An ontology-based content management system for a dynamic operating context: issues and prototype evaluation

Maurizio Agelli, Felice Colucci
CRS4, Parco Scientifico e Tecnologico della Sardegna, Pula (Cagliari), Italy
agelli@crs4.it
felice@crs4.it

Abstract: Content management systems are very useful tools for organizing and sharing information resources and may considerably benefit from using ontology-based description schemes. Ontologies set a common ground for resource acquisition, enabling different users to share a common view of a knowledge domain, and may considerably enhance the search paradigms by exploiting semantic relationships between concepts. However, ontologies may evolve since they reflect knowledge schemes that are by nature dynamic. Moreover this evolution should be the result of a collaborative process of ontology maintenance. These and other issues are addressed in the present work and some practical solutions are proposed. Also, a very simple prototype implementation of an ontology-based content management system is described. Finally, the results of a short experimentation of this prototype within a small community are presented.

Keywords: content management systems, ontologies, knowledge sharing, ontology evolution, communities of practice

1. Introduction

Nowadays, the availability of reliable information sources is a key factor for any decision making process in private and public organizations. However, the amount of information sources that are available on-line and off-line is overwhelming and this abundance of content may become virtually useless without the right tools that allow retrieving the useful information items. Search engines, although restricted to on-line resources, play an important role, but the results they provide are quite often imprecise and narrowing down the search may be a very complex and time-consuming task.

Content management systems (CMS) provide an integrated framework, based on metadata and workflows, for organizing and sharing information resources within a community of users. Many description schemes have been standardized, defining metadata sets and vocabularies and providing a good level of interoperability among applications. However, metadata vocabularies are generally too loosely related to the operating context of the end users and finding the most suitable tags for describing a resource can lead to inaccurate or ambiguous results. Information should be contextualized using a description that makes sense in the domain where the information will be used. What distinguishes knowledge management systems from content management systems is that they deal with information plus semantics, not with information alone (Riedl 2002).

Ontologies, adding the required semantics to vocabularies of concepts, may considerably improve the classification and search paradigms. It is much easier to identify the right description tags exploiting semantic relationships between ontology concepts, rather than looking up in a long list of keywords provided by a metadata vocabulary. Also, ontologies may integrate different views of a knowledge domain, enabling different users to reach the same concept going through different semantic paths. An ontology, intended as a set of structured terms that describe some domain, provides the skeletal structure for building a knowledge base (Swartout et al. 1996).

The centralized technological infrastructure obtained by combining a CMS with an ontology is still weak from an organizational perspective because it has to face a distributed social form made by communities that operate in different and dynamically changing contexts. Knowledge should be autonomously managed where it is created and used. Each community should formalize its own context and then create a mapping with the contexts of the other communities through social interaction (Maier 2002). The process shall lead to the definition of a common ontology that is understood and accepted across the different contexts and can be used to classify the knowledge repository.

2. Framework overview

A methodological and technological framework was set up with the aim to encourage the sharing of information resources within research teams working in the domain of digital media systems and applications. The objective was to provide an online toolset that could help to keep track of information sources that were considered worthwhile for a subsequent reuse.

A CMS provided the basic infrastructure for collecting, annotating and retrieving information resources. A set of ontology tools provided the support for ontology construction, resource classification and semantic search.

The ontology described a very dynamic knowledge domain that, like many other information technology topics, was far from being consolidated. A collaborative environment was provided for building and maintaining ontologies through an evolutionary process that involved the participation of all teams.

3. Methodology

3.1 Approach in terms of knowledge management strategies

Knowledge management practices are based on two fundamental strategies: the codification strategy, consisting of making explicit the knowledge that is tacitly held by people, codifying it in documents and making it available for a subsequent reuse, and the personalization strategy, involving the direct transfer of tacit knowledge among people through socialization and personal interaction (Hansen et al. 1999).

The knowledge management approach based on “communities of practices” represents a recent evolution of the personalization strategy and is based on the consideration that knowledge cannot be separated from the communities that create it, use it and transform it. In any type of knowledge work people are required to personally interact and exchange experiences with other people, even where technology provides a considerable support (Allee 2000).

