SCOP: a Java Framework for Building Semantic Virtual Communities in the Web

Marcio Balian and Cedric Luiz de Carvalho*

Federal University of Goiás
Institute of Informatics
P.O. Box 131 CEP 74001-970
Goiânia, Goiás, Brazil
marciobalian@gmail.com,cedric@inf.ufg.br

Abstract. This paper describes the development of a framework to build Virtual Communities of Practice in the web. It is part of a larger project named as DWeb (acronym for Dream Web). The project DWeb aims to contribute to the development of the Semantic Web by offering an environment to set such of communities in the web. It is guided by the fundamental principles of the Semantic Web defined by W3C. The framework can be classified as a tool to build virtual communities environments that support Semantic Web resources. It can be used to construct collaborative systems based on communities and in knowledge management through collaborative environments.

Key words: Semantic Web, Oriented Object Frameworks, Virtual Communities of Practice, Ontologies

1 Introduction

The web is a very huge repository of data. Retrieving information from this repository is a difficult problem to solve. There are many groups of researchers working on it. One of the main causes of this problem is that the web was projected to human beings, that is, the webpages are constructed to be seen by people. In addition, their content are represented using natural languages[1]. So, it is difficult to construct automatic mechanisms, carried out by machines, to retrieve information from the web.

There are some search mechanisms that produces very good results (as *google* [1], for instance). But, they perform only syntactic searches, based on keywords. The lack of using semantics in search mechanisms produces results that do not satisfy users needs. The Semantic Web [2] is a proposal to associate semantics to every resource in the web. The use of web tools, enriched with semantics can produce much better results.

* Corresponding author.

1 English, Portuguese, Spanish etc

To handle natural languages is a difficult task. There are many problems such as to eliminate ambiguities, delimiting the context, etc. One way to make the searches more efficient and effective is to delimit the scope of them. A context can be well defined inside a community environment. Of course, members of a community can be interested in subjects out of the context of their community, but most of the time it is expected that they are concerned about the main subjects of their community.

The web provides an environment where many kinds of virtual communities can be established. For instance, there are communities of interests, communities of relationship, communities of entertainment, communities of practice, etc. The Communities of Practice (CoPs) are characterized by a domain, the practices and the community itself.

A CoP is a community of people with common interests, with a well-defined knowledge domain (context). It focus in the practice and quotidian jobs. It can be seen as an identity of a group of people. Such groups can meet each other to provide a way to exchange knowledge and practices that can be collected and stored for future reference and retrieval.

The CoPs are highlighted by the possibility of extending their dimensions (domain, community and practice) to implement applications for knowledge management through collaborative environments.

The Semantic Web tools are constructed according to the Semantic Web macro requirements: intelligence[2], interoperability[3] and integration[4]. The use of these tools can produce very interesting environments for knowledge management as well as for sharing practices and knowledge.

The authors have been working on a Semantic CoPs Framework (SCOP). The purpose of this framework is to provide developers with a core of basic resources to create Virtual Communities of Practice. It is written in Java and it is based on the Semantic Web principles [5].

In the following, the most important aspects of the construction of SCOP will be discussed. Initially, in Section 2, the main aspects of CoPs are discussed. In Section 3, some related works are presented. In Section 4, the methodology used in SCOP's development is described. In Section 5, the architecture of SCOP is presented. This Section also shows the Semantic Web techniques used in SCOP development. Finally, in Section 6, the final considerations are done.

2 Virtual Communities of Practice

A CoP has three fundamental aspects, also called **dimensions**, that defines it: the knowledge domain, the community itself and the practice.

[2] Implemented through reasoning mechanisms.

[3] Allows information sharing. It can be of three levels: syntactic interoperability (provided by XML), structural interoperability (provided by RDF [3]) and semantic interoperability (provided by ontologies and other resources).

[4] Allows clients to exchange information in a transparent way. It can be implemented through mediation/translation mechanisms [4].

