Proceedings of the

SPDECE-2012. Ninth nultidisciplinary symposium on the design and evaluation of digital content for education

13–15 June 2011

Universidad de Alicante

Alicante, Spain

Edited by

Manuel Marco Such

Pedro Pernías Peco

Printed in Alicante, Spain

Available online at www.sciencedirect.com
ScienceDirect
Procedia - Social and Behavioral Sciences 00 (2011) 000–000
Procedia Social and Behavioral Sciences
www.elsevier.com/locate/procedia

Spdece 2012

"A Linked-Data based approach for managing digital libraries"

Gustavo Candela Romero[a] , María Pilar Escobar Esteban[a], Manolo Marco Such[a], Pedro Agustín Pernías Peco[a], Alexander Sánchez Díaz[b,*]

"aUniversity of Alicante, Carretera de San Vicente del Raspeig, San vicente del Raspeig 03690, Spain"
"bUniversity of Havana, Havana, Cuba"

Abstract

Nowadays, the social networks are spreading abroad different application domains. Also, the digital libraries are improving how their users exploit the catalog services with social capabilities. More recently, the Linked Data model defines a novel vision for the near future web applications. Our paper approaches a solution for managing users in a digital library with a semantic fashion.

"Keywords: RDF; social graph; opensocial; semantic; FOAF;"

1. Introduction

The impact of social networks has incremented the usage of advanced technologies like OpenSocial, gadgets and personalized vocabularies like Friend of a Friend (FOAF). Digital libraries researchers have been moving towards approaches supporting services with semantic parameters as annotations, recommendations, rankings, comments and self defined tags. Commonly, catalogues with these capabilities are called Social Online Public Access Catalogues (SOPAC). Many front end applications like Drupal are extended with modules offering this functionality. Recently, the Linked-Data approach was proposed as a reference model for applications that expose a semantic behaviour with content accessible by Uniform Resource Identifiers (URI). The Resource Description Framework specifications from W3C are a building block for developing applications guided with linked data principles. As example, the JeromeDL digital library uses Semantic Web and Social Networking for allowing users to control their bookmarks and annotations and share them with friends.

* *E-mail address*: gcandela@dlsi.ua.es, pescobar@dlsi.ua.es, marco@dlsi.ua.es,ppernias@dlsi.ua.es, alezsd@yahoo.com.

OpenSocial is based on a client-model schema for the development of software components encapsulated in XML files. This kind of applications requires high traffic between the server and the client requesting data processing from server side. In some cases, this is done with high latency responsiveness time. For this reason, a new type of applications is required for improving the user experience. Rich Internet Applications are a kind of web applications with enhanced client side functionality. Some tools are based on model-driven techniques which integrate methodologies for developing social RIA applications.

The above mentioned technologies arises a new concept of the Web 3.0 as a social and semantic web, providing information that both users and machine can understand. It is a challenge, the development of new techniques that incorporate the actual digital libraries systems into the web 3.0 with a minimal effort. There are different RDF-based data models to manage digital resources and its relations in a library and educational context. Functional Requirements for Bibliographic Records (FRBR) is a example of a model for these purposes that includes any kind of digital resource as bibliographic records, SCORM or LOM. Nevertheless, the person profiles and their social relations are not included in these models.

To our best knowledge, there is a lack between OpenSocial and semantic web. It should be reasonable to extend the well adopted OpenSocial concepts, like person, groups or activities, in a RDF controlled vocabulary. In this paper, we propose an ontology that describes new concepts based on OpenSocial and other vocabularies like FRBR, FOAF and RDFS. This ontology is intended to develop RIA applications combining traditional digital libraries functionalities with semantic support. The proposed approach has an activity as a main concept where entities like person, group or work could be actors.

