Semantically Collaborative Knowledge Management System

Nuno Bettencourt and **Paulo Maio** and **Ricardo Almeida** and **Nuno Silva** and **João Rocha**
School of Engineering - Polytechnic of Porto
Porto, Portugal
{nmb,paom,ral,nps,jsr}@isep.ipp.pt

Abstract

This paper describes the architecture of a knowledge management system that exploits social and semantic web technology in order to ease and promote user-centered modeling, evolution, sharing and access to knowledge in distributed and heterogeneous repositories. For that, several knowledge engineering processes are improved in order to exploit social information between users and communities.

Introduction

Knowledge management employs and applies social policies, procedures and technology to reach its goals (Nonaka & Takeuchi 1995; Prusak 2001). Knowledge engineering technology aims at supporting social policies (*e.g.* access control, collaboration, accountability) and procedures (*e.g.* acquisition, evolution and sharing of information) between members of knowledge communities and between communities.

Despite community-wide formalization of knowledge policies, procedures and technology, knowledge communities tend to complement the specified formalisms with ad-hoc behaviors and technologies.

In fact, one of the main challenges these type of systems face, is related to the ill-specification of the domain description, required by the ever-evolving nature of knowledge. Moreover, the community's and the user's knowledge do not evolve homogeneously by policy imposition. Conversely, every user tends to have his/her own conceptualization of the domain, which evolves differently from that of the community's centralized domain description (conceptualization). Often, initiative for merging (or mapping between) both descriptions can take place, giving rise to a new community-wide description of the domain. Yet, the evolution process iteratively repeats. This must be accomplished by providing capabilities to deal with evolving domain descriptions and respective contents.

Moreover, with the increasing usage of the internet, so far uncommon sources of information are being used for solving daily problems and carrying out activities in organizations. Examples of such repositories are corporative web sites, local PC filesystem files, email, wikis, online forums and blogs. As consequence, the community's operational knowledge is disperse, unregistered, informal and often tacit.

In order to take advantage of the huge amount of heterogeneous ad-hoc content widely spread across such repositories, one has to meaningfully combine and integrate the contents of repositories (both formal, semi-formal and informal) in order to return precise results. Additionally, it is necessary to provide the mechanisms such that the user and the community are able to explicit new knowledge, underlying their behaviors and decisions, either by explicitly defining new facts or combining previously existent knowledge.

The overall problem is therefore characterized according to the following dimensions:

- High number of content repositories, which enlarge both the size of the search space and the size of the result set;
- Syntactic, structural and semantic heterogeneity of repositories, leading to:
  - Difficulties in integrating their contents;
  - Ambiguity in the search query;
  - Incompleteness of the result set;
- Evolving domain requirements, promoting:
  - Evolution of structure and semantics over time;
  - Evolution of content over time (*e.g.* update, deletion, access restrictions);
- Decentralization of repositories (*i.e.* no central control exists), leads to ambiguity and incoherence of content between users' and communities' knowledge.

The rational behind this proposal is that supporting and easing the user's knowledge-related tasks in context of specific community(ies) will improve both the user's and the community's knowledge-related performance. Further, acting collaboratively, the community will support access to, improve and generate new knowledge. Accordingly, the system provides the following conceptual features:

- Transparency in data process from multiple, decentralized and heterogeneous repositories and according to the user's and community's requirements;
- Evolution of domain knowledge according to (and combining) the community's and the user's requirements;
- Collaboratively create new knowledge by the combined effort between users, intra and inter-communities;
- Collaboratively recommend knowledge in order to improve the precision of the result set.

The system will exploit Semantic Web and Social Web ideas and technology, namely:

- Domain knowledge description through Ontologies, which will improve the semantics and reasoning capabilities of the system;
- Ontology engineering (*e.g.* Merging, Versioning, Evolution), to support users and communities on the specification and evolution of their domain descriptions;
- Ontology Mapping, in order to support the information integration problems;
- Social Networks, to manage access control to content;
- Social Networks combined with Bookmarking Systems, to reduce the size of the result set and improve its precision.