The knowledge domain is the context of the community. It characterizes their member's shared interest. Each member should have a minimal level of knowledge about the domain [6]. This characteristic is essential to relate CoPs to the Semantic Web. This relationship will be discussed later in this text and it is fundamental to SCOP. A shared context is essential to the development of a community because it gives sense and orientation to talks between community members. It also can help people to decide in what direction to guide the community. The context can include purpose, content, history and values. Making explicit the knowledge shared domain can be decisive to a community success or fail.

Another dimension of a CoP is the community, that is, the way members kept engaged in conjunct activities, discussions, mutual help and information sharing. The interaction is a key requirement for members to belong to a community. Thus, the way a CoP works comes from the way people became member of it, that is, from practice sharing. Members of CoPs are tied together by means of a mutual commitment in common activities. It is this mutual binding that connects members of a CoP with each other as a social entity [6].

The production of CoPs' members comes in another dimension of their definition. It is centered in the shared repertory that represents the material profile of the community. As examples it can be cited: written files, procedures, documents, politics, rituals and specific languages.

The common goal is the key motivation. The members of communities have the same set of goals. In the case of communities focused in practice, such set of goals are centered in knowledge sharing between a group of practitioners. For this reason, effective CoPs are structured mainly around knowledge sharing activities. The key point that distinguish CoPs from other kind of communities is their main objective: the knowledge sharing between a group of practitioners (through activities such as brainstorm, relationship and exchange of reading material, such as papers, news, reports, for instance).

CoPs as Knowledge Management Support Tool

Knowledge management is a systematized process wherein members of an organization can detect, select, organize, filter, present and use information. The objective of such process is the collaborative exploration of resources of the organization [7].

From those stated characteristics one can note that a well-structured CoP can be seen as a collaborative environment. Such environment offers support to systems applied to knowledge management, mainly if it is used with the Semantic Web resources.

A resource can be considered as a source of information published in an environment, accessible to the Internet users or to the members of a community. A resource is identified by an URI (*Uniform Resource Identifier*). It can be a HTML web page, a person registered in a database, a published PDF document, etc.

The concern of identifying, creating, and managing knowledge is part of an actual and fundamental reality for any organization. Information Technology

(IT) tools, such as those that can be developed using SCOP, and the collaborative systems that give support to knowledge management, contribute to make this process much more efficient.

3 Related works

One of the main characteristics of a CoP is the sharing of resources, that is, the artifacts produced by its members. In a community, built without semantic concerns, the search for resources are done based only in syntactic mechanisms. These mechanisms are very limited. They can not, for example, distinguish the retrieved resources according to the user's context.

The amount of resources in the web have been growing in a exponential and disorganized way. The Semantic Web is an attempt to overcome the information organization problem of the web, improving the resource retrieval process. These kind of problems are the focus of the project described here. In the following, some works related to this project are showed. They provide some support to virtual communities, mainly by improving resources interoperability and adding intelligent functionalities to information retrieval resources.

Etienne Wenger [8], in 2001, studied exhaustively the tools and technologies to support total or partially the Virtual Communities of Practice. In his study, market tools were analyzed according to groups of characteristics of CoPs. From there, many other researchers have used the same criteria to analyze such kind of tools. Those criteria have been extended to consider the requirements of the Semantic Web in the evaluation of virtual environments.

One of the most used tools in virtual communities is the **wikis**. A wiki is a tool for collaborative creation and knowledge sharing. The members of a community can interact with each other through a hypertext environment. A wiki page can be created or edited by any member. Information can be reorganized easily, new pages created or old pages renamed. There are many wikis in the web. Ward Cunningham was one of the first to create a wiki. The purpose of Cunningham's wiki was to provide a collaborative environment for people interested in discussions about Design Patterns [9]. Nowadays, one of the most successfull wiki initiatives is the **Wikipedia** [10]. Such tools are based on collaborative works and can satisfies many requirements of CoPs.

Like the wikis cited before, the tools listed in Wenger's work satisfies many CoPs requirements, but without using Semantic Web resources in their implementation. Works like *Semantic Media Wiki*[11], *Platypus Wiki* [12] and *OntoShare* [13], are specially related to the framework described here since they use Semantic Web techniques. In a conventional wiki, the concepts and pages (information resources) are traversed by using regular links. In a semantic wiki, proposed by Semantic Media Wiki (SMW)[11] and *Platyplus Wiki* [12], for instance, the wiki pages can be seen as RDF resources, that is, resource-property-value statements. Using semantics, one can form structures conceptually more well defined.

