

Integration of domain and social ontologies in a CMS based collaborative platform

Luís Carlos Carneiro^{1,2}, Cristovão Sousa^{1,3} and António Lucas Soares^{1,2}
luis.c.carneiro@inescporto.pt, cristovao.sousa@inescporto.pt,
antonio.l.soares@inescporto.pt

¹ INESC Porto, Campus da FEUP, Rua Dr. Roberto Frias, 378, 4200 - 465 Porto, Portugal

² Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

³ Escola Superior de Tecnologia e Gestão de Felgueiras – Instituto Politécnico do Porto, Rua do Curral, Casa do Curral, Margaride, 4610-156, Felgueiras, Portugal

Abstract This paper describes an approach and an application for the Semantic integration of domain conceptualization in a socio-collaborative network platform. The platform, built based on the Drupal CMS framework, aims to offer SMEs possibilities to access specific knowledge by means of a collaborative community. A model to semantically express the socio-collaborative activities of the platform connected with the domain knowledge related to the project is presented. The model is based on existing W3C ontologies such as FOAF, SIOC and SIOC-types for the socio-collaborative semantics and SKOS to describe the domain knowledge. An architecture to support the model and related applications of the semantic metadata generated in the platform are also described.

1 Introduction

Semantic technologies are in intense development nowadays, aiming at being used both generically (web) and particularly by teams/organizations/networks for specialised business uses. The semantic web goal is transform the web from a linked document repository into a distributed knowledge base and application platform, thus allowing the vast range of available information and services to be more effectively exploited. In order to have new semantic-enable information systems there is the need to integrate specific semantic vocabularies with semantic artefacts, such as ontologies or taxonomies.

The work described in this paper is contributing to the H-Know⁴ (Heritage Knowledge), a European research project (2009-2011) in the area of the management of old building rehabilitation, restoration and maintenance. The project aims to develop a socio-collaborative platform where SMEs can share knowledge about restoration and maintenance activities, inducing learning and training of partners and collaboration amongst partners.

⁴ <http://www.h-know.eu/>

The H-Know platform is built according to a perspective of collaboration enabled by a social network approach. Users have a personal profile with their personal and professional information so as the Entities and Collaborative Places (ColPlaces) they are connected with as well as their partners.

Collaborative places have a inter-organizational collaboration perspective while Entities are used for intro organizational collaboration. The main objective of the ColPlace is to provide a mean where different experts can share information about the activities they are undertaking together.

Both ColPlaces and Entities have a set of tools for collaboration such as an Event Manager, a Gallery Manager, Pages, a Blog, Forums or a File Repository. We can see this tools as "Content Containers".

H-Know platform, is built over the Drupal CMS⁵. Drupal provides a wide number of extensions and customizations, mostly of them implemented as Modules. Some Modules were crucial for the H-Know implementation, such as CCK to define the different types of content in the platform or Organic Groups to create the groups and some specific modules in the area of web semantics, that will be described later on this paper.

In the H-Know platform is produced content which deals with a specific area of knowledge, represented in a knowledge structure. Most of the concepts of the knowledge structure are based on the CIK (Construction Industry Knowledge) ontology from the project Know Construct(2005-2007)⁶. For H-Know, some modules were eliminated, others were revised and some other were created, specific for the area of old building restoration and cultural heritage. In the diagram 1 we see the first level of concepts of our domain.

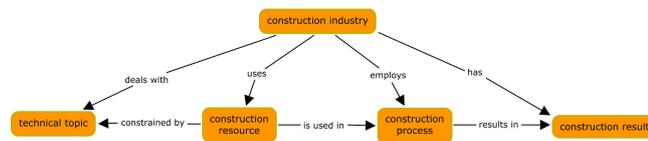


Figure1. H-Know knowledge domain, first level concepts

Basically, we can say that in the Construction Industry we apply Construction Resources to Construction Processes that will lead to Construction Results. Construction Resources are constrained by Technical Topics. This basic structure is based on ISO 12006-2 [8]. Each one of the concepts presented is extended in several sub-concepts which are not presented here.

