A Logic-Based Semantic Web HTML Generator – A Poor Man's Publishing Approach

Eero Hyvönen, Arttu Valo, Kim Viljanen, and Markus Holi
University of Helsinki, Helsinki Institute for Information Technology (HIIT)
P.O. Box 26, 00014 UNIVERSITY OF HELSINKI, FINLAND
firstname.lastname@cs.helsinki.fi

ABSTRACT

This paper presents a method and a tool for publishing semantic web content in RDF(S) for the humans as a static HTML page site.

Categories and Subject Descriptors: H.5.4 Information Systems: Hypertext/Hypermedia

General Terms: Design

Keywords: Semantic Web, ontology, logic, content publishing

1. THE IDEA IN A NUTSHELL

Semantic portals[1] are widely used for publishing dynamic semantic web content. This approach has, however, created publication obstacles from the viewpoint of content providers: 1) Only content of certain type conforming to the portal's application ontologies can usually be published. 2) The publication process is dependent on the organization maintaining the portal application. 3) Ordinary Internet Service Providers (ISP) do not allow ordinary users to run servers or provide semantic portal services for their customers. 4) Content publishers do not necessarily have the required technical skills to maintain semantic portals. 5) The portal may provide some of the content only in the "hidden web" which hinders Internet search engines such as Google from indexing some or all of the content. The notion of semantic portal is in these respects in contrast with the very idea of the current web, where everybody can publish content easily and independently by just maintaining HTML files in a public directory.

To address these publishing problems, we have developed a method and a tool "Semantic Web HTML Generator" (SWeHG) [4] for transforming RDF(S) repositories into repositories of semantically indexed and mutually linked static HTML pages.

The input to SWeHG consists of HTML templates and an RDF repository conforming to a set of RDFS ontologies. In addition, the semantics of the tags need to be provided as Prolog predicates (unless a suitable predicate set is already available). The output of SWeHG is a semantically linked space of HTML pages of the following kind: 1) *Resource pages (RPage)* depict selected resources with their metadata. For example, a photo with its metadata can be rendered as an RPage. 2) *Index pages (IPage)* classify RPages along conceptual hierarchical classifications, called *facets* or *views* [5]. By using IPages, RPages can be found along different facets. 3) A *home page (HPage)* defines the entrance page to the HTML repository.

For example, figure 1 presents the HPage of a photo exhibition

Figure 1: A photo exhibition generated with *SWeHG*.

SWeHG generated for the Helsinki University Museum. On the left, two frames containing index views are seen: an alphabetical index and a classified index based on the concepts and subconcepts of an underlying RDFS ontology. In the frame on the right, a selected photo with its metadata and recommended links to related photos is rendered. The recommendation links are generated based on the underlying ontologies, photo metadata, and logical rules that define semantic recommendation relations between photos. For example, if a person is depicted in a photo, then links to photographs depicting her relatives can be generated with labels explaining the family relation in question. The museum can publish the exhibition by just copying the pages into a public HTML directory. The RDF repository and ontologies in this case were originally developed for a stand-alone kiosk application. Using SWeHG the same content can be republished on the Semantic Web.

2. TAG SEMANTICS IN LOGIC

SWeHG is based on descriptions on two levels: 1) The layout of the HTML pages is described on the *HTML level* by templates using custom tags. 2) The semantics of the tags is defined on the *RDF level* in terms of logical rules based on the input RDF(S) content. The idea is that an HTML designer can design the layout of the page repository to be generated by using tags without knowing details of the underlying RDF structures, RDFS ontologies, and Prolog programming. RDF(S) related knowledge as well as programming capability in Prolog is needed only for the system programmer when defining the tags. The same tags can be re-used in applications conforming to similar ontological schemas.

SWeHG provides the HTML designer with three special tags: getProperty, getLinks, and getView. The tag <getProperty name=*p*> tag is used for rendering a label related to the resource underlying

[1] See, e.g., http://www.ontoweb.org and http://www.mindswap.org.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

an RPage. For example, the metadata property values of the photo in figure 1 are rendered in this way. The relation p can be specified by the system programmer on the RDF level freely by a logical predicate.

The tag <getLinks> is used for rendering links between RPages. For example, the tag

```
<swehg:getLinks name="SameLocation"
   listType="ul" listStyle="text-size: 10;"/>
```

could expand into the following HTML code linking photographs taken at the same location:

```
<ul style="text-size: 10;">
  <li><a href="entry.Mediacard_00071.html">
     View from Eiffel-tower</a></li>
  <li><a href="entry.Mediacard_00143.html">
     Cafe Parisienne</a></li> ...
</ul>
```

On the RDF level, the criterion `SameLocation` for the linkage could defined by the predicate below[2]. It associates the attribute `SameLocation` with the HTML link label 'Same Place' and the predicate `photosWithSameLocation` defining the link relation.

```
swehg_relation_rule( 'SameLocation',
  'Same Place', photosWithSameLocation).
photosWithSameLocation(Context, Target) :-
  photo(Context), photo(Target),
  rdf(Context, _:place, Location),
  rdf(Target, _:place, Location),
  not(Context == Target).
```

