# Content Object Modelling with Protégé

Jon Carey – Producer

Helen Lippell – Information Architect

David Wood – Senior Software Engineer (david.wood.01@bbc.co.uk)

British Broadcasting Company (BBC),

BBC New Media, London

## Introduction

This paper describes the process by which a project team within BBC New Media used Protégé and its OWL plug-in in order to provide a semantic description of a complex existing ontology and manage its further development.

## Context

Content Management Culture is one of the biggest ongoing projects within the BBC's New Media division. Our remit is to provide predominantly web content management solutions to the new media sections of various BBC divisions. In order to do this we have built a custom interface on top of a commercial content management system (CMS).

The semantic 'content object' model, managed by Information Architects, which underlies the CMS was previously stored in Microsoft Word documents. The static nature of this format impeded both the management of the model and its implementation in a technical environment. Migrating the model into Protégé has standardised the modelling practices and increased the visibility and reusability of modelled components, thereby facilitating faster prototyping.

## Content object modelling

We set out to find a software solution that would allow us to manage a number of content object models and their constituent objects and the attributes within these objects. For the purposes of the content management system we have divided all editorial content into its constituent parts.

A typical web page will be split into: article text, main image, other images, web links etc. all of which are content objects and exist as separate entities in their own right. This allows reuse of even the smallest content objects across a number of web pages without duplicating the object itself.

All content objects are defined by their attributes, which describe each property of the content object, and business rules, which control how an object can be created. Thus, the attributes of an image, for example, could be:

- Image type
- Image size (width and height)
- Image description
- Image ‘Alt’ tag
- Image caption
- Descriptive metadata

And its business rules could be:

- The image type must be one of a specific format (e.g. jpeg/gif)
- The image associated with an article must be of a specific size (e.g. 203 x 152 pixels)
- The image description must have a maximum number of characters in a text string
- The image ‘Alt’ tag and caption must have more than zero characters
- The image must have at least one piece of descriptive metadata

Some attributes may be mandatory in a given object and some may be optional. An attribute can be shared across more than one object; e.g. Headline. There is always a minimum set of attributes for any given object, e.g. the objects’ file name and title.

As such:

- The model is the full set of all possible objects and all the attributes that those objects may have.
- A partner instance of the model is a collection of objects as defined for a specific partner (i.e. a department within the BBC). It will, in most cases, not be all possible objects
- An object is a collection of attributes which make up a content entity such as an image or a web link
- An attribute is a property of the object (e.g. headline, author, size)
- The attributes are of a specified type (e.g. text, date)
- The attributes have business rules, including cardinality (one or many), type, validation rules (e.g. number only or a complex regular expression) etc

Partner instances of the model are made up of a particular grouping of objects tailored to suit a particular partner. Some objects will be used across more than one partner instance of the model. There may be differences between different instances of the same object according to a partner’s specific need.

There are at present two partner instances of the model and there will be more in future as we deploy our system to new partners within the BBC.

## How we used Protégé

After a requirements gathering exercise, and a process of analysing various modelling tools, Protégé was selected on the basis of its support for OWL semantics. Two core requirements were that:

- The model couldn't be broken – certain rules would need to be enforced
- XML formats could be generated from whatever output was provided, for use within – but not exclusive to – the CMS

Our first step was to define a meta model of our existing content objects which we could use as a basis for 'how content objects could be created'. This defined three core building blocks of *Container*, *Property* and *ContentObject* – the subtypes of which would form the base of the model, and a starting point for our Protégé classes. Additional classes were I – to represent partner uses of an object, and *ReferenceData* – used to indicate controlled vocabularies from an external resource. Individuals within the model represent specific instances of content objects defined as subtypes of the *ContentObject* class. OWL restrictions are used to enforce content modelling constraints, such as the types of Property objects used for a certain *ContentObject*, or fixing a value of a meta model property.

We had to customise Protégé in the following ways:

- Interaction with CVS – using the NetBeans Java CVS library to version control the OWL files
- OWLTest classes were needed to check the sanity of the meta model
- XML output plug-ins were created to generate output, both for the purposes of documentation, and for use within our CMS
- A TabWidget plug-in was created for management of non-semantic presentation information

## Conclusion

Protégé has provided us with an extensible solution to our content object modelling issues that would have been difficult - if not impossible - to achieve with other software. With the content object model now stored in a semantically defined format (OWL) rather than a Word document, Protégé has empowered our information architecture team to create, prototype and document the modelling process for a technical solution, with minimal future technical involvement.