From Online Community Data to RDF

Uldis Bojārs, John G. Breslin
[uldis.bojars,john.breslin]@deri.org

Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland

Abstract

Large amounts of data are created within online community sites (forums, blogs, etc.). These can serve as a valuable source of information for web users, and usually contain rich meta-information. Most of this information is stored in relational databases, but unfortunately remains locked into these databases and cannot be used by other applications.

The SIOC project is aimed at providing guidelines for making this information available on the Web and for using this information for connecting online community sites together. SIOC aims to let other sites know more about the structure and contents of online communities, and to make more use of tagging and semantic metadata in these sites.

This position paper describes the approach we have adopted for making online community site data available in RDF from many applications, and we will illustrates it through the example of a SIOC export tool for b2evolution blog engine. As opposed to extracting data directly from a relational database, we attempt to tie our RDF data producers into the associated application logic for each system and reuse built-in functions and APIs where possible to generate RDF data.

1. Introduction

The “Social Web” contains large volumes of content (blog posts, reviews, etc.) posted on online community sites (such as blogs, wikis and bulletin boards). These sites allow users to gather online, create content and enter into discussions. They contain rich metadata about content items and people creating them but most of this data are locked in HTML markup and not available for reuse without “scraping” the markup. In order to facilitate intelligent reuse of the information contained within these sites we need a data format that better suited for the task.

Semantically-Interlinked Online Communities (SIOC) [Breslin2005] is a project aimed at interconnecting online community sites by making their information available in a machine-readable form. A rich data model is needed if we are to express full information about the content and structure of these sites. The SIOC project defines an ontology for describing this information in RDF and provides several open source SIOC RDF exporters. Online community sites typically run content management systems (CMSs) which consist of a relational database (e.g., MySQL) and a presentation layer for displaying content to visitors. If that is the case, the task of a SIOC export tool is to retrieves information from a relational database and export it in RDF.

This position paper is based on the experience of SIOC developers community in exporting RDF from online community sites. We describe an application logic based or indirect approach for exporting RDF, used by many SIOC export tools, and illustrate it on the example of a SIOC exporter for b2evolution blog engine. The rest of the paper is organized as follows: Section 2 describes characteristics of online community sites; Section 3 identifies different approaches for exposing relational databases in RDF; Section 4 illustrates our approach on an example of a SIOC RDF export tool; and Section 5 concludes the paper.

2. Characteristics of Online Community Sites

The following characteristics of online community sites make the indirect approach described in this paper well-suited for them.

Extensible. Most of the online community site engines are built to be extensible and provide well-documented APIs for use by plugin developers. Most of them are also open source (b2evolution, Drupal, B2evolution, etc.). This enables us to use functions and API calls provided when building RDF export tools.

Dynamically Evolving. At the same time these engines may have very fast development cycles with approximately one major release per year and often many more minor version changes. While these changes may affect both the database schema and public APIs, the latter is usually kept stable. Changes to the functions and APIs to be used by other developers are kept to minimum and well documented. The same is not always true for database schema changes.

Large installation base, all over the Web. This software has many installations by web users who are not experts in software development. These users may still want to enable RDF export from their sites provided that this functionality is simple to install and does not require a large effort or specific knowledge. Many people use web hosting providers which may limit what software they are allowed to install.

Most of these sites currently store their data in relational databases and therefore the task of expressing information from these sites in RDF can be viewed as a special case of expressing relational database data in RDF. The characteristics described above may make it challenging to use other approaches such as direct mappings from relational databases to RDF because of a high risk of database structure changes, limitations of web hosting provides, etc. At the same time the extensible, open-source nature of these content management systems make it possible to access data at a higher abstraction level, using existing application logic.

3. Methods for Exposing Relational Databases in RDF

We will consider two approaches for exposing relational databases in RDF – direct (mappings from relational database schema) and indirect (using the application logic to access data).

Direct Mapping. Most of existing work for exposing relational databases in RDF [Bizer2006, Erling2006, TimBL2006], consider direct mappings from relation database schema to RDF. This generic approach can be useful in many cases, but in the case of online community sites it may lead to difficulties in keeping up with changes in the database structure and also in installation and use by inexperienced users.

Using the Application Logic to Access Data. In this position paper we describe an approach adopted by many SIOC export tools – using APIs and the application logic provided by content management systems as a source of information to be exported in RDF. Authors of CMS software are encouraging developers of plugins to use these APIs and not to access the database directly. By doing so the developers of RDF export applications are shielded from any changes to the storage layer as long as the interface remains the same, and can deploy their applications as simple plugins for content management systems.

In a generic scenario you may not always have a choice and direct access to the database can be the only solution. The characteristics of web applications described above enable another option by providing APIs and function calls for access to data. By using application logic to retrieve data we can also make use of caching, data access protection and other functionality built into the application.

To further abstract from specific solutions we can consider two choices – using direct access to the database vs. using the application logic to access data, and declarative vs. procedural method of converting data to RDF:

	Declarative	Procedural
Direct (database)	I	II
Indirect (application logic)	III	IV

Most of existing methods using direct mappings from database schema to RDF correspond to Quadrant I on this table. A generic problem setting allows to create generic solutions and separate software doing the mapping from the mappings themselves. Some of the existing solutions may also be in Quadrant II, but this distinction is not of particular interest for this paper.

