Future Generation Computer Systems 26 (2010) 38–49

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A framework for distributed knowledge management: Design and implementation

Giuseppe Pirró [a,*], Carlo Mastroianni [b], Domenico Talia [a,b]

[a] DEIS, University of Calabria, Rende (CS), Italy
[b] ICAR-CNR, Rende (CS), Italy

ARTICLE INFO

Article history:
Received 30 April 2008
Received in revised form
13 March 2009
Accepted 15 June 2009
Available online 30 June 2009

Keywords:
Organizational knowledge management
Distributed knowledge management
Ontology
Peer-to-peer networks
Virtual office

ABSTRACT

This paper describes a framework for implementing distributed ontology-based knowledge management systems (DOKMS). The framework, in particular, focuses on knowledge management within organizations. It investigates the functional requirements to enable Individual Knowledge Workers (IKWs) and distributed communities (e.g., project teams) to create, manage and share knowledge with the support of ontologies. On the one hand, the framework enables distributed and collaborative work by relying on a P2P virtual office model. On the other hand, it provides a multi-layer ontology framework to enable semantics-driven knowledge processing. The ontology framework allows organizational knowledge to be modeled at different levels. An *Upper Ontology* is exploited to establish a common organizational knowledge background. A set of *Workspace Ontologies* can be designed to manage, share and search knowledge within communities by the establishment of a contextual (i.e., related to the aim of a group) understanding. Finally, *Personal Ontologies* support IKWs in personal knowledge management activities. We present an implementation of the designed framework in the K-link+ system and show the suitability of this approach through a use case. The evaluation of K-link+ in a real network is also discussed.

1. Introduction

In a business ecosystem that is becoming increasingly competitive, knowledge is a critical factor for business activities supporting organizational strategies [1]. Knowledge must circulate among Individual Knowledge Workers (IKWs), who should learn from each other to keep themselves updated and productive. Knowledge Management (KM) is viewed as the set of the technologies, techniques and procedures that are used to assist the creation, access and reuse of knowledge in a collaborative environment [2,3]. Traditional KM systems, based on the "one size fits all" principle, adopt centralized technological architectures. However, since knowledge is the result of different perspectives and social interactions between individuals and groups [4], subjectivity and autonomy of workers is an essential pillar of an effective knowledge creation process. Traditional centralized KM architectures are not well suited to support these aspects.

Recently, Distributed Knowledge Management (DKM) has been proposed as a new vision for KM. DKM is based on the principle that different perspectives within complex organizations should not be viewed as an obstacle, but as an opportunity to foster innovation and creativity [5]. In particular, it becomes crucial for organizations to support the creation of communities of workers in which knowledge can be created, organized and shared. Communities of Practice (CoP) are "places" where knowledge can be created and exchanged [6]. A CoP includes people that share goals and interests and collectively reflect on a problem or an idea. In a CoP, individuals can produce and learn new concepts and best practices, thus allowing the community to innovate and create new knowledge. IKWs access and share knowledge interacting through synchronous (e.g., instant messaging and collaborative editing environments) and asynchronous (e.g., email applications) tools and often work remotely. Current technologies do not properly support this new style of work, so it is increasingly hard to exchange information in a labyrinth of network connections, firewalls, file systems, applications, and IKWs spend much of their time to adapt to their ever changing work environment instead of focusing on their real objectives. What is needed is a flexible work to support the ubiquity of the IKW and enable cooperative work. The "virtual office" approach can comply with this requirement. In a virtual office, the workers collaborate for the most part electronically through the services offered by distributed information systems, though without resorting to physical contact when it is necessary to take decisions and exchange respective views [7].

Overall, the creation and management of CoPs and the support of collaborative work are becoming central for the success of next generation KM systems [8,9]. The Peer-to-Peer (P2P) paradigm can be exploited to fit both these requirements, as it naturally supports the management of communities (e.g., workspaces, peer groups),

---

[*] Corresponding author.
E-mail addresses: gpirro@deis.unical.it (G. Pirró), mastroianni@icar.cnr.it (C. Mastroianni), talia@deis.unical.it (D. Talia).