This project will result on a service that can be used either by humans, automatic agents or other applications, providing an abstraction layer for several data sources where searching, querying, filtering and recommendation are performed transparently, and new knowledge can be created by socially collaborative users.

The rest of the paper is organized as follows. Next section presents the state of the art on technology and ongoing research work to be used on the system. The third section describes the architecture for the proposed system, how the system is layered and which are the most important components. The fourth section addresses aspects related to the use of social networks on the system, as well as to combining them with data access policies. The fifth section describes what kinds of ontology models are to used on the system. Finally, the discussion section gives an overview of the proposed solution and suggests further research.

State Of The Art

This section presents the state of the art in some of the most influencing areas of the system.

Information Indexing and Retrieval

The use of ontologies, classification methods and global identifiers has been used for organizing and creating relations between information in the Internet, but they are not widely applied in knowledge contexts. Recently, applications like Google Desktop[1], Windows Desktop Search [2] and MetaTracker[3] allow indexing and tagging content on local ad-hoc repositories in a more intelligible way. Google Desktop goes one step further by showing simultaneously local and Web content when searching for documents.

There are still many unresolved issues, such as the low precision of information retrieval processes, their incapability to deal with formal repositories and the information resulting from social recommendation, classification and annotation of documents.

Informal Decentralized Repositories

PC filesystem, P2P and web sites (*e.g.* blogs, forums) are examples of informal, decentralized repositories that are huge sources of knowledge. Gathering and recommending information from these semantically heterogeneous (often personal) information repositories is a core goal of the system.

The Wisdom[4] project is a relevant project in the area of Information Indexing and Retrieval through the integration of both the Web and P2P repositories. This project defines a two-layer architecture in which the upper layer specifies a loose-integration relation between peers. This layer is thus responsible for the abstraction of the distribution and semantic heterogeneity of repositories through Information Integration. Instead, Google Desktop is not able to gather information from PCs other than the one is running on. The Nepomuk[5] project aims developing solutions for extending the PC into a collaborative environment, and in this sense its goals are very similar to those of the envisaged system.

Recommendation Systems

However, these tools do not provide any credence about the adequacy and relevance of the results. This problem is addressed in the Trust layer of the Semantic Web stack, and it is partially solved by Recommendation Systems, based on either Document Content, Collaboration Filtering and Social Network. Document Content-based systems are adequate for text-based documents but not for multimedia and highly-subjective and interpretative documents (*e.g.* anecdotes, poetry).

[1] http://desktop.google.com
[2] http://desktop.msn.com
[3] http://www.gnome.org/projects/tracker/
[4] http://www.dbgroup.unimo.it/wisdom-unimo/
[5] http://nepomuk.semanticdesktop.org

Collaboration Filtering instead is very efficient for all kinds of documents because it relies on the analysis of recommendations made by users with similar profile. However, it is very sensitive to malicious attacks and poorly customizable. Social Network-based Recommendation Systems minimizes the Collaboration Filtering problems by exploiting Social Network (*e.g.* FOAF), helping individuals classifying information according to his/her social relationships. Social Network-based Recommendation Systems is a very active research field, for example (Golbeck 2006; Massa & Avesani 2007).

Information Extraction and Emergent Semantics

Information Extraction is the fundamental technology to automatically acquire semantics from information repositories, typically focused on the analysis of text-based documents. Additionally, annotation is a very important concept in this concept. An instance of a annotation is new knowledge representing either:

- Relationships between parts of (unstructured) document and ontology entities (*e.g.* "José Mourinho" isA SoccerManager). This provides the means to drive information processing through ontologies, and Information Indexing and Retrieval services to exploit this information;
- Relationships between documents, such as comments and services.