Quadrant III (a hybrid approach) - generating RDF by declaring rules for mapping data accessed via the application logic - may be interesting for future exploration.

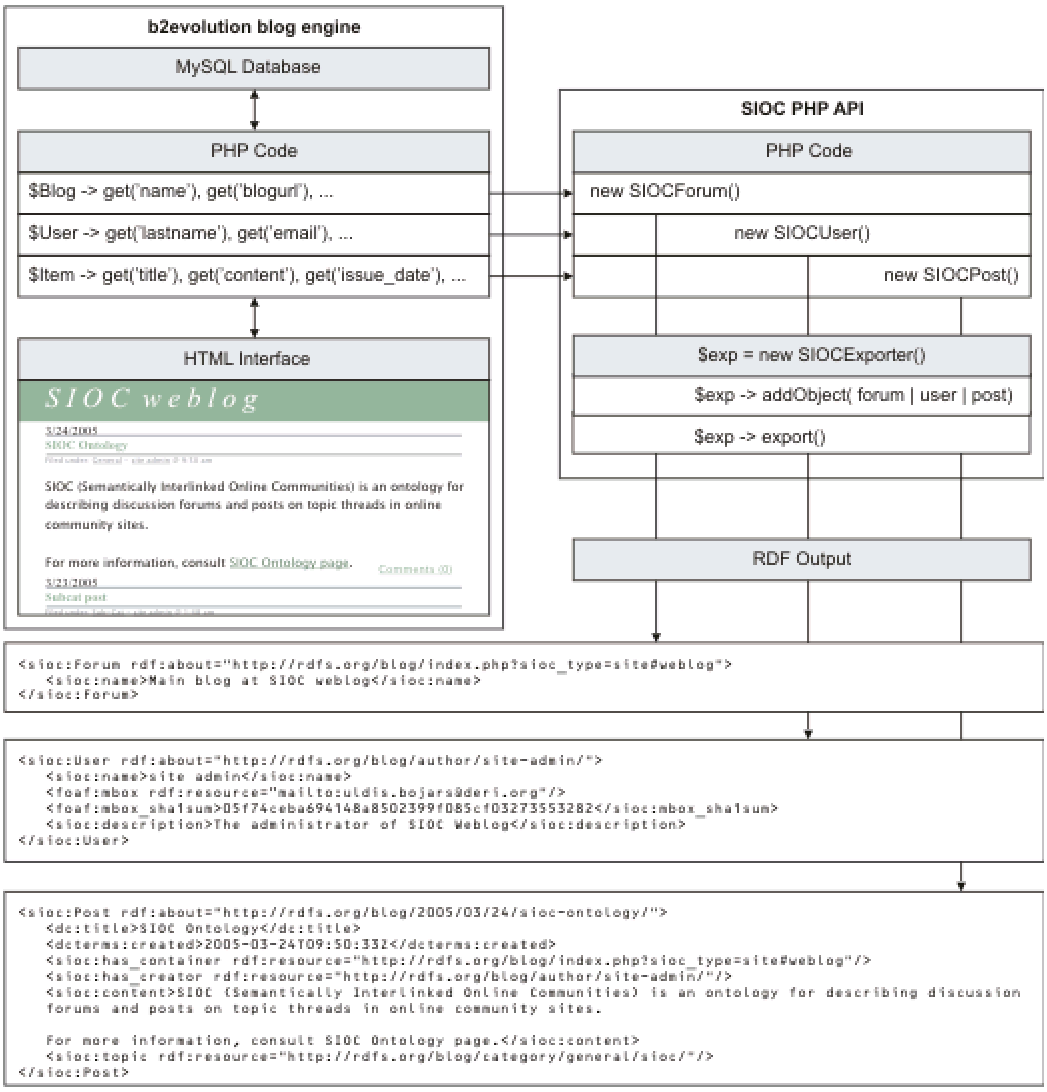
Quadrant IV represents the method described in this paper – using a procedural approach and existing application logic to access data. This indirect approach is application specific (unless there is a standard API used by all applications) and can be more difficult to express in a declarative form.

While direct mappings can achieve greater performance they may require more maintenance because they do not use the application logic and database access abstraction layer already provided by many applications.

4. Example: SIOC export plugin for b2evolution

The SIOC initiative provides on the one hand a SIOC Core Ontology (W3C member submission published on July 31, 2007 [1]) that can be used to describe information about contents and structure of online community sites in RDF, and secondly, several SIOC export applications for blogs, forums, and mailing lists [2]. SIOC API for PHP was also created in order to make development of such SIOC plugins and exporters as easy as possible. It shields developers from technical details of how information is represented in RDF – they are operating at the level of SIOC concepts instead. Thus, developers only have to extract content from the database (handled by the internal logic of CMS) and pass it to the API that will render RDF data.

