RDFa in Drupal: Bringing Cheese to the Web of Data

Stéphane Corlosquet, Richard Cyganiak, Axel Polleres and Stefan Decker

Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
{firstname.surname}@deri.org

Abstract. A large number of websites are driven by content management systems (CMS), which manage not only textual content but also structured data related to the site's topic. Exposing this information to the Web of Data has so far required considerable expertise in RDF modelling and programming. We present a plugin for the popular CMS Drupal that enables high-quality RDF output with minimal effort from site administrators. This has the potential of greatly increasing the amount and topical range of information available on the Web of Data.

1 Introduction

Semantic Web technologies have matured to the point where they are increasingly being deployed on the Web. Large amounts of RDF data can now be accessed over the Web as *Linked Data*. This data is used by a variety of clients, such as RDF data mashups that integrate information from various sources, search engines that allow structured queries across multiple sites and datasets, and data browsers that present a site's content in new ways.

But the traditional Web still dwarfs this emerging Web of Data. Thus, the task of "RDFizing" existing websites, which contain structured information such as events, personal profiles, ratings, tags, and locations, is important and of high potential benefit. The recently finished RDFa[1] standard supports this by allowing RDF to be embedded into existing HTML pages.

The Web features some huge websites with millions of pages and users. But a lot of the Web's interest and richness is in the "long tail", in smaller special-interest websites, such as cheese reviews, which will be our example for this demonstration. Our goal is to make domain data from such sites available as RDF. This is challenging for several reasons:

No dedicated development staff. Smaller websites usually run off-the-shelf software, such as CMS, wikis, or forums. Site operators cannot build the RDF support themselves, it has to be provided by the software or plugins.

Per-site schemas. The domain schema differs from site to site. The mapping from the site's schema to RDF vocabularies or ontologies cannot be predefined by a software developer; it must be defined by the site operator.

No ontologists. Site administrators will have little interest in learning the details of RDF and description logics. The process of configuring RDF support has to be simple and straightforward, or else it won't be used.

We show a practical and easy-to-use system that overcomes these challenges and allows the publication of high-quality RDF data from websites that run on off-the-shelf content management systems. It targets the popular Drupal CMS.

2 Drupal and the Content Construction Kit

We will start by briefly introducing Drupal[1] and some of its terminology. Drupal is a popular open-source content management system (CMS). It is among the top three open-source CMS products in terms of market share[5]. Drupal facilitates the creation of websites by handling many aspects of site maintenance, such as data workflow, access control, user accounts, and the encoding and storage of data in the database.

A *site administrator* initially sets up the site by installing the core Drupal Web Application and choosing from a large collection of *modules* that add specific functionality to the site. Site administrators need a fair bit of technical knowledge to choose and configure modules, but usually do not write code; this is done by *module developers* instead. After the site has been set up, Drupal allows *non-technical users* to add content and handle routine maintenance of the site.

Each item of content in Drupal is called a *node*. Nodes usually correspond to the pages of a site. Nodes can be created, edited and deleted by content authors. Some modules extend the nodes, for example a comment module adds blog-style comment boxes to each node.

Another example is the *Content Construction Kit (CCK)*, one of the most popular modules used on Drupal sites. It allows the site administrator to define types of nodes, called *content types*, and to define *fields* for each content type. Fields can be of different kinds such as plain text fields, dates, email addresses, file uploads, or references to other nodes. When defining content types and fields, the site administrator provides the following information:

- label, ID, and description for content types and fields,
- fields can be optional or required,
- fields can have a maximum cardinality,
- fields that reference other nodes can be restricted to nodes of a certain type.

For example, for a cheese review website, the site administrator might define content types such as *Cheese*, *Review*, and *Country of Origin*. The *Cheese* type might have fields such as *description*, *picture*, and *source of milk*.

Thus, site administrators use the CCK to define a site-specific content model, which is then used by content authors to populate the site. The focus of the work we are presenting here is to expose this CCK content as RDF on the Web.

[1] `http://drupal.org/`

3 Weaving Drupal into the Web of Data

Given a Drupal CCK content model consisting of content types, fields, and nodes that instantiate the types, what would be a good way of representing it in RDF? We consider the following features desirable for the RDF output which are in line with the Linked data principles and best practices [3, 4]:

Resolvable HTTP URIs for all resources, to take advantage of existing tools that can consume Linked Data style RDF content.

Re-use of published ontology terms. To support sites of arbitrary subject matter, we cannot in any way pre-define the RDF classes and properties that should be used to express the data. The site administrator has to select them when setting up the content model. But to enable queries across sites, it is necessary that the sites use the same (or mapped) vocabularies. This requires that both sites import vocabulary terms from somewhere else.

