Ontologies for Sharing, Ontologies for Use

David Shotton and Chris Catton

Image Bioinformatics Research Group
Department of Zoology, University of Oxford, South Parks Road, Oxford OX1 3PS, UK
E-mail: david.shotton@zoo.ox.ac.uk, chris.catton@zoo.ox.ac.uk
Direct phone: +44-(0)1865-271193/281993; Department fax: +44-(0)1865-310447

Principles of ontology design

1. Ontologies designed for sharing within a community should be kept small or very small, each being simple subsumption (*is_a*) hierarchy in which sibling classes are disjoint from each other, and where each class represents a single notion. This makes the ontologies easier to manage and easier to validate by domain experts. It is easier to achieve consensus about a small ontology, just because there are fewer classes and properties to disagree about. Small ontologies are also easier to import by people wishing to use them for third-party applications.
2. It follows from this that in any collection of related small ontology modules, covering a particular domain of knowledge, each class must belong only to one ontology module. If a class appears to belong to two modules, it probably conflates two notions which need to be distinguished and separated.
3. Ontology building is an empirical activity - while each ontology must be fit for purpose, clear and unambiguous, it does not have to be theoretically perfect or complete.
4. When writing an ontology that covers a small, specialist area, one should not specify what does not need to be specified. In particular, one should think carefully before specifying the domain and range of a property, since this can cause problems for others extending from your ontology.
5. Third-party ontologies should be used wherever they are available and suitable – there is no point re-inventing wheels.
6. More complex ontologies for specific applications can be built by combining small ontology modules, and then by adding restrictions to enrich their meaning.

These principles owe much to the insights of the Manchester ontologists Alan Rector and Robert Stevens.

Gruber (1993, in *Knowledge Acquisition*, **5**: 199-220) stated:

> '*An ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities . . . An ontology serves a different purpose than a knowledge base, and therefore a different notion of representational adequacy [McCarthy and Hayes, 1969] applies*'.

However, in the current wave of interest in the semantic web and ontologies, this distinction has been blurred.

Ontologies for sharing, ontologies for use

We propose a distinction between **a public shared ontology** that:

- acts primarily as a structured defined vocabulary,
- should define a very limited domain,
- should be a simple subsumption hierarchy with disjoint sibling classes,
- should not import any other ontology – i.e. it must stand alone, and
- should ideally, for adherence to standards, be written in OWL;

and **an application-level ontology** that:

- imports one or more public ontologies describing particular domains,
- restricts and cross-relates the public ontologies, thereby enabling more powerful reasoning for use within a particular application (e.g. a database), and
- should be written in OWL-DL to permit use of a Description Logic reasoner both for validation and for inference of additional relationships defined by restrictions.

The application-level ontology is still a 'shared conceptualization', but it is now shared only implicitly, by the users of the application. The public ontology, being simpler and more loosely defined, can gain a far greater degree of consensus.

The ontology development life cycle

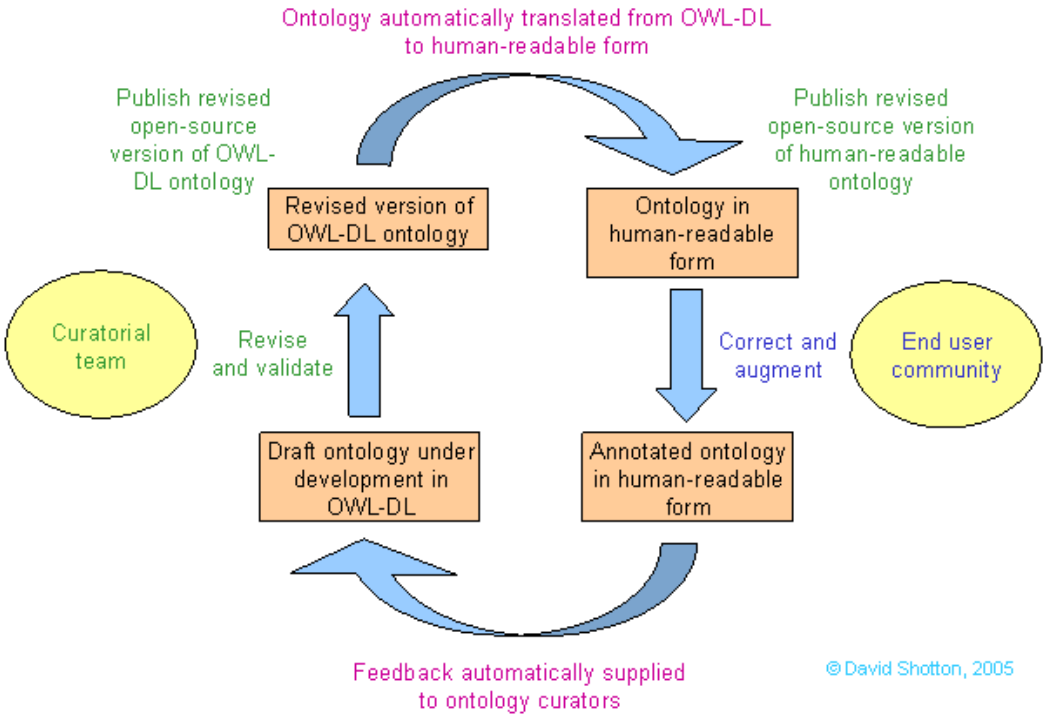
Simplifying the scope of public shared ontologies should make it easier to 'bring them to market'. However, there will still be a need for ontology evolution. The ontology development life cycle is best described as an interative process of