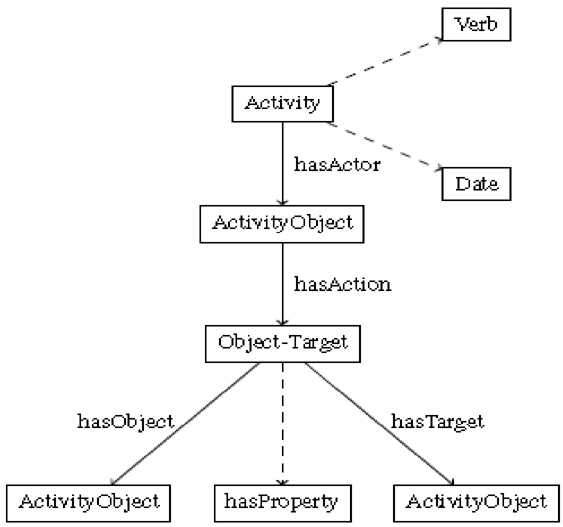

Fig. 1. Activity diagram

Clearly, the model can be extended to other applications domains applying different flows of activities. The following section describes the activity skeleton of the social vocabulary. The section III shows how it is used the approach in a study case scenario. The section IV explains both the relational and semantic models. How to query and insert data in the RDF graph using SPARQL language is explained in section V. Finally, the main results are summarized and the work in progress is mentioned.

2. Social vocabulary

The proposed ontology is mainly focused on OpenSocial resources: user, activity, group or application data. Also, it uses FOAF agent class and extends *knows* relation. XML Schema is included for typed literals. OpenSocial describes an activity with four elements:

- Actor: the entity that creates the activity.
- Action: the event performed by the actor.
- Object: the entity on which the action has been performed.
- Target: the entity receiving the action impact.

How to translate four items using triples arises while trying to represent the above resources in RDF. A naive solution is depicted in the Figure 1 using empty RDF nodes. In our approach, the *ActivityObject* can be any entity involved in the activity. The element Object-Target represents a generic binary relation which could be extended depending on the application domain. For instance, in our case an Object-Target can be a Work-Person, joining a Work with a Person. This abstract concept of the activity provides an easy artifact for extending the model to any domain.

3. Case of study

Virtual Library Miguel de Cervantes (BVMC, 1999) is a digital library focused on the digitization and public exhibition of classic works of Spanish literature. The project dates back to 1999, originated at the University of Alicante and is sponsored by the Banco de Santander. Today the Fundación Miguel de Cervantes Virtual Library, chaired by writer Mario Vargas Llosa, is responsible for defining its overall strategy and work plan.

The BVMC has digitized more than 150,000 works, 15,000 of which are documents scanned as TEI XML or HTML, making them available in other electronic formats. The rest, mainly consisting of PDF, is distributed as such in facsimile format.

A first version of social management based on Apache Shindig and OAuth was developed for the BVMC. Apache Shindig is the reference implementation of OpenSocial API specifications, a standard set of Social Network APIs providing the code to render gadgets. OAuth is used for delegating the authentication context. Both protocols were combined to aim social services with multimodal access capabilities. This model is based on the Client-Server architecture what causes a high consume of resources in the server side and hence communication delays. Our approach extends the open social model embedded in the XHTML code with the semantic information displayed by the browser. This way, the communication is optimized reducing the activity between the client and the server.

The ontology proposed fits perfectly in the BVMC providing new services based on social networks. Basically, the goal was to manage in the BVMC domain the concepts of Person and Activity according to the Linked-Data design. Combining all the works digitalized by the BVMC with these new social concepts, it is possible to create a new version of the BVMC based on the Semantic Web.

This new version of the BVMC includes comments, votes or recommendations allowing users to share their knowledge within a social network. For instance, a person can vote or comment a specific work and this information can be used by another user for other purpose encouraging the collective knowledge. These events are registered as activities of different kind that are notified to different users according to their circle of friends. Every user has a dashboard where its activities are stored as a collection of events.

The Figure 2 shows how an activity can be created giving values to every concept. It illustrates a sub property *hasRecommended* where a user recommends a FRBR work to another one. With this model, the catalog services can be extended with social parameters like votes and summaries.

