs of entities from Watson. Searching and retrieving rich descriptions of URIs is realized through the Watson Javascript library, which is based on AJAX for the communication with the Watson server.[6] Finally, the description of each URI (type, relations, link to Watson for "More..." information) is displayed thanks to a JQuery tooltip plugin.[7]

What to do with it? The original motivation for the development of MOAW was the possibility to easily bring URIs for the purpose of editing semantic data in the (next) version of Tabulator.[8] However, MOAW is a nice little tool that can be used anywhere a web form is present and it makes sense to fill it with URIs. It can be envisaged, for example, to use it for tagging resources with URIs instead of terms in various systems (blogs, collaborative bookmarks, etc.) Another interesting scenario can be, whenever HTML is edited online, to facilitate the integration of RDFa annotations.

The current prototype corresponds to an early implementation. We intend to improve it both on the level of robustness and of the features it provides. The source code can be retrieved from the web page of the project (http://watson.kmi.open.ac.uk/MOAW) and is open to any form of contribution.

[5] http://www.bassistance.de/jquery-plugins/jquery-plugin-autocomplete/
[6] We rely on the DWR framework (http://getahead.org/dwr/), which takes care of cross site scripting issues
[7] http://bassistance.de/jquery-plugins/jquery-plugin-tooltip/
[8] http://eprints.ecs.soton.ac.uk/14773/