The concept of annotation is characterized according to many different dimensions, identified and described in (Bettencourt *et al.* 2006). While annotations will be generated through Information Extraction technology, it is our conviction that no solutions exist that support the extraction of semantics based on the user's actions and behavior in a semi-structured environment like PC file systems, P2P and the Web.

Information Integration

Multiple ontology and schema alignment algorithms exists (Euzenat & Shvaiko 2007), exploiting different perspectives of the information repositories (*e.g.* schema *vs.* instances, properties *vs.* sub-class-of *vs.* attributes, syntax *vs.* structure *vs.* linguistic) and technologies (*e.g.* statistics, ML, semantic providers).

Moreover, despite proposals are far from perfect, most of them require the user intervention either at system setup or run time, highly constraining the automation, confidence, accuracy and extension of the Information Integration processes. Furthermore, almost no research exists on the negotiation of the alignment between information repositories, and in particular between ad-hoc repositories. The characteristics of these repositories especially concerning the heterogeneity and informality suggest the need for automatic solutions even if not perfect. It is our conviction that the peer's topology and interrelationships provide relevant information to the Information Integration process.

The Wisdom project does not support the emergence and natural evolution of ontologies. Similarly, Nepomuk project aims to support and drive the user's structuring of information (thus constraining the user), but allows an easy integration between different types of repositories. Neither Wisdom nor Nepomuk are able (or aim) to negotiate the alignments between repositories. Instead, they impose the local repository (user's) alignment to the Information Integration process, thus applying a centralized approach.

Despite the recent research in the domains just identified, less research has been done for their use, combination and deployment in knowledge management scenarios. The currently running Nepomuk project is a very promising initiative similar to the project we envisage, but a very important difference remains: Nepomuk is not concerned with the personal and community evolution of knowledge and its domain descriptions.

System Architecture

This section describes the architecture of the proposed system. The architecture comprehends three layers (Figure 1):

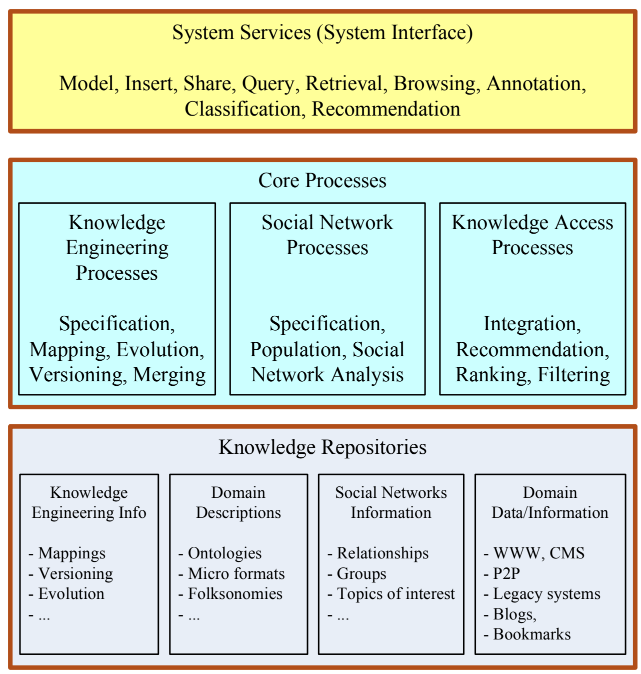

Figure 1: System Core Components

- The System Services, are the functions that the system provides to external entities (*e.g.* users, agents, web services, other applications). These services are high-level services that meet the functionalities that users and other entities expect from the system. While the requirements may change, Services are expected to be adapted according to the user actions over the knowledge, with minimal user/programmer intervention. For example, considering a new information relationship the user creates, a new query command will be automatically provided to the interface, allowing him/her and respective community to access the information in transparent way;

