A topic hierarchy on the web*

Valentin Zacharias
ontoprise© GmbH
Amalienbadstr. 36 (Raumfabrik 29)
76227 Karlsruhe
zacharias@ontoprise.de

Stephan Grimm
Research Center for Information Technologies
Haid-und-Neu-Straße 10-14
76131 Karlsruhe
grimm@fzi.de

Abstract

We present the architecture and interface of a metadata registry for a large e-learning site. The metadata registry is very simple to integrate by both content and application providers. It takes its inspiration from currently successful metadata architectures and aims to be an evolutionary change to the web – using long established standards where possible.

1 Motivation

Unitracc[1] is an internet based e-learning system for the area of canalization. The system already contains a large number of information units, especially digital versions of two standard textbooks about canalization. Besides learning material the system also offers tools that help public authorities manage and monitor underground infrastructure.
Currently we are trying to enrich the system with third party content; in the beginning we expect this to be mostly information about tools and machines supplied by the vendors. Third party content can either be directly entered into the unitracc system or found by a crawler from sites known to have information about canalization. In both cases we would like people to adopt our topic hierarchy; this would make it easier to integrate the information into the unitracc system. In order to achieve this goal we need to make the metadata registry as open and easy to use as possible. In this context it is unlikely that content providers will download an OWL[2] or FLogic[3] Ontology to learn about the topics and their relations. It is also unlikely that they will be willing to spend large amounts of time learning about RDF[4], OWL or how to embed metadata in web pages. In addition we want to allow other sites providers to profit from the existence of annotated data in the easiest way possible – hoping that this will increase their motivation to contribute annotated data.

2 Inspiration

We did a survey of existing topic annotation approaches to find those that actually work in a heterogeneous environment. The ones we found to be the most successful where the various kinds of tags used in the blogging and social software community.
The probably most successful format is the tag format pioneered by Technorati[5]. The number of blogs that publish information according to this standard is estimated to be in the millions[Gibson,2005], an important tag like "Life 8" is used to annotate almost 15000 blog entries. In this architecture there is no explicit schema, everybody is free to use any tag she chooses – a concept known as folksonomy. There are different ways to annotate an information unit with a tag, the simplest one is known as "reltag microformat". This format consist of just an html link with the attribute rel="tag". Usually such a link points to an uri of the form http://www.technorati.com/tag/[tagname], for example including
 Life8
annotates a webpage with the tag Life8. It is important to note, that the tag URI can be used to access information about the tag – giving a list of information units recently annotated with this tag, related tags and links that often appear in these information units. Using tag URIs that can actually be accessed is also used to a great extend by other social software like del.icio.us[6] or flickr[7]. Both of these services allow to augment the URI in order to get different information, for example http://del.icio.us/tag/[tagname] returns a list of links annotated with a certain tag and http://del.icio.us/rss/tag/[tagname] returns recent changes to this list as RSS file.

* The authors acknowledge support by the German Federal Ministry of Education and Research under the ksi_underground project. The expressed content is the view of the authors and not necessarily the view of the project as a whole.

[1] http://www.unitracc.de
[2] http://www.w3.org/2001/sw/WebOnt/
[3] http://www.cs.umbc.edu/771/papers/flogic.pdf
[4] http://www.w3.org/RDF/
[5] http://www.technorati.de
[6] http://del.icio.us
[7] http://www.flickr.com

3 Implementation

Starting from the above described observations we are currently building a metadata registry with the central goal of simplicity for content providers.

2.1 Getting information from the registry

All URI for the topics are of the format http://unitracc.de/topic/[topicID]. These URIs are also the starting points when interacting with the registry. Calling the URI returns a HTML page describing the topic, its name, related topics, super- and subtopics. HTTP Get parameters can be used to augment the registries response:

- "format" with possible values html and xml allows to get versions suited for human or computer processing.
- "docs", with an integer value asks the registry to return a number of documents annotated with this topic. "from" can be used to give an index where the list begins. "search" can be used to search in the documents annotated with the current topic for a search string.
- "news" returns a RSS file with recent changes to this topic and new additions. This parameter cannot be combined with "format" or "docs".
- "trans", true or false. If documents annotated with subtopics of the current one should be returned or if changes and additions to subtopics of the current one should be returned.
- "lan", currently "en" or "de" allows to choose the language of topic descriptions.

For example http://unitracc.de/topic/leak_test ?format=xml&docs=20&from=15&trans=false , returns the documents 15-35 from the documents annotated with topic leak_test. The list is returned in a simple, self-explanatory xml format; documents annotated with subtopics of leak_test are not returned. Documents are sorted by the time they where added to the index, newest come first.

Calling the base url http://unitracc.de/topic/ returns a list of all topics, again formatted in html or xml and with the possibility to get the most recent changes as RSS.

Another entry point are URLs of the form http://unitracc.de/document/url=[DOCURL] where DOCURL is a URL known to the system. Calling such an URL returns the information stored about the document at that location, i.e. topics, the time it was indexed and the content of the document stored in the text index. Depending on the form parameter this is returned as xml or html.

The third entry point at /search offers the usual text search interface. Search terms entered into this interface are first analyzed for references to topics, i.e. for each topic a number of lexical references exist (name and synonyms) and an occurrence of one of these in the search string counts as reference to this topic. The index is then searched for documents that contain the search terms and/or are annotated with the topics (or their sub-/supertopics) referenced in the search string. A document that is annotated with the exact topic referenced in the search string is ranked higher than a document annotated with a sub-/supertopic. A document whose content contains a search term is ranked lower than a document that is annotated with a referenced topic.

2.2 Posting information

We will allow sites that annotate their content with the unitracc topics to register with our site and we will crawl these site periodically. Metadata about information on these sites can be included by using the above described relTag microformat. So including

Leak Test

in such a site annotates this page with the topic leak test.

Two possibilities exist to alert the metadata repository to the presence of new content: a HTTP POST to the base uri http://unitracc.de/topic/ with a parameter "document" that specifies the url of the changed document. The document at the supplied URL is then fetched, searched for any topics specified in the relTag microformat and added to the index. To prevent SPAM the HTTP user agent of a post request must be a key that is obtained by registering with unitracc[8].

The second possibility is to make such a POST request on the URL of a topic; this stores the information that a particular document has the topic on which the POST request was called. So making a POST request on URL http://unitracc.de/topic/leak_test with the parameter document="www.leaktestingspec.com" adds the information that www.leaktestingspec.com is about leak testing, even if the page has no annotations on it. If the registry hasn't seen the url before, the document at this url is also fetched and the content is added to the full text index.

3.3 Implementation Details

The unitracc metadata repository is implemented as Servlets using Java 5 and Apache Tomcat 5.5.9[9]. The HTMLParser[10] package is used to analyze retrieved documents. The actual index of the documents and their metadata is stored using the open source software Lucene[11].

References

[Gibson, 2005] Bud Gibson. *Imitation is the surest sign of success.* http://thecommunityengine.com/home/archives/2005/07/imitation_is_th.html, 2005.

[8] To further simplify things and allow for easy testing out of every browser, we allow to simulate such post requests by using GET with parameters action="post" and user_agent="…"

[9] http://jakarta.apache.org/tomcat/

[10] http://htmlparser.sourceforge.net/

[11] http://lucene.apache.org/