Communities of practice contribute to creating a common language and context that can be shared by community members. They also contribute to developing taxonomies within the common repositories managed by community members where individuals could submit knowledge artefacts that could be reused by others (Lesser et al. 2001).

The proposed framework is based on a mix of the two knowledge management strategies. The people-to-document approach of the codification strategy is adopted in the CMS, that acts as a common knowledge repository. The people-to-people approach of the codification strategy is adopted by setting up a collaborative process, based on communities of practice, aimed at defining an ontology matching the different operating contexts.

In figure 1 is described a case where two teams, working in different operating contexts, interact in order to define a common ontology that is used to contextualize the repository of information resources.

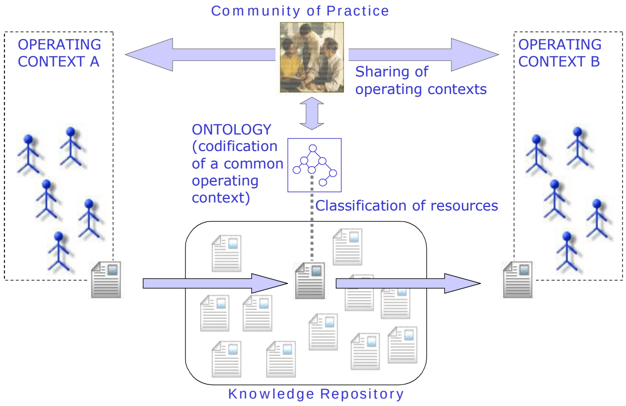

Figure 1: interacting with the knowledge repository from different operating contexts

3.2 The metadata schema

The choice of a suitable metadata schema is a fundamental step towards the specification of a knowledge repository. Many metadata schemas exist, addressing specific application requirements, however the Dublin Core (DC) metadata set, defined by the Dublin Core Metadata Initiative (DCMI) has been chosen for the present work, for several reasons:

- it is an open standard, adopted by W3C, and provides full interoperability with other applications;
- its metadata attributes can be easily embedded in HTML/XHTML, allowing an easy detection by search agents (Powell 2003)
- depending on specific needs, extensions and element refinements may be added to the standard metadata set defined by DC.

DC also provides a set of encoding schemas that may be used to identify, e.g. through vocabularies of terms, the possible values that metadata may assume. In the present work an ontology-based encoding schema for the DC:subject metadata field is proposed.

3.3 The knowledge model

The metadata set should not only annotate information resources, but must also place them in the context of a knowledge scheme. A knowledge scheme can be effectively represented by an ontology, i.e. by a formal and agreed description of a knowledge domain in terms of concepts and relationships. Therefore, mapping information items to ontology concepts provides a homogeneous view over the repository of information sources and allows identifying, through the ontology network, new relations among resources and concepts.

An ontology can be seen as a collection of concepts (also called *classes*) and properties (also called *slots*). Properties may describe relationships to other concepts. In particular, an inheritance relationship (*IsA* or *SubclassOf*) allows the building of a hierarchical view of the ontology (i.e. a taxonomy), that is particularly suitable to the knowledge classification task that is at the basis of the present work.

The DC:subject metadata field maps information resources to domain concepts. The whole of information resources and domain concepts is nothing but a knowledge base. Creating a knowledge base involves adopting a knowledge model in order to achieve interoperability with other knowledge representation systems and to enable knowledge sharing and reuse.

Many ontology representation languages exist, however, in the scope of the present work, the RDF knowledge model has been considered. RDF (Resource Description Framework) is a knowledge representation language defined by W3C with the aim of enabling software agents to directly process web resources (W3C 2006). The RDF knowledge model is based on a simple predicate logic that defines relationships between resources. The RDF Schema (RDFS), that is an integral part of the RDF recommendation, allows defining application-specific vocabularies of concepts (classes) that can be used by RDF to describe the knowledge items (instances). RDFS classes can be hierarchically organized using the *rdfs:subClassOf* relationship. Moreover, since a class may have more than one parent, a concept can be located going through different inheritance paths. This is an important aspect, because it allows integrating in a unique ontology different views of the same knowledge domain. The only main concern with RDFS is the impossibility of modelling axioms. However, this potentially serious limitation of RDF can be overcome either by adding extra application layers on top of RDFS or by modelling axioms as RDF objects (Staab et al. 2002).