Davies, Duke and Sure describe **OntoShare** [13], a knowledge management environment for Virtual Communities of Practice that aims to support communities and make them effectives. The OntoShare environment can catalog and extract keywords from sources of information shared by an user[5]. It also can make such information available to the other users of a CoP, whose profiles indicates that they may have interest in those information.

Making available information to its members, even in an organized way, do not guarantee the success of a community. It is also necessary to define the semantic connections between each stored information. A tool for knowledge management can do much more than to help in information retrieval. It can allows reasoning, abstraction and can help users to share new information, stimulating the community in a consistent way.

When users add new information to share, a new knowledge resource is created and extended with annotations. Ontologies are defined using RDF Schema (RDFS). Each user has a profile in the environment. The profiles are represented as a set of topics or ontological concepts representing the interests of each member. The profile is dynamical and is frequently adapted by the environment according to user's actions to reflect his real interests.

We propose a framework (SCOP) that can help in providing semantic resource interoperability besides members and CoPs management. It will be discussed in the following sections.

4 The Framework SCOP

A framework is a structure of interrelated classes that forms a kind of skeleton, i.e., a basic structure to help in construction of applications in a specific particular domain. The use of such structure minimizes the effort of developing applications since the developer can abstract concerns about the system architecture [14].

To design a framework, it is necessary to project a structure of classes according to the requirements of the application domain. This project must allow adaptation to the peculiarities of any new application to be developed. Thus, to develop a framework is a complex task.

Although there is not an ideal methodology to be used in framework construction, some methodologies were evaluated in order to define the one used in SCOP construction.

4.1 Methodology Used

The SCOP was constructed following the steps described bellow. They were selected combining two methodologies: Example Driven Design [15] and Hot Spots Driven Design [16].

1. Detailed study about the application domain;

[5] In this text, *user* means member of a community.

2. Project of class structure;
3. Implementation of class structure;
4. Evaluation of performance, by testing application constructed using the framework;
5. Other tests and improvements.

In the following subsections, the above steps are detailed.

4.2 Analysis of the Application Domain

In step 1, tools constructed according to the Semantic Web principles that support communities were analyzed. Then, the main requirements and individual characteristics of each single example application evaluated were identified. After that, the common requirements and characteristics of each single application analyzed were identified, as well as the possible applications to be constructed in the domain. From this, it was defined that SCOP should offer to the developer:

- Objects representing people in the environment;
- Objects representing CoPs in the environment;
- Services to control people participation in CoPs;
- Services to control participation levels and respectives privileges to use the CoPs' environment and its tools;
- Persistence to objects representing CoPs and its members, since these objects should must be retrieved at any time by final users of the developed environments;
- Services to create and edit CoP resources, using tools projected by the users;
- Persistence service for the resources handled through CoPs' tools, since they must form a resource base always available to the members;
- Services for association between CoPs and tools they use, including persistence of such associations;
- Services for maintenance[6] of RDF properties in CoPs, including persistence of such properties;
- Services to interrelate resources in CoPs, or to relate them to values (literals), through properties;
- Persistence for the relations of the item above;
- Services for maintenance of namespaces in the environment, including persistence of such namespaces;
- Services for resources and properties organization through namespaces in the environment;
- Services to import and search concepts in OWL ontologies.

[6] Maintain, in this text, means create, update or delete.

4.3 Project and Implementation of the Framework

After analyzing the domain of the application and identifying its requirements, the structure of classes was projected and implemented. It was used the Java language [17] and the framework Jena [18] in implementation.

Jena is a framework, developed in Java, to construct applications to the Semantic Web. It was developed by the group *HP Labs Semantic Web Programme* under "open source code" license. Its architecture is centered in a RDF [3] database. The framework has classes to represent and manipulate graphs, resources, properties and literals. A graph is represented by the interface `Model`.

