INTEGRATING THE JENA RDF API WITHIN SAKAI : TOWARDS A SEMANTIC COLLABORATING LEARNING ENVIRONMENT

E Bodéré, P Clément, A Genoud, M Le Trocquer, M Moras, L Pochard, Vincent Ribaud, Philippe Saliou

INTEGRATING THE JENA RDF API WITHIN SAKAI : TOWARDS A SEMANTIC COLLABORATING LEARNING ENVIRONMENT

INTEGRATING THE JENA RDF API WITHIN SAKAI : TOWARDS A SEMANTIC COLLABORATING LEARNING ENVIRONMENT

E. Bodéré
P.Y. Clément
Master of Software Engineering by Immersion
erwan146@caramail.com
p_clement@hotmail.com

A. Genoud
M. Le Trocquer
Master of Software Engineering by Immersion
genoud_aude@yahoo.fr
letrocquermickael-iup@yahoo.fr

M. Moras
L. Pochard
Master of Software Engineering by Immersion
maxime.moras@libertysurf.fr
Laetitia0907@msn.com

V. Ribaud
Ph. Saliou
Computing Department
University of Brest
ribaud@univ-brest.fr
saliou@univ-brest.fr

ABSTRACT

In this paper, we describe a semantic extension to the CLE Sakai. Semantic means the ability to create, retrieve, query and navigate with knowledge about the entities managed by Sakai. RDF is the support of this semantic extension, through the integration within Sakai of Jena, an open-source RDF API provided by HP laboratory. New features are briefly presented and some use cases discussed. Open issues concludes this on-going work.

Keywords

RDF, semantic web, CLE, Sakai.

1. INTRODUCTION

Sakai is an open-source course management system (CMS). Described as the “digital home of students’ coursework” [1], the CMS is becoming the primary mechanism for faculty to provide students with scholarly information and resources, and in turn, for students to access such materials.

RDF is a W3C standard intended to manage metadata. RDF models metadata as 3-tuples which assert that some resource (identified by URI) has some property (identified by URI) which has a value identified either by URI or given literally. The most common day-to-day use of metadata is to help us find things but it can change the way to organize, manage and use things

Social software supports activities in digital social networks. Knowledge sharing and exchange is an important part of these activities. Because knowledge management and learning are closed, the use of social software can support learning. Hence, we investigated the Sakai tools able to support social activities for knowledge management such as resources sharing, “wikiing”, blogging and so on. Coupled with RDF metadata, these tools offer new features: providing additional semantic to social usages add significant value allowing view, navigation and query along semantic rather than simply chronological connections.

2. COLLABORATION AND LEARNING ENVIRONMENT

2.1 Sakai tools

Sakai defines itself as an online Collaboration and Learning Environment. Many users of Sakai deploy it to support teaching and learning, ad hoc group collaboration, support for portfolios and research collaboration.

The Resources Tool is a core feature of Sakai that allows an instructor to assemble a collection of documents or other files supporting a course and organize them into folders and subfolders for access by students. When an instructor creates a new resource item in Sakai, he or she can choose to upload an existing file from his or her local system, author a text document or HTML page directly through the Sakai interface, or link to existing content via a URL. The instructor can then enter Dublin Core metadata for the item (as of Sakai 2.0), along with information on copyright and access permissions. Once a resource item has been created, it can easily be selected and attached to postings within other Sakai tools, including Wiki and Blog tools.

2.2 An example of RDF application

RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”). RDF is a model describing qualified (or named) relationships between two (Web) resources, or between a Web resource and a literal. Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications.

Let us take an example from a W3C Note Describing and retrieving photos using RDF and HTTP. This note describes a project for describing & retrieving (digitized) photos with (RDF) metadata. The pictures are digitized and stored as JPEG images. Metadata is written into the pictures with the data-entry program (and can also be edited if corrections are necessary). Requests from the Web are served by a server, by sending either the picture or the metadata, depending on the form of the request.

The metadata is separated into three different schemas:

Dublin Core schema[1]. The Dublin Core schema [2] is a general schema for identifying original works, typically books and articles, but also films, paintings or photos. It contains such properties as creator, editor, title, date of publishing and publisher.

Technical schema[2]. This schema captures technical data about the photo and the camera, such as the type of camera, the type and the date of film, etc.

Content schema. This schema is used to categorize the subject of the photo by means of a controlled vocabulary. This schema allows photos to be retrieved based on such characteristics as portrait, group portrait, landscape, architecture, sport, animals, etc.