The architecture of b2evolution's SIOC export plugin (Figure 1) illustrates the application logic based approach. Information is contained in a MySQL database. The plugin uses existing b2evolution's functions to access information in the database, which is then passed to a SIOC export API for PHP to generate RDF/XML output.


b2evolution blog engine
MySQL Database
PHP Code
$Blog -> get('name'), get('blogurl'), ...
$User -> get('lastname'), get('email'), ...
$Item -> get('title'), get('content'), get('issue_date'), ...
HTML Interface
SIOC weblog
3/24/2005
SIOC Ontology
SIOC (Semantically Interlinked Online Communities) is an ontology for describing discussion forums and posts on topic threads in online community sites.
For more information, consult SIOC Ontology page.
Comments (0)
3/23/2005
Subcat post
SIOC PHP API
PHP Code
new SIOCForum()
new SIOCUser()
new SIOCPost()
$exp = new SIOCExporter()
$exp -> addObject(forum | user | post)
$exp -> export()
RDF Output

```
<sioc:Forum rdf:about="http://rdfs.org/blog/index.php?sioc_type=site#weblog">
   <sioc:name>Main blog at SIOC weblog</sioc:name>
</sioc:Forum>
```

```
<sioc:User rdf:about="http://rdfs.org/blog/author/site-admin/">
   <sioc:name>site admin</sioc:name>
   <foaf:mbox rdf:resource="mailto:uldis.bojars@deri.org"/>
   <foaf:mbox_sha1sum>05f74ceba694148a8502399f085cf03273553282</sioc:mbox_sha1sum>
   <sioc:description>The administrator of SIOC Weblog</sioc:description>
</sioc:User>
```

```
<sioc:Post rdf:about="http://rdfs.org/blog/2005/03/24/sioc-ontology/">
   <dc:title>SIOC Ontology</dc:title>
   <dcterms:created>2005-03-24T09:50:33Z</dcterms:created>
   <sioc:has_container rdf:resource="http://rdfs.org/blog/index.php?sioc_type=site#weblog"/>
   <sioc:has_creator rdf:resource="http://rdfs.org/blog/author/site-admin/"/>
   <sioc:content>SIOC (Semantically Interlinked Online Communities) is an ontology for describing discussion
   forums and posts on topic threads in online community sites.

   For more information, consult SIOC Ontology page.</sioc:content>
   <sioc:topic rdf:resource="http://rdfs.org/blog/category/general/sioc/"/>
</sioc:Post>
```

Figure 1: Architecture of the SIOC RDF export plugin for b2evolution

Sometimes an API call needed for the SIOC exporter is not provided by the CMS engine. Then we have to “fall back” to accessing the database directly. For example, SIOC export plugins for b2evolution and WordPress each contain one direct query to the database, in both cases having to do with user account information. In our experience such code is the first to break as the CMS engine evolves. A possible solution is to eliminate direct database access by asking CMS developers to include API calls for requesting the required information.

5. Conclusion

This position paper described an approach adopted by many SIOC RDF export tools – using the application logic provided by content management systems in order to access data stored within a relational database and generate RDF data. When compared with a traditional approach of creating mappings from relational database schema to RDF, using the existing application logic ensures a tighter integration with a content management system, an increased resistance to database schema changes as the software evolves and allows us to use existing facilities of CMS engines such as caching and access control.

Web applications such as online community site engines can be regarded as a special case where function calls are well documented and development of extensions is encouraged. This enables us to use existing API calls when exporting data in a machine-readable form. Two different approaches (direct and indirect) for generating RDF gives us a choice and further research may be needed to determine what is the best decision in each situation.

We only looked at exporting specific information contained within online community sites and did not aim at answering arbitrary queries over RDF. When in need to answer arbitrary SPARQL queries, a hybrid approach - using the application logic to access the database and defining mappings from these API calls to RDF - may be interesting.

Acknowledgments

This material is based upon work supported by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131. We gratefully acknowledge Conor Hayes for his feedback on this paper.

References

[Bizer2006]
C. Bizer, R. Cyganiak, J. Garbers, O. Maresch, editors. "D2RQ V0.5 - Treating Non-RDF Relational Databases as Virtual RDF Graphs", 2006, http://www.wiwiss.fu-berlin.de/suhl/bizer/d2rq/spec/

[Breslin2005]
J.G. Breslin, A. Harth, U. Bojārs, and S. Decker. "Towards Semantically-Interlinked Online Communities". In The 2nd European Semantic Web Conference (ESWC ’05), Heraklion, Greece, Proceedings, May 2005, http://sioc-project.org/publications.

[Erling2006]
O. Erling, I. Mikhailov. "Mapping Relational Data to RDF in Virtuoso", OpenLink Software, 2006, http://virtuoso.openlinksw.com/wiki/main/Main/VOSSQLRDF

[TimBL2006]
T. Berners-Lee. "Relational Databases on the Semantic Web", Design Issues for the World Wide Web, 2006, http://www.w3.org/DesignIssues/RDB-RDF

[1] Semantically-Interlinked Online Communities (SIOC) Ontology Submission Request to W3C

[2] SIOC Ontology: Applications and Implementation Status