3.4 User roles and workflows

Creating and using an ontology-based repository of information resources involves different kinds of activities, which are described hereafter.

1. Ontology Editing
   Creating an ontology that encodes a given knowledge domain is the first activity required to build the repository. It is quite a long and laborious task even for domain experts and it should rely on a specific tool (ontology editor) that facilitates the design, the construction and the consistency check of the ontology. Also, using a version control system is highly recommended in order to keep track of the whole ontology development roadmap.

2. Knowledge acquisition
   As soon as a first version of the ontology is available, the knowledge acquisition process can start. Knowledge acquisition is essentially a collaborative task carried out by a community of users that submit relevant information items to the repository, classifying them on the basis of the available ontology concepts.

3. Knowledge validation
   All submitted information items should be validated by trusted users prior to making them available for knowledge retrieval. The validation involves both an evaluation of the resource itself and a full review of the associated metadata.

4. Knowledge retrieval
   Knowledge processing encompasses all activities concerning the utilization of the repository, such as semantic search, automatic generation of reports, etc.

5. Ontology maintenance
   Ontology maintenance is assumed to go on throughout the entire lifetime of the knowledge repository and relies on the contribution of all users. In the case that the users operate in heterogeneous contexts, the community of practice paradigm may be useful to encourage knowledge sharing leading to the creation of a common ontology. The ontology maintenance process does not involve direct modifications of the ontology, instead it produces a feedback directed to the people in charge of editing the ontology. Ontology maintenance should not prevent the normal usage of the repository.

A minimal set of user classes has been identified and is reported in table 1. In figure 2 is depicted the interaction between the users and the main functional blocks.

Table 1: user classes and roles

User Class	**User roles**				
	knowledge acquisition	**knowledge retrieval**	**knowledge validation**	**ontology maintainance**	**ontology editing**
Domain Expert	√	√	√	√	√
Trusted User	√	√	√	√	
Generic User	√	√			

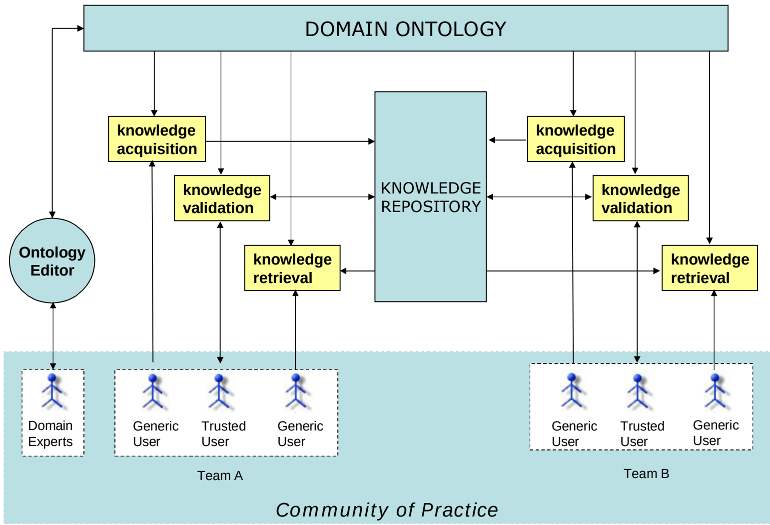

Figure 2: user classes and functional blocks

Workflow management, which is supported by most of the CMS, should be tailored according to the possible states a knowledge item may assume, as shown in figure 3. Any knowledge retrieval activity may only access knowledge items that are in the "valid" state.