This is an excerpt of some metadata in RDF

<rdf:Description rdf:about="">

<s0:creator>Bert Bos</s0:creator>

<s0:format>image/jpeg</s0:format>

<s0:date>1999-06-26</s0:date>

<s1:camera>Canon Eos 5</s1:camera>

<s0:title>Marian with sheep</s0:title>

<s0:subject>Landscape, Animal</s0:subject>

<s0:description>Marian brings the sheep to the field in the morning. The lamb she carries was born that night.</s0:description>

</rdf:Description>

This means that the resource about we are talking on, was created by Bert Bos, is a jpeg image ...

2.3 Semantic annotation of resources

An annotated system is a system which "knows about" its own content in order that automated tools can process annotations to improve the use of the system. The goal is to create annotations with well-defined semantics. For example, the semantic annotations can identify people, organizations, and projects, as well as including traditional metadata, such as the author's name and date of publication. When these statements are integrated within a RDF repository, we can then support SPARQL queries like "give me all the people which work in projects on the Semantic Web". According to the features of the Semantic Blogger [3], the software will exploit the semantic annotations to provide three key behaviours:

Semantic View: context sensitive, schema driven views of the content (over and above fixed templates).

Semantic Navigation: new navigation modalities (over and above unlabelled links).

Semantic Query: richer query and discovery mechanisms (over and above free text search).

Uren and al. [4] provide a complete survey of semantic annotation for knowledge management.

3. A SEMANTIC EXTENSION OF SAKAI

3.1 Overview of the project

3.1.1 The team

Thanks to a strong partnership with Thales company, Brest University offers a kind of apprenticeship during the last year of a Master of Software Engineering. Young engineers are immersed in an imitation of the real world. The plan of action is built on a 7-month project, followed by a 5-month internship. Young engineers make up two teams of 6; each team is led by one associate professor acting as project manager. A contract defines the customer-supplier relationship. An adaptation of the Thales ISO 9001 baseline is the project baseline. Each project team has its own office with individual working post and common installations. Each project team uses a different and complete software engineering tools suite (Oracle and Open Source). At the end of the academic year, the team members obtain a Master's degree in Software Engineering. This year, one team's project aims to provide a semantic extension to Sakai, called Sakai Semantic Edition (Sakai SE).

3.1.2 Project objectives and outcomes

Information Management can be defined as the set of activities that people perform in order to acquire, organize, maintain and retrieve information for everyday use. A space of information (SI) for a person or a group includes all the information items that are under that person's or that group's control. A SI also includes applications, tools (such as a presentation tool) and constructs (e.g., associated properties, folders) that support the acquisition, storage, retrieval and use of the information in a SI. Sakai offers several spaces of information: personal (MyWorkspace), shared by a group (Site), shared between two persons (DropBox).
The main goal of Sakai SE is to enhance these SI with semantic annotations. Main functions are:

a) Indexation allows to describe the information item with a set of metadata (called a notice) in order to exploit the notice without the need to access to the item ;
b) Search on metadata

[1] Referred as s0 in the example above

[2] Referred as s1 in the example above

c) Semantic hyperlinks between notices and/or items
d) Navigation offers knowledge walkthrough along the semantic paths
e) RDF modelization defines a common sense
f) Automatic extraction provides RDF metadata
g) Formalization of knowledge through the use of vocabularies and schemas

3.2 Jena : a Semantic Web Toolkit

Jena is a leading Semantic Web programmers' toolkit. It is an open-source project, implemented in Java, and available for download from SourceForge.

3.2.1 Jena overview

This overview is based on the Jena overview in [5].

Jena offers a simple abstraction of the RDF graph as its central internal interface. The main contribution of Jena is a rich API for manipulating RDF graphs. RDF graphs in memory or in databases. Jena provides additional functionality to support RDFS and OWL.

The two key architectural goals of Jena are:

• Multiple, flexible presentations of RDF graphs to the application programmer. This allows easy access to, and manipulation of, data in graphs enabling the application programmer to navigate the triple structure.

• A simple minimalist view of the RDF graph to the system programmer wishing to expose data as triples.

An simplified overview of the Jena architecture is shown in Figure 1. Applications typically interact with an abstract Model which translates higher-level operations into low-level operations on triples stored in an RDF Graph.