0167-739X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2009.06.004

and allows content and knowledge to be created, shared, exchanged and transformed through synchronous and asynchronous collaboration. In particular, the P2P technology helps to support the core principles of DKM, that is, *autonomy* and *coordination* of workers. In a modern technological environment, IKWs should be granted a high level of autonomy for the management and exchange of personal knowledge, while coordination assures that the generation of common knowledge is achieved through the collaboration among autonomous entities rather than a process of homogenization [5]. With the adoption of P2P technologies, peer groups, bringing together different people from different organizations, can be formed and dissolved dynamically. This would be much more difficult in a corporate-based infrastructure and/or proprietary network. Finally, P2P architectures, having no single point of failure or central storage facilities, are more suited to guarantee important properties such as fault tolerance and adaptiveness to a changing environment.

This paper presents a P2P framework based on the virtual office model and supports the key characteristics described so far, i.e., autonomy and efficient collaboration among workers, flexible management of communities of practice, and coordinated knowledge generation and exchange. Beyond exploiting the P2P paradigm, these functionalities are also provided by giving a central role to the semantics of information. In fact, in a complex and heterogeneous environment, the integration of knowledge objects and activities is only possible when their semantics is correctly captured. It is more and more evident that organizational performance can be improved by better exploiting intellectual assets [10,11]. In particular, ontologies are used by communities to define conceptual models that allow a precise meaning of symbols to be established and shared among workers [12]. The proposed Distributed Ontology-based Knowledge Management framework, combines the use of P2P technologies, which supports the DKM and the virtual office model, and the use of ontologies, which harness the semantics of knowledge.

However, the definition of a single ontology over the enterprise is difficult, and becomes almost impossible in a distributed environment, where peers are independent of each other. In the DOKMS architecture presented in this paper, ontologies support the management of semantic information at different levels, in order to assure both the autonomy and the intelligent coordination of workers. In particular, knowledge objects are defined and managed at the organizational level (e.g., for basic organizational assets and interests), at the community level (e.g., to model a particular aspect of the organizational knowledge domain) and at individual level (to support personal perspectives about a knowledge domain). Specifically, the *Upper Ontology* layer is exploited to establish a common organizational knowledge background shared by all the IKWs. A set of *Workspace Ontologies* are defined to process and manage knowledge within communities, by the establishment of shared sets of concepts, related to the specific objectives of the communities. Finally, according to the autonomy principle of the DKM, *Personal Ontologies* support IKWs in Personal Knowledge Management activities. The use of ontologies allows links between knowledge objects and ontology concepts to be created. These associations foster the sharing of knowledge and improve the performance of discovery operations aimed at searching the communities for objects having specified semantic characteristics. In a dynamic environment, knowledge must be continuously updated and renovated, thus adequate supports are required to enable ontology construction and evolution. In the DOKMS architecture presented here, a procedure, based on distributed voting, allows the ontologies to be modified and extended with democratic policies.

The described functionalities of a generic DOKMS architecture have been implemented in the K-link+ system. This system was implemented to comply with the requirements emerged in the context of an Italian research project named KMS-Plus,[1] whose main objective was the definition and management of personal and organizational knowledge for the integrated support of enterprise activities. The remainder of the paper is organized as follows. Section 2 describes the architecture of a generic DOKMS. Section 3 describes the layered ontology framework defined to efficiently support the DOKMS. Section 4 presents the K-link+ system, a specific implementation of the DOKMS architecture. Section 4.3 presents a use case of K-link+ in a software company scenario and underlines the benefits of the system. Section 5 presents a performance evaluation of K-link+. Section 6 discusses related work and Section 7 concludes the paper.