- The Core Processes, are the internal functions that are used solely or combined to manage the repositories content and provide the System Services functionalities. Because these processes are wrapped by the System Services, they can evolve and new processes can be included in the system, improving the System Services in a transparent way. For example, one can imagine a new service that is able to semi-automatically negotiate the mappings between two communities' ontologies, freeing the users from a significative part of that burden task. Users are requested to confirm or reject the negotiated mappings instead of performing the entire mapping process;
- The Knowledge Repositories encompass the content repositories (*e.g.* data, information, documents) and operational modules that provide unified access to those repositories. Only a few repositories are mandatorily part of the system (*i.e.* Domain Descriptions and Knowledge Engineering Information). The content repositories (*e.g.* Social Network and Domain Data/Information) are expected to be dynamically connected (and disconnected) from the system. Due to their technological heterogeneity, they will be accessed through wrapping services. While many types of repositories can be connected, there are only a few different types, allowing a relatively modest effort for providing the wrappers and the connection configuration.

Terminology

According to the descriptions presented so far, it is now advisable to clarify some terms to use in the rest of paper:

Ontology is the artifact that will be used to acquire and represent knowledge domains descriptions, *i.e.* conceptualization of the domain. Ontologies are the common domain representation paradigm in the system. Accordingly, unless otherwise explicitly stated, domains descriptions will be embodied in ontologies and ontologies represent domain descriptions;

Knowledge is the set of ontologies' instances and the resources they relate. Ontologies are therefore templates that relate resources and values. For example, if one wants to state that a specific part of a document D is related to a subject S, he/she will use an annotation ontology that would provide the constructs to link D to S;

Resource is something that can be identified through an URI and therefore can be referred by other resources or entities. Resources face access and collaboration restrictions. In fact, while referred resources are commonly accessible (namely for reading), this is not mandatory;

Values are primitive objects of the system that are assigned to resources' properties contributing to their existence and characterization. Values *per se* have no identity.

Due to their important role in the system, the Core Processes are further described in the next sub-sections:

Knowledge Engineering Processes

Through knowledge engineering processes, users are able to model their domain of knowledge and customize it (*i.e.* evolution, versioning) according to their needs and understandings. Knowledge engineering processes provide functionalities for the management of ontologies (*i.e.* domain descriptions):

- Specification of ontologies (from scratch);
- Evolution of ontologies, including not only changes in a single ontology but also the specification and change in relations between ontologies;
- Versioning of ontologies, according to their evolutions;
- Mapping between different ontologies, including between different versions of the same ontology;
- Merging of different ontologies, giving rise to a single encompassing ontology. Notice that this functionality does not imply the use of a single ontology for the same domain in the system.

Applying and combining social network analysis, domain descriptions and contextualized knowledge, the system will proactively propose engineering transformations on ontologies, including merging, evolution versioning and mapping (these processes will be further described on the Operational Ontologies section).

Knowledge Access Processes

Knowledge sources are structural and semantically heterogeneous. Knowledge access processes allow integrated, contextualized and recommended access to the domain descriptions and knowledge. These processes will act at two levels:

- At the language and model level, which implies the application of normalization mechanisms. For example, while ontologies are the common domain description paradigm in the system, other formats exist, *e.g.* micro formats, folksonomies and glossaries, that are less formal but allow higher and facilitated social participation. Users might profit from domain descriptions using these formats because the system provides normalization components capable to convert between these formats and ontologies (Maedche *et al.* 2002). Moreover, because ontology languages have higher semantic expressivity than those required by the other formats, the normalization is semantically lossless;

- At the semantic level, in which ontology mapping is required, in order to semantically transform the knowledge content between domain descriptions.

Despite its importance in the process, ontology mapping does not provide the envisaged integration capabilities, namely the contextualization and recommendation of the result set. For that it is necessary to exploit the user's Social Relationships and his/her social context.