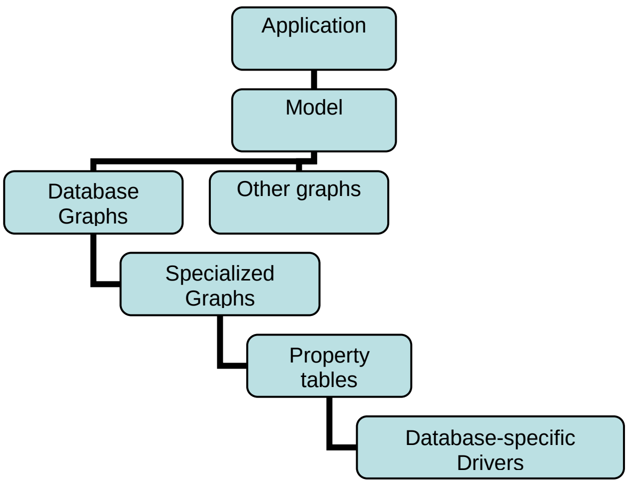

Figure 1 : Jena2 Architectural Overview

3.2.2 Jena storage

At an abstract level, the Jena storage subsystem need only implement three operations: add statement, to store an RDF statement in a database; delete statement, to remove an RDF statement from the database; and the find operation. The find operation retrieves all statements that match a pattern of the form <S,P,O> where each S, P, O is either a constant or a don't-care. Jena's query language, RDQL, is converted to a set of find operations in which variables are permitted in the find patterns. The variables enable joins across the patterns.

A widely used implementation technique is to store RDF statements in a relational database using a single statement table, often called a "triple-store." This is a table that stores each RDF statement as a row and has columns for the subject, predicate and object.

3.3 Components of Sakai SE

3.3.1 Schemas

A schema contains properties. RDF provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources. That is the role of the RDF vocabulary description language, RDF Schema. RDF Schema defines classes and properties that may be used to describe classes, properties and other resources.

The Dublin Core is used for all entities of Sakai SE. A technical schema groups properties other than Dublin Core properties (see §2.2). A technical schema may be present in several versions. Versions are managed through a schema group.

3.3.2 SPARQL queries

SPARQL is a query language for getting information from RDF graphs. SPARQL contains capabilities for querying by triple patterns, conjunctions, disjunctions, and optional patterns. Results of SPARQL queries can be ordered and presented in several different forms. Through the integration of the Jena SPARQL engine, Sakai SE offers triple querying and queries management.

3.3.3 Sakai tools

Sakai SE is essentially made of tools supporting the management of spaces of information (cf. §3.1.2).

Some tools were modified in order to incorporate RDF facilities, mainly the Resources tool and the Users tool.

New tools were developed including the Semantic Schema that allows schemas management and the Semantic Search that provides SPARQL queries use.

Sakai provides tools for content authors to represent structures that link together smaller units of content such as resources, pages into larger structures such as portfolios and sites. After an investigation of tools such as OSP portfolio, blogger and wiki, we select wikis because they can be seen as a lightweight platform for the management of these structures. One way to organize the organic growth of Wiki content is to add structure by enriching Wiki pages with RDF metadata. This adds significant value allowing semantic view, navigation and query.

Blogging is also a lightweight Web publishing paradigm that provides a very low barrier to entry but the components of the Blogger tool (blogs and entries) are not managed as Sakai entities and we were unable to bind these components with RDF URI (see §3.4.1 for details).

3.4 Some features of Sakai SE

3.4.1 Technology

Jena has object classes to represent graphs, resources, properties and literals. The interfaces representing resources, properties and literals are called Resource, Property and Literal respectively. Recall that in Jena, a graph is called a model and is represented by the Model interface.

RDF triples can be associated to different Sakai entities : sites, users, groups, content resources, wiki, wiki pages and so on. Each instance of these entities is associated within Sakai with a kind of internal identifier. In order to be able to write or read a triple related to one of theses instances, we have to bind the triple with the right instance of Sakai entities. Hence, for each instance, we build an URI based on the type of the instance (the entity its belongs to) and the proper value of this internal identifier. For example, the file Overview.doc (a ContentResource instance) has as an internal identifier the value *c40dd7fc-...508d* and its associated URI is *sakai:content: c40dd7fc-…508d*.

Hence, there is a one-to-one relationship between Sakai objects and RDF URI allowing to bind objects to the related graph (the graph having this URI as root).