In a platform with the characteristics described and with such a kind of knowledge structure, we want to be able to answer "Competency Questions" such as: "Which projects exist about the legal systems of a construction?" or

⁵ <http://drupal.org/about>

⁶ <http://www.know-construct.com/>

”Who is publishing more about the rehabilitation process of a bridge?”.

From the context described above we have a goal which is improving the knowledge organization and inference of the content produced in the platform, by semantically expressing the socio-collaborative activities held there connected with the domain knowledge of the H-Know project.

2 Design and Integration of Social and Domain Ontologies in the platform

2.1 An approach for the integration of semantics in a socio-collaborative platform

To build the integration of semantics in a already developed and published socio-collaborative platform, an approach with several steps is presented in the diagram 2.

The approach is organized in 3 different sets of steps, represented with different colours, with the labels of the relations in the diagram representing the order of the steps in the approach.

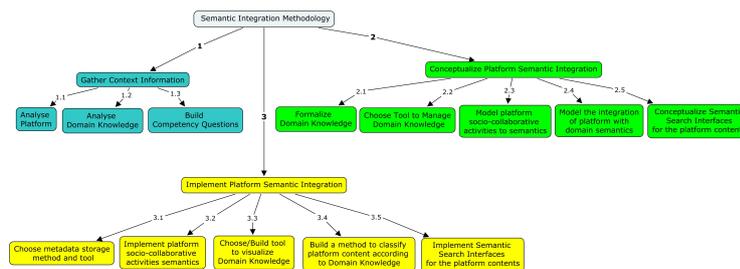


Figure2. Semantic integration approach

To integrate semantics in a socio-collaborative platform, first of all we need to understand the context of the problem. This means analysing the platform structure, purposes and background (1.1) and having a global vision about the domain knowledge managed in the platform (1.2).

These two steps allow us to build competency questions (1.3), essential to define the importance and the objectives of the semantics for the platform and to understand what kind of questions the semantic enhancing of the platform can answer that traditional ways of searching cant.

The conceptualization of the semantics in the platform can start after these base tasks are finished. We should start by defining an approach to formalize the Domain Knowledge (2.1) and a tool to build and manage that knowledge (2.2).

Then, picking up the structure of the platform, its actors and activities, a model to semantically describe the platform socio-collaborative activities should be designed (2.3). To relate the domain knowledge of the problem with the socio-collaborative activities, we need to build another model to integrate both semantics (2.4). The last step of this conceptualization process should focus on the designing of searching interfaces enhancing the semantic generated metadata of the platform (2.5), taking into account the “Competency Questions” previously formulated.

The last group of steps are intended to implement the conceptualizations previously done. First, the method and tool for the semantic metadata storage must be defined (3.1). Then, from the semantic model defined to describe the socio-collaborative activities, we should implement it in the platform (3.2), so it can generate and store semantic metadata. Since the platform has a specific domain knowledge managed there, a tool to visualize that domain must be chosen or built (3.3) together with a method to allow platform users to classify the content produced (3.4). The last step of all the process is making use of the semantic metadata generated, implementing the semantic interfaces (3.5) that were previously defined.

2.2 Formalization of the platform Domain Knowledge

To describe the domain knowledge previously presented in a formal way, **SKOS** data model, which is intended to represent Knowledge Organisation Systems (KOS) like thesauri, term lists and controlled vocabularies, was the chosen solution.

Using the SKOS data model to translate the domain knowledge, we can define each concept as an individual of the **skos:Concept** class and the **skos:narrower** and **skos:broader** properties, to construct the knowledge structure. For non-hierarchical linking we can use the property **skos:related**. When we have hierarchical transitive relations between the concepts we use the relational properties **skos:narrowerTransitive** and **skos:broaderTransitive**.