The interface `OntModel` [19] extends the interface `Model`, allowing, besides the RDF graph operations, structures and operations on ontologies, as classes, subclasses, instances and function properties, for instance.

SCOP uses a persistent ontological model (`OntModel`). The services offered to controll communities, members, tools and resources reflect directly in the ontological model.

4.4 Evaluation and Testing

To test the framework, an application to manage Virtual Communities of Practice was developed. Each community created using the application includes a wiki tool. Members of these communities can interact with each other using wiki pages.

With this application it was possible to maintain members and to manage their participation in communities. Members of the communities can create and edit wiki pages.

SCOP handles each wiki page as an instance of the class **CopResource** (see Section 5). So, each wiki page is a Semantic Web resource and it have attributes and services. Thus, community members can interconnect resources (in wiki page form). As result, it is generated a dynamical resource database, interconnected as a RDF graph.

The application mentioned above could be easily developed after reading SCOP's documentation and examining its source code. The developers of such application found no difficulties in choosing which classes to use, which ones to extend, which methods to implement, and so on.

SCOP is proposed to be a framework that allows to work with the three dimensions of a CoP, using the Semantic Web resource sharing technics, as can be seen in the following subsection.

5 SCOP's Architecture and Functioning

SCOP offers a basic and generic structure to help developers of tools. They can propose new tools. They also can propose new resources and use the structure offered by the framework to interrelate them, independently of the tool used to generate them or independently of the resource architecture.

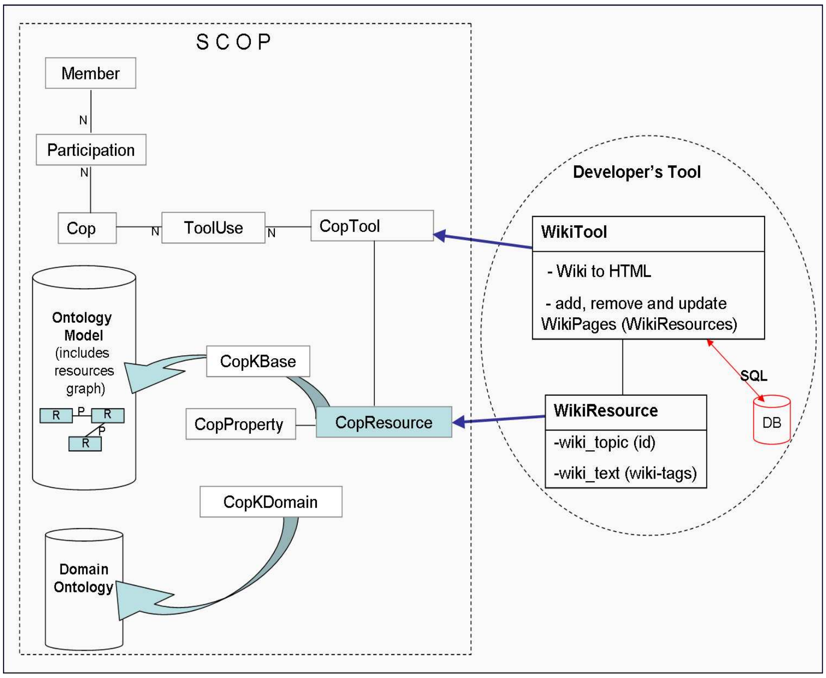

Fig. 1. Example of using the proposed architecture: a wiki tool.

Each CoP has a resource base that is structured as a RDF graph. The resources in this base can be constructed by using different tools. It can be used the tools provided by the framework itself or the tools created by developers. This allows improvements in the interoperability and intelligence levels of the environments. A chat, a forum, a wiki, or a tool for presenting virtual lectures, can be examples of tools that can be constructed by developers. SCOP provides high level services to link the URIs of such tools in order to construct a RDF graph. This graph forms a RDF base containing the resources produced using the tools projected by developers.

There is not a rigid architecture for the information resources maintained using the proposed framework. The developer can model the tools and its resources according to his necessities. For instance, a wiki tool can be projected in such a way to maintain its wiki pages. The wiki pages will be considered as resources, which will have its own representation and storage forms, independently of SCOP.