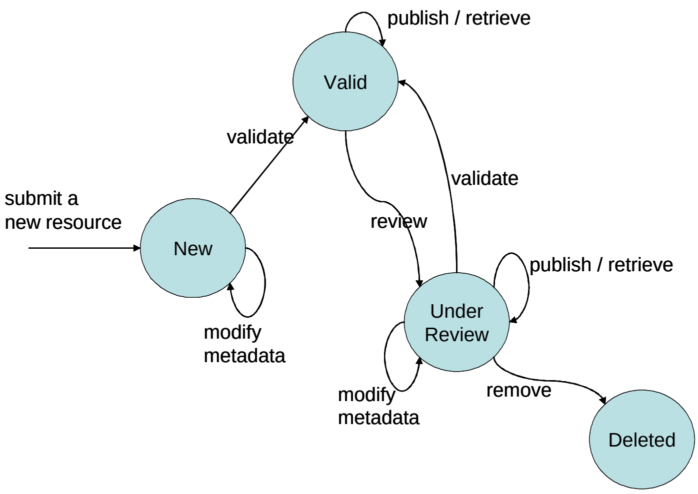

Figure 3: possible states of a knowledge item

4. Key issues

4.1 Encouraging knowledge acquisition

The intrinsic value of the repository is strictly correlated to the amount and quality of the information resources it contains. The possibility to submit new resources in a quick and easy way as soon as they are evaluated by the users is vital. If the knowledge acquisition phase involves laborious and unfriendly tasks, users will be reluctant to add new resources and the repository will never reach its critical mass in a reasonable time. Therefore, only a minimal set of DC metadata elements should be required to be provided when a new resource is added to the repository, while the remaining metadata may be added later on. The user should always be able to switch to a full metadata entry form, containing all metadata fields.

The use of automatic metadata detection tools is also recommended. However, in order to avoid affecting the performance of knowledge acquisition, these tools should operate in background.
Finally, it is important that the choice of DC:subject metadata term is done using techniques that allow an easy identification of closely related concepts (e.g. using a taxonomy navigation tool).

4.2 Dealing with ontology evolution

Ontologies represent a point of reference in the attempt to structure information resources within an agreed knowledge scheme. However, ontologies are not necessarily static and may generally evolve for different reasons, e.g. because they reflect knowledge schemes that are by nature dynamic, or because the more skilled the domain experts become, the more thoroughly ontologies are detailed. A big challenge in the design of an ontology-based CMS is the ability to cope with ontology changes that may occur during the lifetime of the system (Stojanovic et al. 2002).

The main issue in managing ontologies changes is to avoid invalidating any resources already present in the repository. Another concern is how to update the knowledge base after some parts of the ontology have been improved. The possible operations that involve ontology modifications are addressed by the following cases:

1. Adding a new class.
   Adding new classes is a common practice that occurs during ontology maintenance, e.g. because some concepts need a more detailed classification or because the scope of the domain has slightly changed. Adding a new class also involves specifying relations with other classes, in particular its position in the taxonomy tree (i.e. identifying the super-classes the new class inherits from).
   After a new class has been added to the ontology, there is a possibility that knowledge instances related to parents or siblings of that class may be better reclassified. These instances remain valid but should be marked as “pending for review”, in order to highlight them in a future revision process.

2. Removing a class.
   Removing a class is a potentially dangerous operation that may cause inconsistencies in the knowledge base. For example, removing a class that is referred by some knowledge instances ends up in an inconsistent status of the knowledge base. Therefore, removing a class of the ontology can be allowed only if that class is no longer referred by any knowledge instance. In most cases it is preferable to mark the class as obsolete.

3. Making a class obsolete.
   An obsolete class cannot be used anymore to index new knowledge instances. Existing instances that are related to obsolete classes are marked as ‘pending for review’.

4. Modifying semantics.
   Modifications in the semantics (e.g. altering the taxonomy tree) can be allowed at any time, since it does not cause inconsistencies in existing instances. However, the same considerations made for adding a new class shall be applicable in this case too.

5. Renaming a class.
   Renaming a class should not cause particular problems as long as it maintains a unique identifier in the context of the knowledge base.

The design of a strategy for managing the ontology evolution should consider the above described cases. As a general rule, every time a new version of the ontology is produced it should be verified whether the new ontology version causes incongruences in the knowledge base. If so, an explanatory report should be produced in order to allow domain experts to build a correct version of the ontology.

5. Prototype application framework

5.1 General architecture

A prototype application framework, named ORKO (Ontology-based Repository of Knowledge Objects), was developed with the aim of (1) experimenting and validating the solutions proposed in the present work; (2) highlighting unexpected issues; (3) laying the foundations of the prototype evolution.