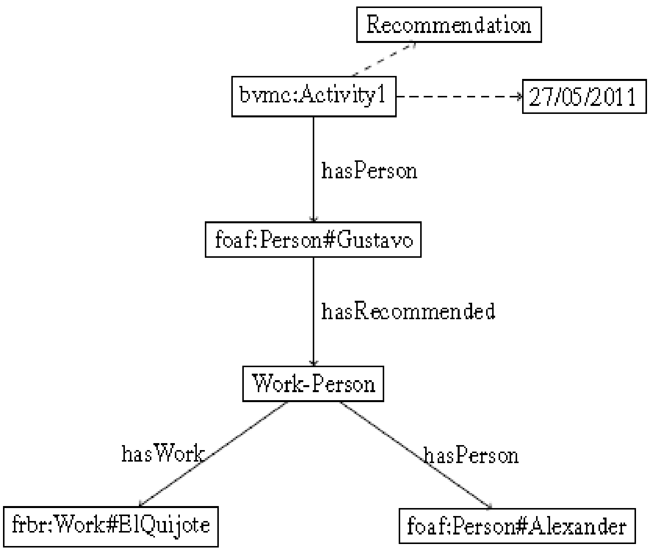

Fig. 2. Activity example

As it is shown in our approach, when it comes to encourage the collective intelligence, the person stays in the background while the main concept becomes the activity.

The new knowledge is created by the combination of several activities performed by the users. Moreover, our abstract activity conception implies that this generic concept can be extended to any educational domain based on activities.

The information can be consumed by the traditional model Client-Server as well as directly by the browser. This new functionality is provided by XFN (The XHTML Friends Network) micro formats. XFN outlines the relationships between individuals by defining a small set of values that describe personal relationships. In HTML and XHTML documents, these are given as values for the *rel* attribute on a hyperlink.

Based on Semantic Web and SOPAC, our approach has the ability to customize the user experience via the administrative control panel and takes advantage of online, web 2.0-like interaction to build collective intelligence.

4. Generating a relational database from the semantic social schema and vice versa

RDF schemas and instances can be efficiently accessed and manipulated in main memory. For persistent storage the data can be serialized to files, but obviously, for large amounts of data the use of a database management system is more reasonable and effective. Currently existing RDF stores, as D2RQ

Platform, are using relational database management system to persist the data treating Non-RDF Relational Databases as Virtual RDF Graphs. Storing RDF data in a relational database requires an appropriate table design. There are different approaches that can be classified in generic schemas that do not depend on the ontology and ontology specific based schemas. The simplest generic schema is the triple store with only one table containing the columns Subject, Predicate and Object. The greatest disadvantage of this schema is that performing a query means searching the whole database and join queries become very expensive. As a matter of fact, this schema can be improved providing two more tables to store resource URIs and literals separately as shown in table 1, 2 and 3.

On the other hand, the basic ontology specific schema consists of one table with one column for the instance ID, one for the class name and one for each property in the ontology. Naturally, each row corresponds to one instance.

Table 1. Triplets

Subject	Predicate	isLiteral	Object
R1	R2	false	R3
R1	R4	true	L1
…	…	…	…

Table 2. Resources

ID	URI
R1	…#1
R2	…#2
…	…

Table 3.Literals

ID	URI
L1	Value
…	…
…	…

This schema can be improved including one table per property being known as hybrid schema (table 4). In that case, changes to the ontology do not require changing existing tables, as adding a new class or property results in creating a new table in the database.

Table 4. Hybrid schema

Class A	Property 1		Object
ID	Subject	Object	ID
…#1	…#1	..#3	#3
…	…	…	…

Based on the hybrid schema described in the previous paragraphs, our approach has one table per class and one per property to store RDF triplets. The Figure 3 represents an *ActivityObject* as a Work or a Person that creates the Activity.

The Person entity contains all the fields described in FOAF vocabulary while the Work entity contains all fields described in FRBR vocabulary. The inheritance is represented by one table per subclass as is shown in Person and Work that are subclasses of *ActivityObject*. The Activity entity has an actor referenced by *ActivityObject* entity.