Social Network Processes

Social network processes manage and provide social relationship information about and between users and communities. The so called communities of interest (*e.g.* forums) acquired a lot of success in answering questions to technological problems, and are increasingly gaining enthusiasts in many different areas (*e.g.* Wikipedia, Del.icio.us). These communities allow the exchange of experiences and problems between similar-interested individuals (knowledge providers) with different expertise.

Such communities have been recently complemented with the concept of Friend Of A Friend (FOAF [6]), promoting social relationships between members. It is our conviction that these social relations contain the missing link concerning the subjectivity and social dimension in information management and recommendation systems. We claim that these social relations can be exploited in classifying and entrusting the information returned by the service. In fact, contextualized, recommended access is accomplished by the combination of Ontology Mapping, Social Network Analysis and Domain Knowledge (Figure 2).

Therefore, acting together, knowledge engineering and social network processes facilitate access to:

- Semantically contextualized knowledge, because the processes consider the semantics of both the user requests and the information;
- Socially contextualized knowledge, because the system considers the information resulting from the participation/collaboration of the communities on the annotation, classification and recommendation of documents and facts.

[6]http://www.foaf-project.org/

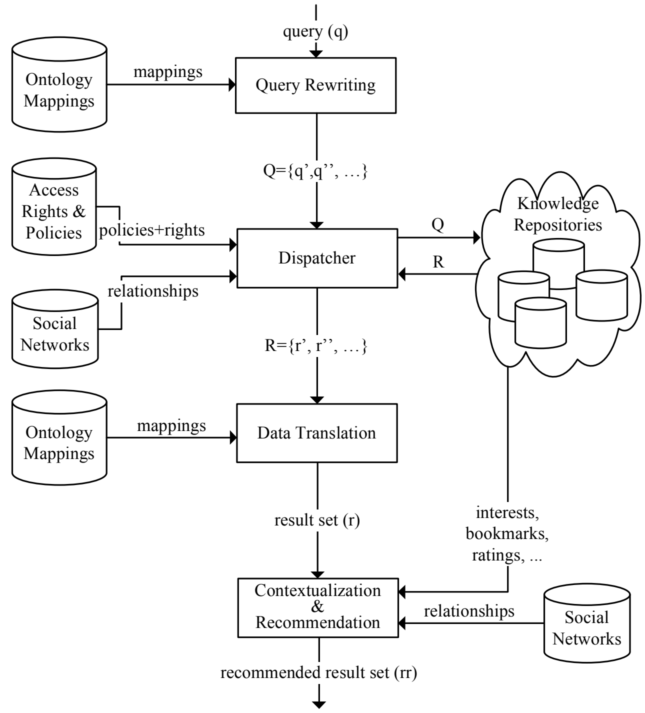

Figure 2: Information Integration and Collaborative Recommendation

Security and Collaboration Policies

This section addresses the collaboration policies in the system. We distinguish between two dimensions:

- The social characterization of users, based on the social network analysis of the available information.
- The access control characterization of resources, and how social networks should interact with policies;

Social characterization of users

The entire framework deals with social networking issues so it is necessary to have access policies based on the following parameters:

- Who, represents who wants to have access to the resource *e.g.* user, agent;
- When, related to the date/time period, to which the user wants the query to be reported;
- Social network, represents the network layer in which who is situated (*i.e.* work, friend, love) used to calculate the trust between who and the resource author.

To maintain a social network, it is necessary to create relationships between users. Consequently it is also important to mention how those relations grow in the system. There are three important ways for users to connect to others on a network:

- by direct invitation;
- by registering or creating an association to a group of interest and afterwards using direct invitation;
- by the system proposed relationships.