The entire prototype, whose building blocks are outlined in figure 4, was totally developed using open source components. In particular, Plone (Plone 2006), a CMS based on Zope application server, was chosen because it provided full support for workflow management, creation of custom content types and web publishing. Protégé (Protégé 2006), a knowledge based framework written in Java, was adopted as ontology editor. Ontologies were exported to an ontology server and made available remotely through XML-RPC web services.

Although Protégé already includes all required tools to build a knowledge base, it was originally conceived as a centralized application. A web-based distributed paradigm was definitely more appropriate for the ORKO prototype, whose primary purpose was to serve communities. Therefore it was preferred to rely on a traditional CMS for storing, searching and publishing knowledge items, using Protégé only for the tasks related to ontology creation and maintenance.

The ontology server was designed to provide access to different versions of the ontology. In fact, it may happen that the ontology version in use by the CMS is not the latest available, owing to inconsistencies that have arisen during the evolution of the ontology itself. The trusted users that are in charge of validating the knowledge repository shall be able to switch to a new version of the ontology only upon evaluating the actions that such an upgrade involves.

The ontology maintenance process was supported by Zwiki (Zwiki 2006), a wiki engine based on Zope and fully interoperable with Plone, that provided an informal discussion tool within the community of practice for sharing ideas about ontology evolution, enhancements and defect fixing.

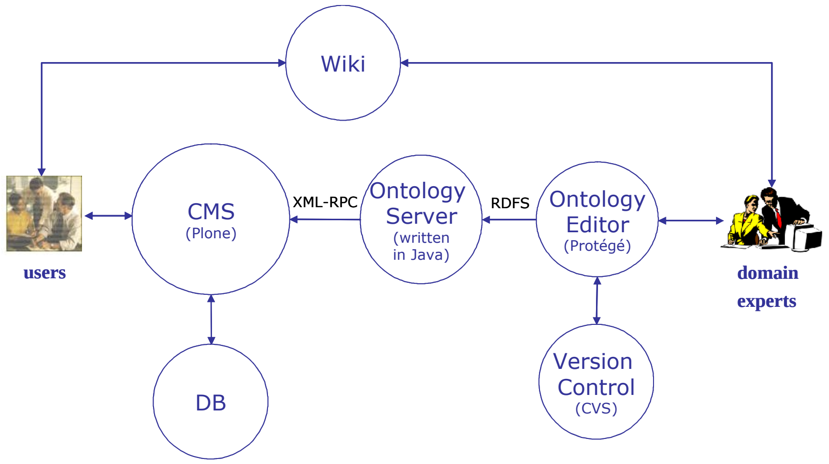

Figure 4: general architecture of the ORKO prototype

A sample screenshot of the ORKO prototype, related to the knowledge acquisition phase, is shown in figure 5.

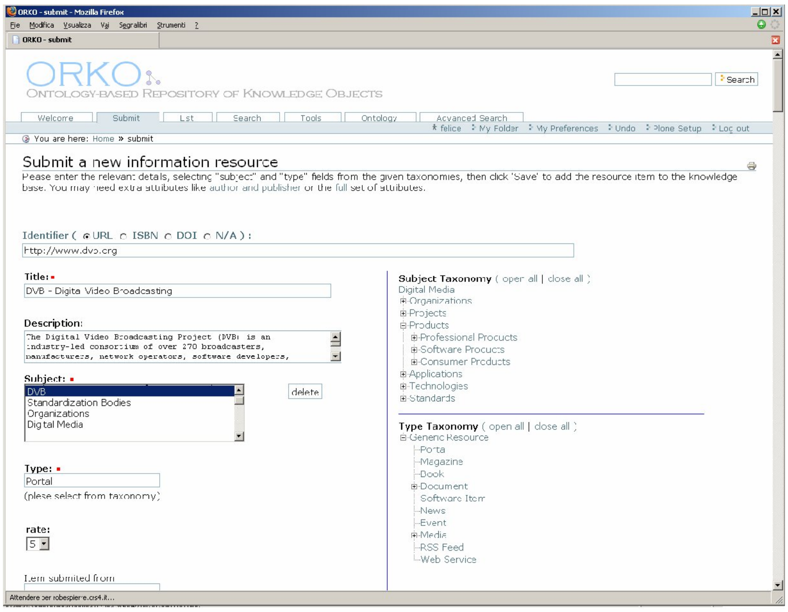

Figure 5: sample screenshot of the prototype application

6. Evaluation and implication for future work

The ORKO prototype was tested for a 10-month period by two small teams working on a joint research project related to multimedia content delivery. At the end of this period, about 1200 resource items were available in the repository and the ontology was made up by about 550 classes. Approximately 85% of the repository items were online resources. During the evaluation period many improvements were made in the prototype upon users' suggestions.