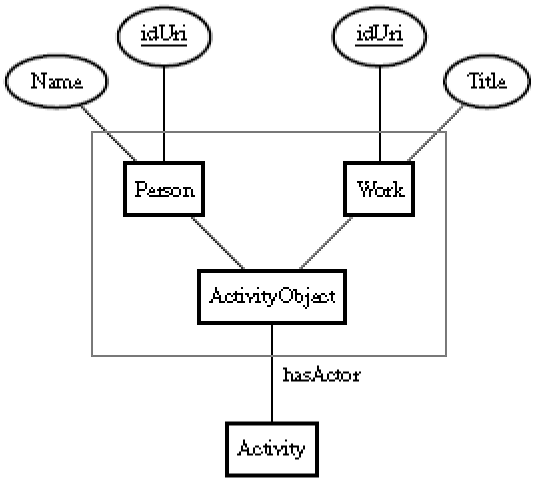

Fig. 3. Activity

The activities are stored in different tables depending on its type. The property *hasAction* and each subclass have their own table like *hasReviewed* or *hasRecommended*. In the same way, the Object-Target class and all the combinations have their own table. As is shown in Figure 4, the main benefit of this generic schema is that creating new types of actions can be done adding just a table linked to *hasAction*. Every Activity is linked to a Person using the dashboard entity as is shown in Figure 5. The Dashboard has a Visibility to ensure the security.

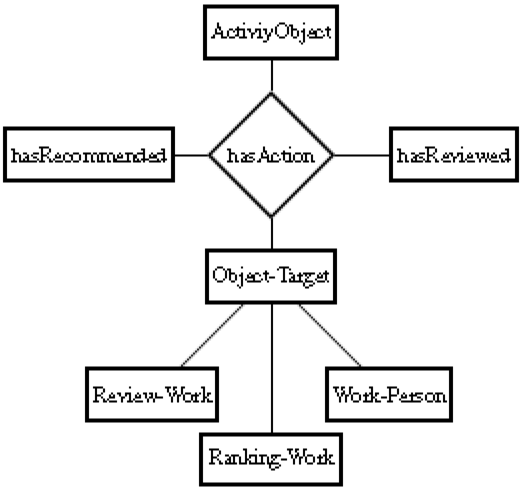

Fig. 4. Activity Schema

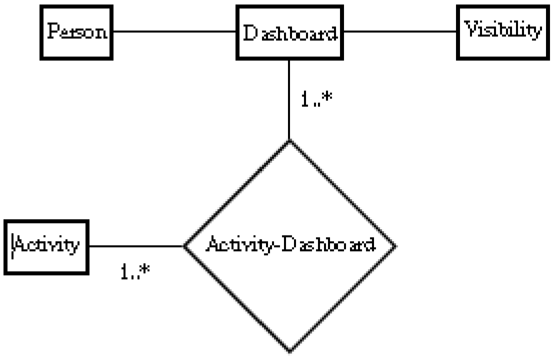

Fig. 5. Dashboard

4.1. Mapping database model to RDF Graph model

Although a relational database is necessary for physical storing purposes, a logical representation in a RDF graph model is also required. Due to a lot of information exist in relational databases, a plethora of tools are developed to generate analogous RDF representation. In this sense, the D2RQ platform is used to generate an RDF graph from a relational database. It is based on a mapping file which details the relation between graph nodes, tables and columns names. Although, it is possible to generate this file automatically only using the relational model, also it could be restricted by an ontology vocabulary. This last case was used in our approach. From the D2RQ perspective, this file is written using a RDF representation named N3 annotation which uses the abbreviation “a” for rdfs:type. The file should be headed with namespaces prefixes and a mapping to the database system with the appropriate driver description, authorized credentials and optional parameters about the connection session.

There are two types of objects for describing the mappings: d2rq:ClassMap and d2rq:PropertyBridge. The first type is used to relate tables' names with RDF classes while the tables' columns are related with object properties in the ontology using the last one. In the case of tables representing many to many relations, the appropriated joins involving the foreign keys are necessary. To illustrate this, the following shows two mappings of the N3 file for the Activity resource. The first one maps the Activity table from the database to the Activity class of the ontology. And the second one, maps the date column of the Activity table to the corresponding data property of the ontology.

```
map:Activity a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern "@@Activity.idUri@@";
d2rq:class bvmc:Activity;
.
map:activity_date a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Activity;
d2rq:property bvmc:date;
d2rq:column "Activity.date";
.
```

5. Querying with Sparql

As Semantic Web technologies are getting mature, there is a growing need for RDF applications to access the content of huge, live, non-RDF, and legacy databases without having to replicate the whole database into RDF. The D2RQ Platform enables applications to access these graphs through the Jena API, as well as over the Web via the SPARQL Protocol and as Linked Data.