(a) curation and validation by a central team who take responsibility for the ontology and its publication, and

(b) revision and extension of the ontology in the light of experience by domain experts within the community.

This is summarized in the following diagram:

Figure 1: An iterative model of ontology development, discussion, revision and validation



Ontology viewing and editing

The best 'human-readable form' for the ontology is at present uncertain. Protégé itself is usable, but its user interface is too complex and cluttered for the casual user. Ideally, the Protégé team would further develop the prototype Protégé Web Browser application developed by a student

(http://protege.stanford.edu/ plugins/protege_browser/index.html), which in its present form is acknowledged by the team to be flaky.

In the absence of that, a very good alternative for viewing OWL ontologies is DumpOnt (http://www.daml.org/2003/09/dumpont/), a program that displays the class and property hierarchies present in an RDF Schema or in a DAML+OIL or OWL ontology. It will display any Web-published ontology in your browser if you use the following URL format:
http://www.daml.org/cgi-bin/dumpont?http://ontologyURL/ontologyfilename.owl, e.g.
http://www.daml.org/cgi-bin/dumpont?http://www.co-ode.org/ontologies/pizza/pizza_20041007.owl.

A nice feature of DumpOnt is that clicking on a class takes you to the OWL RDF for that class. However, it cannot be used for editing the ontology, which in most cases will rapidly be seen as a major limitation.

The ontology editor SWOOP (http://www.mindswap.org/2004/SWOOP) is emerging as a leading candidate for an all-round editor of OWL-DL ontologies.

Problems with existing ontologies, and a suggested revision strategy

The ImageStore ontology that we have developed as a central component of the BioImage Database (http://www.bioimage.org) has grown organically over the last three years, in the period before the design principles elaborated above were developed. As a result it contains classes relating to a wide variety of domains, from image formats to publication citations, and from the personal details of individual image creators to details of the subjects of the images. As such, it is ripe for untangling into a set of clean simple orthogonal ontologies fit for sharing. We believe many others' home-grown ontologies will be in a similar state. Looked at in the light of the principles outlined above, such existing ontologies may include the following shortcoming:

1. They may contain a rich mixture of related class terms that are not clearly differentiated.
2. They may contain duplications, i.e. the same concept appearing more than once under different superclasses.
3. They may contain logical inconsistencies.
4. They may contain a number of classes bearing vague names that are too general to be useful.
5. They may contain ambiguous class names.
6. They may contain class names that suggest that they should be part of separate third-party ontologies.

In part these shortcomings will be because each ontology is too large and complex for any one person to hold in his/her head in its entirety.

We thus propose that the items within such an ontology should be clearly separated into a number of independent ontology modules, each of which should be small enough to be comprehended in its entirety by a group of domain experts, who can revise it according to the principles of simplicity outlined above.

This simple act of deciding to separate classes from the original single ontology into a set of non-overlapping ontological modules is, we believe, a necessary first stage for reviewing the accuracy and completeness of coverage of each topic. During that review, items in the existing ontology should be omitted unless there is a clear reason to include them, and the existing sub-class structure should be critically reviewed. The depth of branching in each ontology module should be the minimum consistent with the required description.