The tag <getView> renders into a hierarchical index-like decomposition of category resources used in IPages. Each category is associated with a set of subcategories and additional individuals of the categories. A view is defined by specifying 1) the root resource selector, 2) a binary subcategory relation predicate, and 3) a binary relation predicate that maps the hierarchy categories with the individuals used as leaves in the view. For example, the tag

```
<swehg:getView
  roots="buildings" branches="subclass"
  leaves="photoOf" listType="ul" />
```

expands recursively into a hierarchical unordered tree (ul), where the leaves are links to photo record resources related to different building categories. The predicate definitions defining meaning of the attribute values can be, for example, the following:

```
buildings(URI) :-
  rdf(URI, rdf:type, 'http://some.org#building').
subclass(SubCategory, SuperCategory) :-
  rdf(SubCategory, rdfs:subClassOf, SuperCategory).
photoOf(Class, Record) :-
  rdf(Instance, rdf:type, Class),
  rdf(Record, dc:subject, Instance).
```

Here `buildings` selects the class `building` as the view root, and the hierarchy is expanded along the `rdfs:subClassOf` property. The `photoOf` predicate relates each building type c of this tree with a set of photo record resources which are used as the leaf categories of c. These are presented as HTML links to the corresponding RPages. The view expansion into HTML can be controlled with the help of additional tag attributes for, e.g., ordering the categories.

[2]The examples are presented in SWI-Prolog (http://www.swi-prolog.org) syntax. Here RDF triples are presented as rdf(Subject, Predicate, Object). Underscore "_" is an unnamed variable.

SWeHG also performs an analysis for the generated HTML pages and identifies the following potential problems: *Self loops* (a link that points to the page itself), *Bad links* (link pointing to a non existing page), *Dead ends* (an RPage with no outbound links), *No way in* (an RPage with no inbound links from any RPages or IPages), *Not in index* (an RPage with no inbound links from any IPage), and *Unused rules* (rules that are newer referred to when generating the HTML repository). This helps the designer in debugging the specifications.

3. DISCUSSION

Logic and dynamic link creation on the semantic web has been discussed, e.g., in [2, 1]. Our approach is different in it's use of HTML templates and Prolog for describing the static HTML output. In the *RDF Twig* tool[3] the RDF to HTML transformation is based on XSLT. A problem here is that an RDF graph can be serialized in many ways in XML. In *Spectacle*[4] the RDF to HTML transformation is based on APIs. Then the user must write programs that use the API, and also an application server is needed. In contrast, our approach is based on tags, is declarative, and the result is a set of static pages whose linkage structure is inferred by logical linking predicates.

Our initial experiences with SWeHG indicate that the presented approach is feasible. The idea of using logic and Prolog for defining the semantics of the tags seems powerful. Complicated semantic link relations and views can be defined and modified easily and declaratively. However, more work and testing is still needed in order to evaluate the usability of *SWeHG* and in determining what additional features may be needed in the system. More work is also needed in optimizing the efficiency of the code and in providing better development tools for the HTML designer and system programmer using the system.

A semantically linked HTML page repository generated by SWeHG can be tried out on the web at http://www.cs.helsinki.fi/group/seco/swehg/. The RDF(S) there comes from the MuseumFinland semantic portal [3] and contains 1200 objects from the museum collections of the Espoo City Museum in Finland.

Acknowledgments

M. Kiesilä, V. Komulainen, R. Köppä-Laitinen, and J. Muhonen participated in the implementation project. Our work was mainly funded by the National Technology Agency Tekes.

4. REFERENCES

[1] P. Dolong, N. Henze, and W. Neijdl. Logic-based open hypermedia for the semantic web. In *Proceedings of the Int. Workshop on Hypermedia and the Semantic Web, Hypertext 2003 Conference, Nottinghan, UK*, 2003.

[2] C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall. Conceptual open hypermedia = the semantic web? In *Proceedings of the WWW2001, Semantic Web Workshop, Hongkong*, 2001.

[3] E. Hyvönen, M. Junnila, S. Kettula, E. Mäkelä, S. Saarela, A. Syreeni, M. Salminen, A. Valo, and K. Viljanen. MuseumFinland – Finnish Museums on the Semantic Web. User's Perspective. In *Proceedings of Museum and the Web 2004 (MW2004), Arlington, Virginia, USA*, 2004.

[4] E. Hyvönen, A. Valo, K. Viljanen, and M. Holi. Publishing semantic web content as semantically linked HTML pages. In *Proceedings of XML Finland 2003, Kuopio, Finland*, 2003. http://www.cs.helsinki.fi /u/eahyvone/publications/ xmlfi nland2003/swehg_article_xmlfi 2003.pdf.

[5] A. S. Pollitt. The key role of classifi cation and indexing in view-based searching. Technical report, University of Huddersfi eld, UK, 1998. http://www.ifla.org/IV/ifla63/63polst.pdf.

[3]http:/rdftwig.sourceforge.net/

[4]http://www.aidministrator.nl/spectacle/