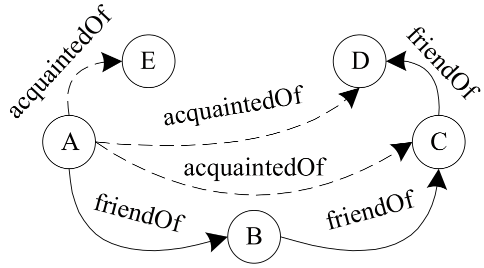

Figure 3: Social Networks - Self Introduction

By using direct invitation (Figure 3), user A invites user C to make part of his/her social network. User A can only invite user C if one of two premisses is achieved: user C allows to be invited by user A if the path between them has a distance less than a threshold defined (*e.g.* 3 in this case) by user C; or if user C is allowing to be invited, even if no path or relation exists between user A and C. On this example, if user D specifies that would only be reached by users with a path distance less than 2, user A could not send him an invitation and user D would not even be seen by user A. Another example, user E does not have any relationship with any user on the social network but might be invited by user A if defined with a discovery mode enabled. For example, notice that the distance calculus has been simplified by only measuring the distance path with node counting. In a real application environment, the distance between users has a much more complex calculation formula.

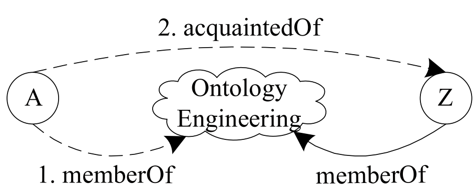

Figure 4: Social Networks - Group Introduction

Using group registering or association mode (Figure 4), user A becomes part of a group to which user Z is also associated. Since both users belong to the same group and therefore have a relationship that connects them through their interests, user A is able to make a direct invitation to user Z. As before, the premisses needed for user A to reach user Z are basically the same, only the distance calculus is different, since there is not a node connecting them, but a group of interest.

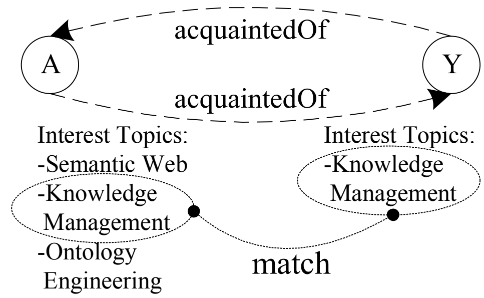

Figure 5: Social Networks - Proposed Introduction

Proposed relationships (Figure 5) is a new form of introduction. To gather information about the user, the system will record user actions, run information retrieval processes over information entered, uploaded documents, bookmarks and any other information that might enrich the user profile. Consequently, users will be suggested of other users in the network with the same tastes, ways of thinking and groups of interest with which they might like to discuss ideas, share information or ask for advices.

The proposed relationships process is automatic, proposes connections based on profiles' similar information (*i.e.* workplace, tastes, groups of interests) and only proposes those that have either set their status to discoverable mode (even if not related by any path) or that are reachable through a path no deeper than the threshold defined.

Access control characterization of resources

By having social networks related to data access policies, it is possible to have resources sharing based on access policies and relationships. Combining these two, information availability and access is defined by policies and based on the user relationship with the author. All the work related to social semantic networks will result in an extension to the existing FOAF-Realm (Kruk 2005) architecture, where multi-domain social networks are covered.

Dealing with a framework like the one described, where data and meta-data can be shared among all users, privacy and data security are some of the problems that automatically arise. Sharing ontologies and instances is one of the core issues. Users must be able to have their public data available and their private data secured (accessed only by people/users/agents they trust). Next are described some of the problems arising with an architecture of this type:

- how and what security policies can be defined over ontologies and their instances? For example, when a user/agent has access to a class, does it have access to its super, sub or sibling classes?
- when a user/agent has access to a resource, how does this applies to the "subsequent" resources (*i.e.* accessible from

the resource)?

It is our believe that some of the ways to resolve the addressed questions should be transformed into options in which users can define their system configuration. This being, a system administrator should be able to define if a user or group has access to system meta-data, or which meta-data they can access, as in hierarchical systems, and any user can define their own policies for the information he/she introduces in the system, if they do not conflict with inherited policies.