Using lexical labels of SKOS (**skos:prefLabel**, **skos:altLabel** and **skos:definition**) we give the exact meaning we want to each one of the concepts of the domain knowledge. Language tags are used to describe each concept in different languages, an important requirement of the H-Know project.

In the cases we need to define different properties than the ones offered in the core of SKOS, to express extra information, we specialize the SKOS model. Following the recommendations of SKOS primer documentation [4], SKOS allows an application designer to create new properties. For example:

```
hknow:employs rdfs:subPropertyOf skos:narrowerTransitive
```

An example of a concept described using SKOS data model can be:

```
hknow:Space rdf:type skos:Concept;  
skos:prefLabel "Space"@en;  
skos:prefLabel "Espaço"@pt;
```

skos:definition "A material construction result contained within or associated with a building or other construction entity"@en;
 skos:definition "Um resultado de construção material, contido ou associado a um edifício ou outra entidade de construção"@pt.

2.3 Semantic description of platform socio-collaborative activities and their connection with the domain knowledge

Picking up the main elements of the platform, a model was built to semantically describe the platform socio-collaborative activities.

The first important thing to do in order to model the platform structure to ontologies is to focus on the very essential of Drupal's pages: a **node**. Everything from a user profile to a content item is a node.

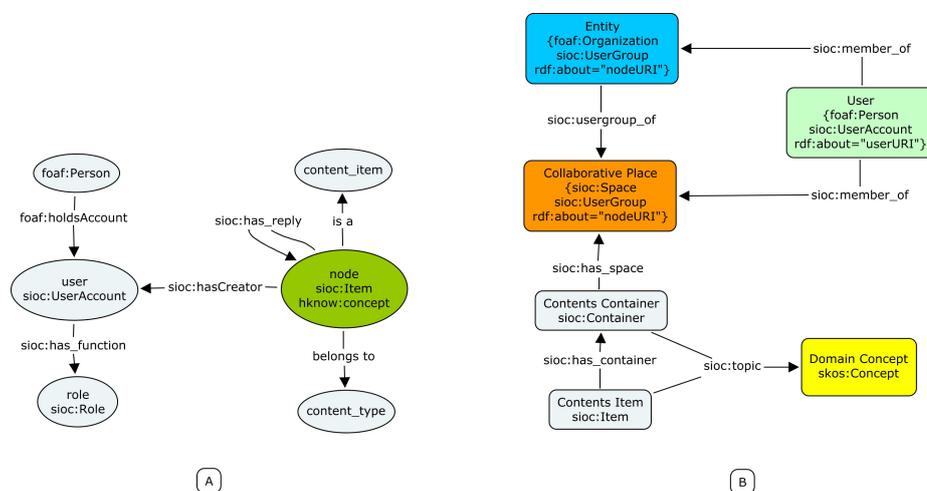


Figure 3. A) Drupal node structure mapped into ontologies (based on [2]) B) Integration of SIOC + FOAF + SKOS into the H-Know platform

In the schema 3 A), every Drupal node is considered a **sioc:Item**. A `sioc:Item` is a class that describes something that can be in a container [6]. It has subclasses that can specify different types of Items such as `sioc:Post` to describe a Forum post. With the SIOC Types Module, we can create new types of `sioc:Item` to describe other types of content.

Platform users are represented with the class **sioc:UserAccount**. A user, apart from the platform, is classified as a `foaf:Person`, with his own set of characteristics and interests, independent from the platform. A User is connected to

the Person he represents by the property **foaf:holdsAccount**. To connect the node with the user that created it, we use the property **sioc:has_creator**.

Coming one level upper in abstraction, we built another model (3 B)) to describe the specific elements of the platform and the relations between them. From it we can discuss many of the options took to classify the platform. In this diagram we specify for the main elements, their `rdf:about` property, to define the subject of the triple statements.