The Figure 1 shows an example of application of the proposed architecture. It can be seen, in the circular area at the right, a wiki tool, projected by a developer (class `WikiTool`). A wiki maintain its resources, represented by the class `WikiResource`. These two classes extend, respectively, the classes CopTool and CopResource and can use all the structure of SCOP.

The relationship between resources are established by RDF statements, using the available properties (instances of `CopProperty`). The relationships are persisted in the ontological model of the environment, forming a RDF graph of resources. The class `CopKBase` is responsible to form, maintain an retrieve resources and its respective relations.

A wiki resource (wiki page) is defined with two attributes, an identifier of the wiki page (wiki topic) and its text (wiki text), codified in the conventional sintax of the wiki engines. The wiki tool stores its resources in its own database. The persistence operation on these resources are implemented in `WikiTool`. The developer is free to implement his tool, his resources and the way to make them persistent, using a database or a file system.

The classes `Cop`, `Member` and `Participation` represent communities, members and participation control service, respectively. The class `ToolUse` and the abstract class `CopTool` offer services to control the use of the tools developed for the CoPs. The developer must implement, in his tool, the methods to maintain the resources he want to use.

The class `CoPKDomain` has services to load from the ontology that defines a domain (an ontological model), codified in OWL [20]. The ontology represents the relationship between the concepts of the domain of the community. The developer can use the class `CoPKDomain` to load an ontology that expresses the knowledge domain of each CoP. When a member requests for information retrieval, the ontology can be traversed using the methods of this class. The result can be compared with the knowledge base of the CoP. Thus, the developer can decide if he want to show the results according to the concepts in the ontology or if he want to modify the knowledge base of the CoP following the ontological structure.

The ontology loaded represents the domain of the CoP. Such ontology is used in insertion and retrieval of resources in the CoP. In insertion, the ontology can be consulted or just used as a guide to insert resources. These resources should correspond to one or more classes of the ontological structure. In resource retrieval, the ontology can be traversed in such a way that each member can get more expressive results, even if the resources are stored without consulting the loaded ontology (imported).

In SCOP, an ontology and the RDF knowledge base are different things. The framework provides methods to traverse the ontoloy. The developer of the system is who defines the use of its concept (classes) structure. He can use it, for instance, in insertion of resources of a CoP, or as a guide to put the resource in the knowledge base (RDF graph) in a proper way.

The developer can also allows insertion of resources without consulting the ontology. He can create, for instance, an interface that suggests to the community member just resources, properties and values found in the knowledge base of the CoP (a RDF graph). For instance, a member of a CoP about **animals** would want to insert a resource about **cammel**. The developer can have loaded (imported) an ontology about animals and can traverse it to find the concept cammel. Then, he can get all child concepts of cammel, as well as its parents. He also can get the properties that link cammel with its children and parents.

With these information, he can, for example, search (now in the RDF graph of the CoP) for the term cammel. If this resource does not exist in the RDF data base of the CoP, the developer can insert it in the graph, including its children, parents and properties found in the ontology. When he locate the class cammel in the ontology he found that this class is a subclass of **mammal**. This information can be included in the RDF graph of the CoP using, for instance, the property **isA** to link it to the resource mammal of the CoP.

Suppose a case, for instance, where a developer had used SCOP to build a CoP based in a wiki web and where members insert and retrieve resources through wiki pages. Each wiki is considered as a resource. When a page is inserted, the system can show an interface suggesting to the member to classify the resource as an instance of a class retrieved from the ontology. The system can also suggests to link the page to another resource retrieved from the knowledge base (a RDF graph of the CoP) through properties also retrieved from the knowledge base of the CoP. In case there is a relevant property, the developer can foresee that the member will create new properties expanding the graph RDF that interrelates the CoP resources.

If the developer opted to not guide the user in inserting resources according to the ontology, he can, in the search for resources (retrieving process), compare them with the ontology, giving more intelligent answers according to the community expected context.