Although the ORKO prototype showed to be a profitable tool for sharing knowledge within a small research community, a number of problems were identified. Firstly, in spite of the enhancements made during the test phase, usability was still an important issue and users were not encouraged to actively participate in the knowledge acquisition process. In particular, the necessity to navigate through a taxonomy tree in order to find the required description tags (rather then entering directly some text) was reported as a disadvantage by most skilled people. Secondly, the ontology maintenance process lacked a real involvement of users, who preferred in many cases to produce an imprecise classification of the resources rather than undertake a discussion within the community of practice about the required ontology enhancements. Finally, the need to validate every resource item by a trusted user could raise scalability issues as the number of resources increases.

Future work should look into a different approach to overcoming the above described issues. An interesting possibility could be adopting a collaborative classification system based on freely chosen tags. The ontology would be no longer visible to the users, however it would still play an important role in background, mapping the many user-generated tags to a formal representation of a common knowledge domain. Moreover, since these tags can also contribute to improving and extending the ontology, all users would be implicitly involved in the ontology maintenance process.

7. Acknowledgements

(1) “Driving Innovative Exploits for the Sardinian Information Society” (DIESIS) project funded by the European Regional Development Fund (ERDF); (2) "Distributed Architecture for Semantic Search and Personalized Content Delivery” project funded by the Italian Ministry of Education, University and Research (MIUR).

References

Allee, V. (2000) "Knowledge Networks and Communities of Practice", *OD Practictioner*, Vol.32, No.4, [online], http://www.odnetwork.org/odponline/vol32n4/knowledgenets.html

DCMI (2006) "Dublin Core Metadata Initiative Organization", [online], Dublin Core Metadata Initiative, http://dublincore.org/

Hansen, M.T., Nohria, N., and Tierney, T. (1999) "What's your strategy for managing knowledge?", Harvard Business Review, Vol. 77, No. 2, pp106-116

Lesser, E.L., and Storck, J. (2001) "Communities of Practice and organizational performance", IBM Systems Journal, Vol. 40, No. 4, pp831-841

Maier, R. (2002) "State-of-Practice of Knowledge Management Systems: Results of an Empirical Study", *Revue des Organisations Suisses d'Informatique*, No. 1, pp14-29.

Powell, A. (2003) "Expressing Dublin Core in HTML/XHTML meta and link elements", [online], DCMI, http://dublincore.org/documents/dcq-html/

Plone (2006) "Plone: A user-friendly and powerful open source Content Management System". [online], Plone Foundation, http://plone.org

Protégé (2006) "The Protégé Project", [online], Stanford Medical Informatics, http://protege.stanford.edu

Riedl, R. (2002) "Some Critical Remarks in Favour of IT-based Knowledge Management", *Revue des Organisations Suisses d'Informatique*, No. 1, pp44-49.

Staab, S., Erdmann, M., Maedche, A., and Decker, S. (2000) "An extensible approach for Modeling Ontologies in RDF(S)", AIFB, University of Karlsruhe, [online], http://www.aifb.uni-karlsruhe.de/WBS/Publ/2001/AEAfM_sstetal_2001.pdf

Stojanovic, L., Stojanovic, and N., Handshuh, S. (2002) "Evolution of the Metadata in the Ontology-based Knowledge Management Systems", [online], University of Karlsruhe, http://www.aifb.uni-karlsruhe.de/WBS/sha/papers/evoMeta.pdf

Swartout, B., Patil, R., Knight, K., and Russ, T. (1996) "Toward distributed use of large-scale ontologies", *Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp32.1-32.19

W3C (2006) The World Wide Web Consortium Organisation, [online], The World Wide Web Consortium, http://www.w3.org

Zwiki (2006) "A powerful Zope 2-based wiki engine", [online], Joyful Systems, http://zwiki.org