First of all, it was decided to define a ColPlace with two different SIOC classes: **sioc:Space** and **sioc:UserGroup**, using as resource identifier the URI of the node that is the index page of that ColPlace. This approach was followed because of the multiple behaviour of a ColPlace. It is both a place to aggregate information and platform agents (users and entities). So, since a `sioc:Space` "is defined as being a place where data resides" [6], we use it as the location for a set of containers. Any data that resides in a `sioc:Space`, can be linked to it using the property **sioc:has_space**.

In a ColPlace, we have different types of content containers with their content items associated. The content containers of a ColPlace are classified using SIOC and SIOC Types. Each content Item is linked to its container using the property **sioc:has_container**.

An entity, in its individual identity, is defined as a **foaf:Organization**, "a kind of Agent corresponding to social institutions such as companies, societies". In the context of the H-Know platform, an Entity can also be seen as a `sioc:Usergroup`, "a set of UserAccounts whose owners have a common purpose or interest" [6].

To classify the group behaviour of a ColPlace the **sioc:UserGroup** structure is used. To link the H-Know users to the ColPlaces they are part of, we use the **sioc:member_of** property. The association entities have with a ColPlace is expressed by the property **sioc:usergroup_of**.

To describe the relationships between platform users, we use the RELATIONSHIP ontology⁷, which extends the **foaf:knows** core property, providing extra types of relationships between users such as: **Employed_By**, **Employer_Of** or **Works_With**, enriching the description of the platform interactions.

To connect platform content items with the domain knowledge, it's used as a bridge the property **sioc:topic** to a **skos:Concept**. This property can be applied to most of the classes defined in the SIOC ontology. So, we can for example assign a set of concepts to a container and then propagate those topics to its items.

Each content item may have a set of Domain Concepts related to it. So, we will have triples where the **subject** is the URI of a content item, classified with the SIOC vocabulary, the **predicate** is the `sioc:topic` property, making the bridge between a content item and a concept and the **object** is a `skos:Concept` class representing a concept in the domain knowledge. This way we can specify the concepts which relate to each content item of the platform.

⁷ <http://vocab.org/relationship/.html>

2.4 Conceptualizing a semantic search interface for the platform

After having the information produced in the H-Know platform semantically classified, users should be able to make searches taking advantage of that classification.

One possible approach is to build facet-browsing search interfaces, where users can, by the means of filters, continuously redefine their searching criteria. Starting from a non-filtered set of web references, users can use pre-defined filters to decrease the set of results, getting just what they were trying to find.

The idea is, using all the metadata generated in the platform, to get results relating domain knowledge concepts with the socio-collaborative activities.

We define 2 sets of facets filters: one for the socio-collaborative characteristics of the content item, such as the author of a content, the type of content or the type of collaboration place where it is produced and another set of filters related to the topics defined in the domain ontology.

This way we can get an answer for a question like: "Which blog-entries Manuel created about churches?", filtering creator for "Manuel", type for "blog-entry" and Construction Result Space for "Church".

3 A Technological Architecture for the platform semantics integration

From the conceptualizations presented before in this paper, the diagram 4 describes the technological architecture implemented in the H-Know platform.

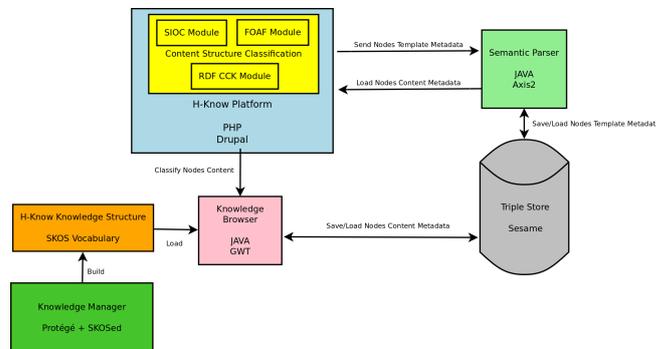


Figure4. H-Know platform Semantic module architecture

So, starting from the bottom layer of the architecture, to store the metadata triples, **Sesame** framework was chosen. Sesame was chosen as the triple store because it provides a fast and reliable native triples store, with a comprehensive

back-end managing interface (Workbench) and a very complete and easy to use Java API with functions to interact with it.