6 Final Considerations

The Semantic Web aims to overcome the lack of semantic (meaning) of information available in the web. The communities of practice facilitates the communication between people interested in a specific context, related to their diary practices.

The combination of Semantic Web objectives and CoPs objectives suggested the construction of a tool to help Java developers to build environments based on more transparent and intelligent sharing resources. Besides that, it can stimulate the creation and growing of Virtual Communities of Practice in the web.

SCOP tries to explore an application domain little explored and provides the developer with a core of basic functionalities to implement a virtual environment for Communities of Practice (CoPs).

The CoPs can give support to collaborative work environments. Since members relate with each other in a specific knowledge domain, the Semantic Web structures, added with the SCOP's core of functionalities allow the developers to go beyond of CoPs basic features, permitting the construction of Knowledge Management systems, based on collaborative work.

The resources (artifacts) produced in CoPs, generated with tools, such as the forum, wiki, file sharer, and others, can be implemented. These resources can be linked with each other or related to an ontology that expresses the context of a

CoP. This guarantee flexibility because many tools can be created to facilitate the relationship between members of communities. These tools can also be included in the framework itself, if its requirements are satisfied.

Thus, SCOP can be saw as an initiative to build an extensible framework to establish Virtual Communities of Practice in the web. New tools can be easily added to the basic core provided by the framework, according to the necessities of each new community. One fundamental aspect of it is the application of the Semantic Web macro-requirements: intelligence, interoperability and integration.

References

1. Team, G.: Google. http://www.google.com, viewed in September 2008 (2008)
2. Berners-Lee, T.: Semantic Web. Scientific American **May 2001 issue** (2001) 232–234
3. W3C: Resource Description Framework (RDF). http://www.w3.org/RDF/, viewed in July 2007 (2005)
4. Widerhold, G., Genesereth, M.R.: The Basis for Mediation. In: Conference on Cooperative Information Systems. (1995) 140–157
5. Consortium, W.: W3C Semantic Web Activity. http://www.w3.org/2001/12/semweb-fin/w3csw, viewed in August 2008 (2008)
6. Wenger, E.: Communities of Practice: Learning, Meaning and Identity. Cambridge University Press (1998)
7. Davenport, T.H., Pruzac, L.: Working Knowledge: How Organizations Manage What They Know. Boston: Harvard Business School Press (1998)
8. Wenger, E.: Supporting Communities of Practice - A Survey of Community-oriented Technologies. Report to the Council of CIOs of the US Federal Government. http://www.ewenger.com/pub/index.htm, viewed in September 2008 (2001)
9. Cunningham, W.: Cunningham and Cunningham, Inc. http://www.c2.com/, viewed in August 2007
10. : Wikipedia. http://www.wikipedia.org/, viewed in August 2007
11. WikiMedia: Semantic MediaWiki Home Page. http://meta.wikimedia.org/wiki/Semantic_MediaWiki, viewed in August 2007 (2007)
12. Campanini, S., Castagna, P., Tazzoli, R.: Platypus Wiki - The Semantic Wiki Wiki Web. http://platypuswiki.sourceforge.net/, viewed in June 2007 (2005)
13. Davies, J., Duke, A., Sure, Y.: OntoShare - An Ontology-based Knowledge Sharing System for Virtual Communities of Practice. ACM - Association for Computing Machinery (2003)
14. Taligent: Leveraging Object-Oriented Frameworks. White paper, Taligent Inc. (1995)
15. Johnson, R.E.: How to Design Frameworks. In: Tutorial Notes for the 1993 Conference on Object Oriented Programming, Systems, Languages and Systems (OOPSLA '93). (1993)
16. Pree, W.: Design Patterns for Object-Oriented Software Development. Addison Wesley Longman, 1st edition (1994)
17. http://java.sun.com/, viewed in April 2007
18. HPLabs: Jena: A Semantic Web Framework for Java. http://jena.sourceforge.net, viewed in May 2007 (2006)

19. HPLabs: Jena 2 Ontology API. http://jena.sourceforge.net/ontology/index.html, viewed in May 2007 (2003)
20. W3C: OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/, viewed in August 2007 (2005)