The ModelD2RQ class provides a Jena Model view on the data in a D2RQ-mapped database. The following example shows how a ModelD2RQ is set up using a mapping file and how Jena API calls are used to extract information from the model.

```
/ / Set up the ModelD2RQ using a mappingfile
ModelD2RQ model1= new ModelD2RQ(
"src/test/resources/mappingtest.n3" ,
"N3" ,
"http:/ /www.exampl eUrl.com" ) ;

StringSparql =
"PREFIX db : <http:/ /www.exampleUrl.com/ Dashboard#>"
+
"PREFIX bvmc : <http:/ /www.exampl eUrl.com/#>" +
"SELECT DISTINCT ? hasValue WHERE f" +
" db : 1 bvmc : hasActivity?hasValue . " +
"g" ;

Query q = QueryFactory.create(sparql);
ResultSetrs = QueryExecutionFactory.
```

```
create ( q, model1 ) .
execSelect ( ) ;
while ( rs.hasNext( ) )
{
	QuerySolutionrow = rs.nextSolution( );
	log.debug ( row.getResource ( "?hasValue " ) ) ;
}
```

6. Conclusion and ongoing work

The proposed approach can be used to build applications exploiting social networks to support functionalities like those include in a social OPAC. Linked-Data as a foundation for the Web 3.0 applications was used for delivering more specialized services implementing these functionalities. Hence, resources are accessible by URIs and could be interpreted also by browsers. With this, the response time of the services was reduced with the communication latency. The social concepts are well defined as classes and predicates in an ontology. This reuses the actual FOAF vocabulary and represents using RDFS the resources in the OpenSocial users graph. In our model, the Activity resource class constitutes an important type of node through social applications exhibit its behavior and processes flow. Around Activity resources an application which manages users can support personalized services from their feedback. There is an evolving integration between the services behavior and the users' experience. It is possible to add more domain specific classes and properties from the general activity skeleton. In the above study case, it is showed how specific relations between different entities like Work and Person were represented as subclasses of the Object- Target class. Similarly, application events can be represented by sub properties from *hasAction* (e.g. *hasRecommend*). Now, we are trying to generate automatically the mapping file and the relational model from the ontology, using the XSLT transformation language. With an RDF representation of the social graph the clients can be provided with attributes or micro formats embedding social metadata and profiles. For these reasons, it will be explored how to generate efficiently the corresponding attributes values from a user social profile.

References

Google. (2008) Opensocial specifications. [Online]. Available: http://www.opensocial.org

K. Antelman, E. Lynema, and A. Pace, "Toward a 21st century library catalog," Information Technology and Libraries, vol. 3, no. 25, pp. 128–139, 2006.

T. B. Lee. (2011) Linked data - connect distributed data across the web. [Online]. Available: http://linkeddata.org

C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," Semantic Web Information Systems, vol. 3, no. 5, pp. 1–22, 2009.

W3C. (2008) Resource description framework specifications. [Online]. Available: http://www.w3.org/RDF/

S. R. Kruk and T. Woroniecki and A. Gzella and M. Dabrowski, "Jeromedl - a semantic digital library," in Proc. of the Semantic Web Challenge in conjunction with ISCW, Valencia, Spain, March 2007.

J. M. Hermida, S. Meliá, A. Montoyo, and J. Gómez, “Developing rich internet applications as social sites on the semantic web: a model-driven approach,” IJSSOE, vol. 2, no. 4, pp. 21–41, 2011.

I. Davis and R. Newman. (2009) Expression of core FRBR concepts in RDF. [Online]. Available: http://vocab.org/frbr/core.html

Apache. (2011) Shindig project. [Online]. Available: http://incubator.apache.org/shindig/

OAuth. (2008) Oauth protocol specifications. [Online]. Available: http://oauth.net/core/1.0

A. Sánchez, L. Iriarte, and M. Marco, “Towards a more social digital library,” in Proc. of INTED2009 Conference, Valencia, Spain, March 2009.

XFN Microformats. (2011) XFN Microformats. [Online]. Available: http://microformats.org/wiki/xfn

D2RQ. (2008) D2rq homepage. [Online]. Available: http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/index.htm

Alice Hertel, Jeen Broekstra, and Heiner Stuckenschmidt. RDF Storage and Retrieval Systems. Available: http://ki.informatik.uni-mannheim.de/fileadmin/publication/Hertel08RDFStorage.pdf