Changing the focus to the Drupal platform, for the node's template semantic classification, a conjunction of three different customized Drupal modules (**RDF CCK Module**, **FOAF Module** and **SIOC Module**) was the solution. RDF CCK is an "out-of-the-box" module, that provides an extension to the content types manager (CCK), to map each content type (node template) and its fields to ontology vocabularies. This module, exports the semantic metadata of each node template in different ways, such as RDFa or to a file with the semantic information of a node. In addition to this module the FOAF and SIOC modules were used, with some modifications, to describe the platform socio-collaborative characteristics which are not expressed by RDF CCK. These modules act the same way as RDF CCK, exporting FOAF and SIOC information of the nodes to RDF/XML files.

To establish a connection between Drupal and Sesame, since Drupal is built over Php language and Sesame API is Java, a "**Semantic Parser**" was built to work as an intermediate between Drupal and Sesame, using a Web Service. This application is used both to save Drupal metadata in Sesame and to load metadata from Sesame into Drupal. This application then uses the API of Sesame to perform the required actions (saving or loading of metadata).

The "**Knowledge Domain**" is managed using **Protégé** with a plugin for the construction of SKOS vocabularies (**SKOSed**).

To implement the semantic classification of the nodes content, it was developed from scratch a Java GWT application, "**Knowledge Browser**", to load the domain knowledge structure of the platform and give users the chance of choosing concepts which relate with the content they are producing. This browser loads the knowledge structure using a Java API called **SKOS API** and interacts with the Sesame triple store using its API.

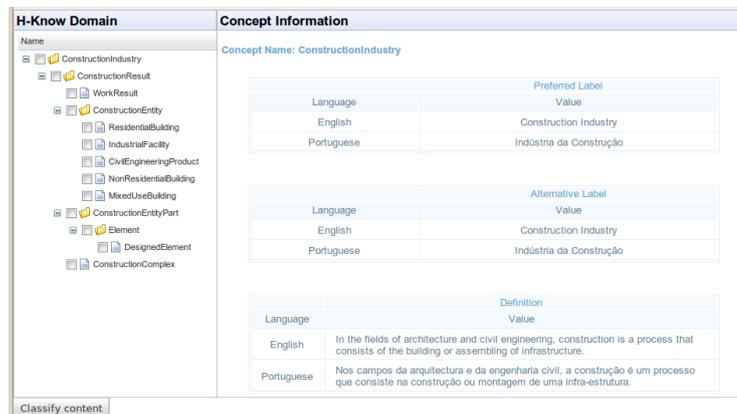


Figure5. Ontology Browser interface

Users select the concepts and then click the “Classify content” button which presents a report of the selected concepts performs the classification. The browser is integrated inside Drupal. In every node edition, a user can use it to classify the content he is publishing.

4 Related Work

Research on semantically enhanced systems is growing although only few of the projects focus on CMSs. Good examples of that are [10] or [11] which describe architectures to integrate semantics in CMSs from scratch. In [9] is presented a generic architecture for the integration of semantic annotation and usage in a CMS. We have a different approach since we use a already available CMS (Drupal).

The closest studies to our work are presented on [2]. We use a module presented on this paper (RDF CCK) which automatically generates semantics from the ontologies defined in CCK (Content Construction Kit). In this paper, they suggest a solution which stores the metadata on the Drupal installation database (a non native triple store), providing a SPARQL endpoint for the generated metadata. Our approach is different since we store the metadata outside Drupal in a native triple store (Sesame).

There was no “out-of-the-box” solution to classify content items according to a domain ontology, so it was developed from scratch, a browser for it which loads SKOS vocabularies. That browser was integrated in the Drupal framework.