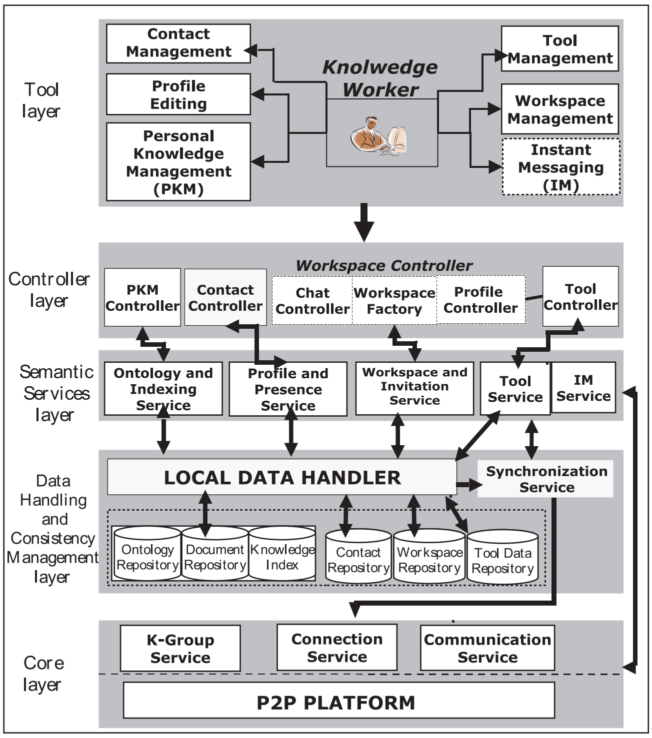


Fig. 1. A generic framework for designing DOKMS.

2. A generic architecture for designing DOKMS

Fig. 1 shows the abstract architecture of the framework. The architecture is based on five layers including basic communication services, data handling services, semantic services, and workspace management services. At the higher level there is a set of tools allowing IKWs to do the actual work. The layers of this architecture are briefly described in the following.

2.1. Core layer

This layer defines the core services whose implementation can be based on any P2P infrastructure. The main services provided by the Core Layer, which are exploited by higher layers, are: the K-Group Service which allows to create new K-Groups (e.g., communities or workspaces); the Connection Service which allows IKWs to join P2P network, and the Communication Service that provides features used to send and receive messages.

---

[1] KMS-Plus, was a research project started as a "pre-competitive" development project financed by the Italian Ministry of University and Research. The outcome of KMS-Plus has been the definition of a semantic-aware Knowledge Management System for SMEs supporting business processes through an integrated view of dynamic and static aspects of enterprise knowledge.

2.2. Data handling and consistency management layer

To favor the autonomy of users, this framework enables different replicas of the same object to be created, so that users can work on their local copies. Since several clients can concurrently work on shared objects, this raises the problem of maintaining data consistency [13,14]. Each IKW can perform read operations, or provisional write operations, directly on its local copy of the object, through the primitives provided by the Local Data Handler. The purpose of this layer is to ensure data persistence, consistency management and synchronization of shared objects. More details on the techniques adopted are given in Section 4.1. Finally, the Local Data Handler manages a set of local repositories to store information about contacts, workspaces and knowledge objects. The synchronization service allows peers to keep their knowledge objects up to date when reconnecting to the network.

2.3. Semantic services layer

The Ontology and Indexing service deals with operations involving ontologies (creation, update) and allows documents to be indexed for keyword-based search. The Profile and Presence Service manages status check operations and enables users to create and publish their profiles within the network. Within these profiles, peers can advertise their expertise in the form of a set of ontology concepts. The Workspace and Invitation Service handles the set up of workspaces and their population, which is performed by sending invitation messages to peers. The Tool Service is used to add new tool instances to workspaces at run time. The Instant Message Service allows peers to communicate each other via a chat system.

2.4. Controller layer

This layer contains a set of controllers that catch operations performed by users and forward them to the underlying layers. The Workspace Controller manages workspace settings through the creation of profiles that contain information about the workspace topics, the set of tools, and the IKWs included in the workspace. The Contact Controller enables peers to discover other peers over the network and add them to a personal Contact List. The PKM Controller is delegated to manage IKWs' Personal Knowledge. The Tool Controller is responsible for allowing users to handle operations (add, update, remove) on tools.

2.5. Tool layer

This layer provides a basic set of tools (document sharing, shared calendar, shared address book, shared sketch pad, shared browser) that can be used within workspaces. In addition, other tools can be developed and included in the system as modular components.

The described framework can be implemented with any underlying P2P architecture. However, we adopted the Sun JXTA [15], as it is widely accepted as the *de facto* standard P2P framework. K-link+ will be described in Section 4.

3. An ontology framework supporting DOKMS

An ontology [16,17] is an abstract representation of a knowledge domain and allows its modeling in terms of concepts, relations between concepts, class hierarchies and properties. Moreover, ontologies permit reasoning about the represented knowledge and offer a way for defining a set of possible instances of concepts and relations to provide links between the model and the modeled reality. In recent years the knowledge management community has been considering ontologies as an adequate support for managing the semantics of information [18]. Next generation knowledge management systems will probably rely on conceptual models that go beyond classical ER models. They will exploit ontologies for defining a precise semantic meaning of a shared terminology. Recently some knowledge management systems based on ontologies have been proposed. The FRODO system [19] exploits ontologies as a mean for knowledge description in organizational memory. Comma [20] combines agent technologies for enabling ontology-based knowledge management systems. Also the problem of building ontology-based systems has been recently investigated in [21].