Several policy languages exists that can be applied to this framework (*e.g.* KAOS (Uszok *et al.* 2004), RuleML(Boley & Tabet 2007), SWRL (Horrocks *et al.* 2003), REI(Kagal 2002) and REIN(Kagal & Berners-Lee 2005)). Nonetheless, while some of them do not work with OWL ontologies (but provide different kinds of functionalities like logic programming), others are more suitable to application on Web Services access control. Most of the mentioned approaches can handle the work of filtering information based on resources, but none is specialized in information privacy and collaboration, which is part of our work.

Our approach focus in the ability to share ontologies and data between users, thus providing ways to maintain data secured and only accessible to trusted users. In order to grant or deny access to ontology entities and instances, it is mandatory to use a path language and associate them with the level of access to the elements represented in the path. To be able to parse through an ontology in RDF language using path mechanisms, there are several projects like Versa, RDF Path and TreeHugger, but these do not handle OWL ontologies. This limits cooperative ontology building since one of the main goals is for users to compose their own ontology by reusing others' classes and properties, as well as building new ones. It is our believe that using path querying over OWL definitions would be sufficient and enough to simply and quickly address classes, super and sub classes and many important relations. For this to be possible, security over meta-data is also mandatory.

In most information communities, hierarchical and group policies for information permission are typically the only way to hide information from others. Nevertheless, users sometimes want their information to be accessed only by best friends, family members, coworkers or any other dynamic rule (involving relationships), which can change with time (*e.g.* a user wants to send an e-mail to all his/her brothers in law - brothers in law is a dynamic fact that depends on marital status, period of time being referred to, among other factors). Our work in this area aims at using social networks together with data access policies to be able to infer which kind of permission a user has for some resource. This enables dynamic data access policies combined with confidence levels given by social network connections between users.

Operational ontologies

The system is based on the notion of community. A community is composed by users and relates to other communities. A user may belong to multiple communities and have relations to users in other communities.

Every community has a team of managers that define its members, default ontologies, knowledge repositories and access policies. Every user has the chance to use or adapt any ontology, add or remove his/her access to repositories and create, change or remove knowledge and grant access to other users or communities. In doing so, he/she is implicitly creating social relationships with those users and communities in the social network repository (ies). Accordingly, user has (almost) complete control over the knowledge he/she creates, but the user is also responsible for the specified knowledge (*i.e.* non-repudiation).

While domain ontologies are added, changed and removed from the system according to the users requirements, some ontologies are orthogonal to the system and rarely change:

Policies ontology, describes the set of policies and access control parameters to any system resource;

Annotation ontology, describes the concepts and properties related to the annotation process;

FOAF ontology, describes social network concepts and relations, providing the base modeling primitive for social relationships information;

Mapping ontology, describes the semantic relations holding between domain ontologies;

Time and Space ontology, describes time and space concepts, supplying two modeling dimensions that are fundamental in human reasoning process.

Commonly, every concept of every domain ontology is subclass of or relates to the main concept of the previous ontologies. Consequently, every domain concept inherits or relates to these orthogonal dimensions, providing important reasoning elements to the system. Due to their nature and role in the system, the management of these ontologies is transparent to the user. Relationships between domain information and these dimensions is created automatically or specified by the user when requested.

One of the mais problems in the distributed decentralized system is the consistency of the information. In fact, content resources tend to be highly interrelated, but because the knowledge's consistency is not controlled centrally, it is hard to maintain. In order for Knowledge Management to be effective, answers returned by a system need to be time situated, which means that it should be able to keep information history and changes overtime, otherwise connections and relations that existed in the past, might get lost with information evolution and reasoning.

The system will adopt a non-monotonic approach, in which every ontology in the system is time-characterized (Santos & Staab 2003). As consequence, ontology's instances, including the relationships definition between ontology's facts and external resources, are all time characterized. Therefore, in case a resource is deleted or is no longer accessible, the relationship is still admissible in the knowledge base, because it was valid at the time it has been created (*e.g.* the information related to a person's job must exist even if the person no longer have that job). The same rational is valid in case the resource's content changes. In order to reduce even more the identified problems, the system has a crawler-based notification system that periodically checks the status of the resources. In case the resource status change, and whenever possible, the creator of the relationship is notified for a corrective or confirmation action.