3.4.2 Multi-dimensional search and browsing

Web information portals provide a point of access onto an integrated and structured body of information about some domain. Many portals use a hardwired navigation structure based on a single rich classification scheme (e.g. Dublin Core) coupled to hyperlinking of related items and free text search. Yahoo is a canonical example. Portals handling more structured information typically have some more richly structured internal descriptive schema (often directly mirroring the database schema used for storage) and offer a rich search interface which can exploit this schema. This allows search to be tied to specific facets of the descriptive metadata and to exploit controlled vocabulary terms - leading to much more precise searches [6].

In Sakai SE, we describe properties, relationships and classifications of the various information items via the content schemas. This allows the users to choose the structure of the portal and allows searches to exploit the structure of the ontology

3.4.3 Evolution and extension of information structure

Information requirements change over time. This sort of evolutionary change requires us to change the metadata and any associated database schema, not just the descriptive ontology. This can be complex. We need to permit metadata to be added, without invalidating existing metadata. This is greatly facilitated by use of RDF semi-structured data representation. RDF enables incremental additions of properties and relations by virtue of its property-centric, rather than record-centric, approach to representation [6].

3.4.4 Querying

Creating an intuitive and powerful Web tool for metadata querying is not an easy task, the more so if we want to develop this tool in a generic way, i.e. without 'hard coding' a particular metadata scheme into it. Moreover, the tool should be flexible in the sense that the query functionality itself must be configurable: the user interface should reflect the important metadata fields and provide a user friendly, yet powerful, mechanism to query the metadata instances.

For the moment, querying is done through a direct editing of SPARQL queries but it requires a SPARQL knowledge out of reach from end-users.

3.4.5 Automation

Easing the metadata acquisition bottleneck is the provision of facilities for automatic mark-up of resources. No one wants to spent time to produce metadata, especially if they are still present in some manner with the resources. Creators, dates, description, keywords are provided within most of resources format (word processors, spreadsheet, images …) and metadata are present in the Resources tool. Sakai SE provides an extractor that translates existing metadata in RDF triples.

4. PERSPECTIVES AND CONCLUSION

A special effort has to be done to provide a quasi natural language querying access to RDF metadata.

An example is Ginseng, a guided input natural language search engine for Querying Ontologies [7]. When the user starts typing, the system predicts the possible completions of what the user enters and presents the user with a choice popup box. Ginseng, thus, guides the user through the set of possible queries while avoiding ungrammatical queries. When a query is completed, Ginseng translates it into SPARQL statements. Ginseng is based on Jena API and it should be possible to incorporate it in Sakai SE.

Social software is used widely in organizational knowledge management and professional learning. Integrating RDF management into social software should lower the barriers between knowledge and learning management. Semantic blogging is a promised issue but the Blogger tool need to be revisited.

Sakai SE is a first step towards a semantic collaborating and learning environment. The entry-ticket for Sakai development is now paid and we will build new tools on these semantic foundations.

5. REFERENCES

[1] Lippincott, Joan K. (2005) "Net Generation Students & Libraries." In Diana Oblinger and James L. Oblinger (Eds.), Educating the Net Generation (p. 13.2). Boulder, Colo.: EDUCAUSE..

[2] Dublin Core metadata initiative. Dublin Core metadata element set, version 1.1. July 1999. Dublin Core recommendation. URL: http://dublincore.org/documents/1999/07/02/dces/

[3] Cayzer, Steve. Semantic Blogging: Spreading the Semantic Web Meme. XML Europe 2004, Amsterdam.

[4] Uren, Victoria; Cimiano, Philipp; Iria, José; Handschuh, Siegfried; Vargas-Vera, Maria; Motta, Enrico; Ciravegna, Fabio. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 4, No. 1. (January 2006), pp. 14-28

[5] Wilkinson, Kevin; Sayers, Craig; Kuno, Harumi; Reynolds, Dave. Efficient RDF Storage and Retrieval in Jena2. First International Workshop on Semantic Web and Databases 2003, Berlin.

[6] Reynolds, D; Shabajee, P. Semantic Portals – Requirements Specification. SWAD-Europe deliverable 12.1.5.

[7] Bernstein, Abraham; Kaufmann, Esther; Kaiser, Christian; Kiefer, Christoph. Ginseng A Guided Input Natural Language Search Engine for Querying Ontologies. Jena User Conference 2006, Bristol.