Expressing Drupal constraints in OWL. Constraints that are defined on the types and fields (domains, ranges, cardinalities, disjointness) should be automatically published as RDF Schema or OWL expressions.

Auto-generate terms where necessary. Re-use of published ontology terms is important for interoperability, but not always possible or practical, as there might be no existing ontology term matching a type or field, or finding them is too hard.

Safe vocabulary re-use. Mixing the content model constraints with constraints of a published ontology might have unintended semantic effects, especially since most site administrators will not be familiar with the details of OWL semantics. For example, a carelessly applied cardinality constraint could affect the definition of a shared vocabulary term, rendering data published elsewhere inconsistent. The system must prevent such effects as far as possible.

These features strike a balance between preserving as much information as possible from the original content model, keeping the barrier to entry low, and enabling interoperability between multiple data publishers and consumers.

3.1 Site Vocabularies for Basic RDF Output

When the module is first enabled, it defaults to auto-generating RDF classes and properties for all content types and fields. Thereby it provides zero-effort RDFa output for a Drupal site, as long as no mappings to well-known public vocabularies are required.

An RDFS/OWL site vocabulary document that describes the auto-generated terms is automatically generated. The definitions contain label, description, and constraints taken from the CCK type/field definitions. The HTML views of all nodes contain RDFa markup for the type and all shown fields, using the auto-generated classes and properties.

3.2 Mapping the Site Data Model to Existing Ontologies

To map the site data model to existing ontologies, the site administrator first imports the ontology or vocabulary. We assume that it has been created using a tool such as Protégé[2], OpenVocab[3], or Neologism[4], and published somewhere on the Web in RDF Schema or OWL format.

For every content type and field, the site administrator can choose a class or property it should be mapped to. Mappings are expressed as subclass/subproperty relationships. For instance, if the field `description` on type `cheese` is mapped to Dublin Core's `dc:description`, then a triple `site:cheese_Description rdfs:subPropertyOf dc:description` would be added to the site vocabulary.

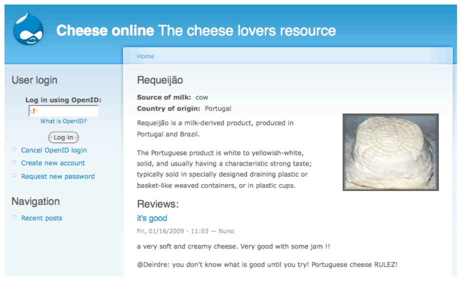
This subclassing is a simple way of minimizing unintended conflicts between the semantics of local and public terms. Per OWL semantics, constraints imposed on the local term by the content model will not apply to the public term. This ensures *safe vocabulary re-use*[2].

It must be stressed that this mapping step is optional, and the main benefit of the Web of Data – exposing site data for re-use by third parties – is realized by the default mapping.

4 The User Experience

This section describes our cheese review site from a user point of view, and shows an example of what can be done to reuse the RDFa data.

Cheese and reviews. Figure 1 shows a cheese entry and its user review. Using the Content Construction Kit, we defined a type (1) `cheese` with fields for the name of the cheese, the source of the milk, the country of origin, a picture and a description and (2) `cheese_review` with fields for the title of the review, a reference to the `cheese` being reviewed and the review.



Requeijão

Source of milk: cow

Country of origin: Portugal

Requeijão is a milk-derived product, produced in Portugal and Brazil.

The Portuguese product is white to yellowish-white, solid, and usually having a characteristic strong taste; typically sold in specially designed draining plastic or basket-like weaved containers, or in plastic cups.

Reviews:

it's good

Fri, 01/16/2009 - 11:03 — Nuno

a very soft and creamy cheese. Very good with some jam !!

@Deirdre: you don't know what is good until you try! Portuguese cheese RULEZ!

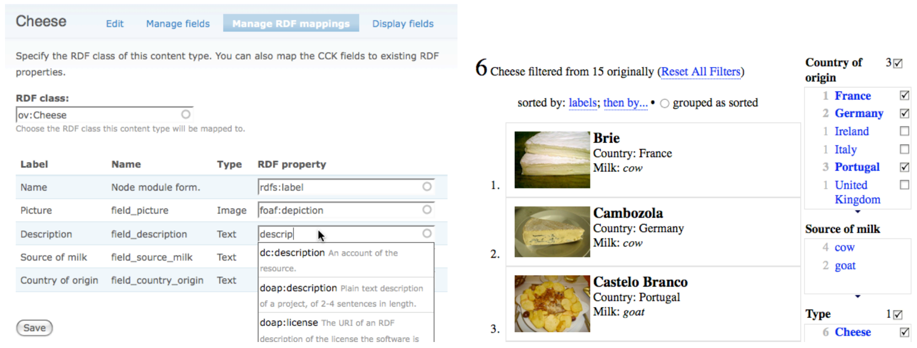
Fig. 1. A cheese with a review. Users can create new cheese entries and add reviews.

