HowTo use ontoElicitor to develop an ontology

Jeff Gentry

February 23, 2007

1 Overview

The *ontoElicitor* package is used to generate feeback on an ontology with the intention of collaboratively developing and refining an ontology. This article will demonstrate how to install and run the *ontoElicitor* package to develop your own ontologies.

This document will assume some basic knowledge of both R and Zope, as well as the administrator having both a working R installation and a working Zope server. If you need instructions for how to interact with your Zope server as an admistrative user, please contact your system administrator.

2 Installation

The first step is to install the *ontoElicitor* package, via *R CMD INSTALL*. The second step is to locate the appropriate `Zope` product, which is located in */inst/zope* of the package's directory structure. Find the tarball, `OntoElicitor.tgz`, uncompress it and copy that directory structure to the *Products* directory of your `Zope` instance.

The *index.html* file for the Zope product has a generic welcome string in there. Click on your *OntoElicitor* instantiation in the Zope Management Interface, then select the *Contents* tab. Finally, click on the *index.html* file. You will see a string that says *Welcome to the INSERT INSTITUTION OntoElicitor, please enter your name and email address to continue:*. You may change that to whatever you would like, in particular change the *INSERT INSTITUTION* portion. Do not change anything else in this file.

Make sure to restart your Zope instance before proceeding.

3 Your initial ontology

The *ontoElicitor* package now uses a *Ontology* class structure, based around graphs. Please see the *Working with graph-based ontologies in ontoElicitor* vignette for more information on creating and interacting with these objects.

4 Building the interactive web pages

The *ontoElicitor* R package is primarily used to generate the actual web content for the ontology. To do this, create a directory on the local filesystem and use the `buildOntoHTML` function. This function takes the ontology object and optionally a string *titleBase* which will be prepended to 'Ontology Feedback Tool' as the title of the main page. The other optional argument is *dir* which is the output directory, defaulting to the current working directory.

In this example we will be using the supplied pre-built SGDI breast cancer ontology, and construct them in a temporary directory.

```
> library(ontoElicitor)
> data(breastCancer)
> tDir <- tempdir()
> buildOntoHTML(breastCancer, titleBase = "SGDI Breast Cancer",
+     dir = tDir)
```

There is one more argument, *dispDepth*. This argument controls the default depth of the graph that is visible in the Javascript tree. By default, the value is *1*, showing the first two levels of the tree.

5 Getting the pages to the web

First create an instance of the *OntoElicitor* product in your ZMI. You can give it any name or title that you wish. Next, click on the icon for this *OntoElicitor* object and click on the *Contents* tab. This will show you the internal structure of your *OntoElicitor*. You will see a subfolder named *Onto*, this is where your web content will be placed.

Connect to your `Zope` ZMI via WebDAV or FTP (instructions on how to do this can be found on the `Zope` website at `http://www.zope.org/Documentation/Articles/WebDAV`. Upload `all` of the web files generated by `buildOntoHTML` into the `Onto` directory of your *OntoElicitor* instance. At this point you should be able to connect with a web client to `http://your.zope.server/INSTANCE/index.html` where *INSTANCE* is the name of your *OntoElicitor*.

6 Reviewing feedback

To look at the feeback given by users, you can click on the *View Feedback* tab in the *OntoElicitor*'s ZMI interface or at `http://your.zope.server/INSTANCE/viewFeedback` (although to do the latter you will still need a valid login for your Zope server). You will be presented with a listing of every bit of feedback provided by users, as well as the ability to download an XML file representing this feedback. This XML file could be used for further processing in another system.

7 XML schema

The schema used for the XML extraction of feedback is quite simple and consists of the following tags:

- *OntoFeedback* : Wraps all feedback items, ie the master tag.
- *session* : Subblock of *OntoFeedback*. Denotes a specific piece of feedback.
- *user* : Subblock of *session*. Starts the block to hold the submitter's information.
- *name* : Subblock of *user*. The name of the submitter.
- *email* : Subblock of *user*. The email of the submitter.
- *feedback* : Subblock of *session*. Starts the block to hold the submitted feedback.
- *feedbackCheck* : Subblock of *feedback*. Will list any (dis)agreement specified via checkboxes during the feedback process.
- *source* : Subblock of *feedback*. The source node of the feedback.
- *text* : Subblock of *feedback*. The text given for feedback.