In the other hand, making use of the technological architecture suggested, this paper presents a ontology model using existing W3C ontologies: FOAF, SIOC and SKOS. Drupal is structured and organized in a way where the integration of SIOC and FOAF to describe the socio-collaborative activities can be naturally integrated. In [5] is presented a model to analyse the social relations between users through the content that they create and an exporter for that model. Another project in the area of socio-semantics is Flink [13], a system for the extraction, aggregation and visualization of online social networks. In this paper we present a model which intends to semantically describe the social network of the H-Know platform, the collaborations held there and the connection of these with Domain Knowledge, described using SKOS data model. One of the most interesting SKOS projects is PoolParty [14] a web application to create and maintain thesauri with a easy to use user-interface. The solution presented aims to integrate both socio-collaborative and knowledge semantics, a kind of approach we haven’t seen so far in other research projects.

5 Conclusions and further work

It was designed a model to express the socio-collaborative activities managed in the platform, integrated with the Domain Knowledge of the H-Know project, using existing standard W3C ontologies. In the other hand, a technological architecture to implement the model was presented.

All the generated metadata is stored outside of the platform in a native triple store application, Sesame. In a sum, we have the content created in the H-Know platform semantically described, ready to be consumed by any application that wants to take advantage of the information available in the H-Know platform.

The next step will be developing the semantic search interfaces that were designed. Since we used standard vocabularies, we can use the metadata generated in existing projects that consume the same type of metadata.

References

1. U. Bojars and J. G. Breslin and A. Finn and S. Decker: Using the Semantic Web for linking and reusing data across Web 2.0 communities. Web Semantics: Science, Services and Agents on the World Wide Web 2008
2. S. Corlosquet and R. Delbru and T. Clark and A. Polleres and S. Decker and A. Haller and M. Marmolowski and W. Gaaloul and E. Oren and B. Sapkota and others: Produce and Consume Linked Data with Drupal! International Semantic Web Conference 2009
3. Stéphane Corlosquet: Drupal RDF Schema proposal | groups.drupal.org <http://groups.drupal.org/node/9311>. visited on April of 2010
4. Antoine Isaac and Ed Summers: SKOS simple knowledge organization system primer. <http://www.w3.org/TR/skos-primer/> visited on March of 2010
5. Uldis Bojars, Benjamin Heitmann and Eyal Oren: A Prototype to Explore Content and Context on Social Community Sites. SABRE Conference on Social Semantic Web (CSSW) 2007
6. Uldis Bojars and John Breslin: SIOC core ontology specification. <http://rdfs.org/sioc/spec/> visited on May of 2010
7. Dan Brickley and Libby Miller: FOAF Vocabulary Specification. <http://xmlns.com/foaf/spec/> visited on May of 2010
8. ISO 12006-2 Building construction — Organization of information about construction works - Framework for classification of information DIS Version 2001
9. G. B. Laleci and G. Aluc and A. Dogac and A. Sinaci and O. Kilic and F. Tuncer: A Semantic Backend for Content Management Systems. Knowledge-Based Systems 2010
10. Garcia, R., Gimeno, J.M., Perdrix, F., Gil, R., Oliva, M.: The Rhizomer Semantic Content Management System. Emerging Technologies and Information Systems for the Knowledge Society, WSKS 2008, Athens, Greece, September 24-26, 2008
11. Minh Le, D., Lau, L.: An Open Architecture for Ontology-Enabled Content Management Systems: A Case Study in Managing Learning Objects. On the Move to Meaningful Internet Systems 2006, France, October 29 - November 3, 2006
12. Aumueller, D., Rahm, E.: Caravela: Semantic Content Management with Automatic Information Integration and Categorization. ESWC 2007
13. Peter Mika: Flink: Semantic Web technology for the extraction and analysis of social networks. Journal of Web Semantics 2005
14. Thomas Schandl and Andreas Blumauer: PoolParty: SKOS Thesaurus Management utilizing Linked Data. The Semantic Web: Research and Applications 2010