[2] `http://protege.stanford.edu/`

[3] `http://open.vocab.org/`

[4] `http://neologism.deri.ie/`

Content type and field mapping. The module adds a "Manage RDF mappings" page to the CCK interface as shown in Figure 2 (left). For a given content type, it offers to map the type to an RDF class and each field to an RDF property. We have mapped the Cheese type and its fields to previously imported RDF terms. In order to ease the mapping process and prevent confusion between classes and properties, the module will only display RDF classes or RDF properties where appropriate. Moreover an AJAX autocomplete search through the imported terms allows the user to quickly identify the most relevant terms for each mapping. These measures help to make the mapping process a fairly straightforward task that does not require deep understanding of the Semantic Web principles.



Specify the RDF class of this content type. You can also map the CCK fields to existing RDF properties.

RDF class:
ov:Cheese
Choose the RDF class this content type will be mapped to.

Label	Name	Type	RDF property
Name	Node module form.		rdfs:label
Picture	field_picture	Image	foaf:depiction
Description	field_description	Text	descrip
Source of milk	field_source_milk	Text	
Country of origin	field_country_origin	Text	

dc:description An account of the resource.
doap:description Plain text description of a project, of 2-4 sentences in length.
doap:license The URI of an RDF description of the license the software is

6 Cheese filtered from 15 originally (Reset All Filters)

sorted by: labels; then by... • ○ grouped as sorted

1. Brie
Country: France
Milk: *cow*

2. Cambozola
Country: Germany
Milk: *cow*

3. Castelo Branco
Country: Portugal
Milk: *goat*

Country of origin 3 ✓
1 France ✓
2 Germany ✓
1 Ireland ☐
1 Italy ☐
3 Portugal ✓
1 United Kingdom ☐

Source of milk
4 cow
2 goat

Type 1 ✓
6 Cheese ✓

Fig. 2. RDF mappings management (left) and Exhibit view (right).

Exhibit view. Next we show a simple example of using the site's RDFa data. Exhibit[5] now supports RDFa natively and it is easy to setup a basic faceted browsing interface on top of an RDFa page. In our example, Exhibit allows filtering of the cheese types via several facets, as shown in Figure 2 (right).

5 Conclusion and Future Work

The presented system differs from existing approaches in several ways. *SIOC exporters* and similar fixed-schema RDF generators cannot deal with the case where the site schema and its mapping into RDF terms are defined by the site administrator at setup time. *Database-level RDF exporters*, such as Triplify[6], require understanding of database technologies and of the application's internal database schema. *Semantic MediaWiki*[7] and similar systems address a different use case: all content authors, rather than just the site administrator, collaboratively define the structure of the site. We address the common case where the site's basic structure should not be changed by individual content authors.

[5] `http://simile.mit.edu/exhibit/`
[6] `http://triplify.org/`
[7] `http://semantic-mediawiki.org/wiki/Semantic_MediaWiki`

The presented system is a working prototype[8]. The module used for the prototype and discussed in this paper is available on drupal.org[9]. Further planned improvements include the use of the semantics of imported ontologies (e.g. domain, range and disjointness constraints) to restrict the selection of available classes and properties in the RDF mapping UI.

Another issue remains for the larger RDF community to solve: a complex part of the process of generating high-quality RDF is the task of finding and choosing good vocabularies and ontologies. There are some services that support this task, but they are not sufficient and this task remains difficult for non-experts. This is a major problem that the community needs to address in order for the Web of Data to succeed.

6 Acknowledgements

The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2), the European FP6 project inContext (IST-034718) and the European Union under Grant No. 215032 (OKKAM).

References

1. B. Adida, M. Birbeck, S. McCarron, and S. Pemberton (eds.). Rdfa in xhtml: Syntax and processing, Oct. 2008. W3C Recommendation, available at `http://www.w3.org/TR/rdfa-syntax/`.
2. A. P. Aidan Hogan, Andreas Harth. Saor: Authoritative reasoning for the web. In *ASWC 2008*, pages 76–90, 2008.
3. D. Berrueta and J. P. (eds.). Best practice recipes for publishing rdf vocabularies, Aug. 2008. W3C Working Group Note, available at `http://www.w3.org/TR/swbp-vocab-pub/`.
4. C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee, editors. *Linked Data on the Web (LDOW2008)*, Apr. 2008.
5. R. Shreves. Open source cms market share. White paper, Water & Stone. `http://waterandstone.com/downloads/2008OpenSourceCMSMarketSurvey.pdf`.

[8] A demo site is online at `http://drupal.deri.ie/cheese/`.

[9] `http://drupal.org/project/rdfcck`