This section describes an ontology framework to enable distributed knowledge management in organizations. The requirements of this framework emerged in the context of the KMS-Plus project. In designing this framework we faced three critical issues. First, in an organization it is not conceivable to have a single and universally accepted ontology. It is preferable to provide a multi-layer ontology support that allows to: (i) define a quite-static part of organizational knowledge that should be accepted by everyone; (ii) cover specific aspects of the knowledge domain faced by the organization (e.g., in an organizational commitment) that will be deepened when necessary. Hence, the organizational background can incrementally grow up. Second, ontologies in an organization need to evolve continuously [22]. This problem becomes more challenging in a distributed scenario where there are no central entities that handle ontology management operations. Third, a large body of information in an organization typically exists outside the knowledge base (e.g., emails, textual documents, databases). In order to reuse this amount of information, appropriate wrappers have to be provided. These should convert information into an ontological format at an affordable cost. However, this is not an easy task; thus it is necessary to provide a different mechanism allowing a fine grained layer of metadata based on ontologies [12] to be created. This way, it will be possible to improve the quality of knowledge reuse and retrieval. These three aspects will be addressed in the next sections.

3.1. Harnessing organizational knowledge through ontologies

In order to manage the semantics of information in our DOKMS architecture, we designed an ontology framework organized in two layers. This framework is shown in Fig. 2 where concepts are represented as circles and relations as dashed lines.

3.1.1. First layer: The organizational knowledge background

The first layer (i.e., organization layer) contains an Upper Ontology (*UO*) and a set of Core Organizational Knowledge Entities (COKEs) represented as ontology classes. The ontologies contained in this layer aim at modeling the basic knowledge background of an organization. In particular, the *UO* represents a basic set of meta-concepts relevant for an organization, typically defined by domain experts. More formally an *UO* can be defined as:

$$UO = \langle C, \mathcal{P}, H^c, H^p, A, I \rangle$$

consisting of a set of concepts C and a set of properties $\mathcal{P}$ respectively arranged in the hierarchies H^c and H^p that associate each concept c_i with its sub-concepts $Sub(c_i)$, and each property p_i with its sub-properties $Sub(p_i)$. A is a set of axioms. I represents the extensional part of the ontology and contains ontology instances. This definition comply with the features of OWL[2] and RDF(S)[3] ontology languages that provide constructs such as *owl:Class* and *rdfs:subClassOf* to define the classes and their hierarchy H^c, and *rdfs:Property* and *rdfs:subPropertyOf* to define the properties and their hierarchy H^p.

The *UO* can be viewed as a semantic network of concepts similar to a thesaurus. For instance, the *UO* for a health care organization will contain concepts and relations related to diseases, clinical practices, drugs, surgery, etc. COKEs aim at giving a semantic

---

[2] http://www.w3.org/2004/OWL.

[3] http://www.w3.org/RDF.