Discussion

The proposed system is able to address the user requirements on modeling and customization of domain description, while at the same time promotes semantic and social integration of contents between repositories, users and communities. Our work over collaboration suggests the exploitation of social network for recommending new connections, enabling access policies to available data, and improving query results based on user social network relations.

Until now, there are no systems capable of integrating different data sources and exploit them has one big repository in a semantically and socially contextualized way. By using ontology-based engineering processes and social network information and processes, the proposed system provides a unique vision over several different origins of data, while maintaining repositories independent but socially engaged.

Not much research in the area of relating social networks and bookmarking systems was done, but it is part of our ongoing work. Another envisaged future trend of the system aims at integrating information management features described above and workflow management capabilities. In fact, in addition to the information management provided by the proposed system, organization processes (workflow) are a fundamental part of the knowledge management initiative. While managers specify formal inter and community-wide master workflows, users *per se* and communities often adopt other tacit processes, either replacing or complementing the masters. Acquiring and managing these tacit workflows is an important task due to the often veiled value.

As suggested for knowledge, processes are required to meet the user's and communities' requirements, therefore evolving dynamically with reduced centralized control. Again, merging and mapping efforts can take place, giving rise to new master workflows. The combination of the information management system proposed with the dynamic workflow system is highly required but poorly studied and supported.

The work described in this document is part of the ongoing Portuguese Government funded EDGAR project (POSI/EIA/61307/2004).

References

Bettencourt, N.; Maio, P.; Pongó, A.; Silva, N.; and Rocha, J. 2006. Systematization and clarification of semantic web annotation terminology. In *International Conference on Knowledge Engineering and Decision Support*.

Boley, H., and Tabet, S. 2007. The Rule Markup Initiative. http://www.ruleml.org/.

Euzenat, J., and Shvaiko, P. 2007. *Ontology matching*. Heidelberg (DE): Springer-Verlag.

Golbeck, J. 2006. Combining provenance with trust in social networks for semantic web content filtering. In *International Provenance and Annotation Workshop*.

Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; and Dean, M. 2003. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. http://www.daml.org/2003/11/swrl/.

Kagal, L., and Berners-Lee, T. 2005. Rein : Where Policies Meet Rules in the Semantic Web.

Kagal, L. 2002. Rei : A Policy Language for the Me-Centric Project. Technical report, HP Labs. http://www.hpl.hp.com/techreports/2002/HPL-2002-270.html.

Kruk, S. R. 2005. FOAF-Realm - Control your friends' access to resources.

Maedche, A.; Motik, B.; Silva, N.; and Volz, R. 2002. MAFRA-A MApping FRAmework for distributed ontologies. In *13th International Conference on Knowledge Engineering and Knowledge Management*, volume 2473 of *LNCS*, 235–250. Sigüenza, Spain: Springer.

Massa, P., and Avesani, P. 2007. Trust-aware recommender systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, 17–24. New York, NY, USA: ACM.

Nonaka, I., and Takeuchi, H. 1995. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. New York: Oxford University Press.

Prusak, L. 2001. Where did knowledge management come from? *IBM Systems Journal* 40(4):1002–1007.

Santos, J., and Staab, S. 2003. Fonte: Factorizing ontology engineering complexity. In *International Conference On Knowledge Capture*, 146–153. Sanibel Island (FL), USA: ACM Press.

Uszok, A.; Bradshaw, J. M.; Johnson, M.; Jeffers, R.; Tate, A.; Dalton, J.; and Aitken, S. 2004. KAoS Policy Management for Semantic Web Services. *IEEE Intelligent Systems* 19(4):32–41.