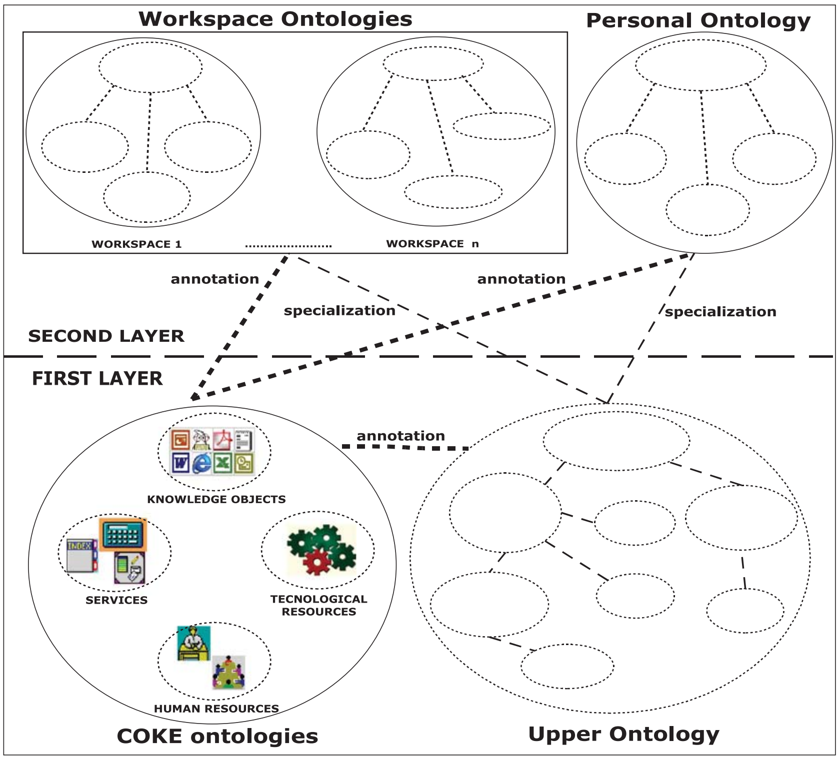


Fig. 2. The user view of the Ontology Framework.

description of well-known organizational sources of knowledge. We identified four COKEs:

- The *Human Resource* COKE describes organizational groups (Community of Practices, Project Teams) and individuals. For each IKW, personal data, skills, group memberships and topics of interest are represented. A group is described through its objectives and topics and contains information about the participant IKWs.
- The *Knowledge Object* COKE describes textual documents, database elements, emails, Web pages, through common metadata (e.g., data of creation, document type, author, URI). In particular, this COKE supports ontology-based content retrieval.
- The *Technological Resource* COKE describes tools through which knowledge objects are created, acquired, stored and retrieved. For each tool, this kind of COKE provides information about its version and features.
- The *Service* COKE describes services, provided by IKWs, in terms of provided features and access modalities. Example of services can be Web services or P2P services such as the JXTA services [15].

Each COKE has its own definition also in terms of attributes. For instance, the COKE *Knowledge Object* (KO), which describes different types of unstructured textual documents, contains attributes such as *name*, *size*, and *author*. Instances of the same COKE share the same structure, so allowing for the management of implicit and explicit knowledge stored in structured, semi-structured or unstructured formats. As shown in Fig. 2, *annotation* relations can be defined between the COKEs and the *UO*. This means that COKE instances can be semantically associated to the concepts of the *UO* by following the principle of superimposed information, i.e., data or metadata "placed over" existing information sources [23]. For instance, let us consider a human resource skilled in Java. An annotation relation can associate the corresponding COKE human resource instance to the Java concept contained in an *UO*. This annotation can be exploited when searching for human resources skilled or interested in Java, for instance, if a group must be created to carry out a particular commitment related to Java programming.

3.1.2. Second layer: Extending the organizational knowledge background

The second layer of the devised ontology framework (shown in the upper part of Fig. 2) is composed of a set of *UO* extensions called Workspace Ontologies (WOs), and one or more Personal Ontologies for each IKW. A *Personal Ontology* (PO) is the specialization of one or more *UO* concepts and is used to deepen a particular aspect of the knowledge domain in which an IKW is interested. More formally, a *PO* can be defined as follows:

$$PO = \langle UO, UOC', UOP' \rangle$$

where the *UO* is the Upper Ontology and UOC' and UOP' are the sets of new concepts and properties added by the IKW. A *PO* operates at individual level as semantic support for personal knowledge management operations. It is defined by the IKWs that use the *UO* and need to extend it for their specific goals in the organizational activities. In order to enhance social aspects of knowledge management, the framework also allows *WOs* to be created. A *WO* specializes one or more *UO* concepts and is used to support cooperative work in a workspace. Even in this case, IKWs can annotate COKEs instances relevant to the workspace to *WO* concepts and retrieve them by semantic search. More formally, a *WO* can be defined as follows:

$$WO = \langle PO, WT \rangle$$

where *PO* has the same structure as the *PO* and *WT* is a set of concepts about workspace topics, on which an agreement among workspace members has been reached. The relations existing between the *UO* and the *WO* and the *PO* ontologies are *specialization* relations, since such ontologies specialize one or more *UO* concepts. In an organization it is not feasible to have a completely predefined modeling of organizational knowledge through ontologies. Therefore, we designed a distributed voting mechanism that enables ontologies to evolve in a collaborative and democratic way. The next section provides an overview of this mechanism.

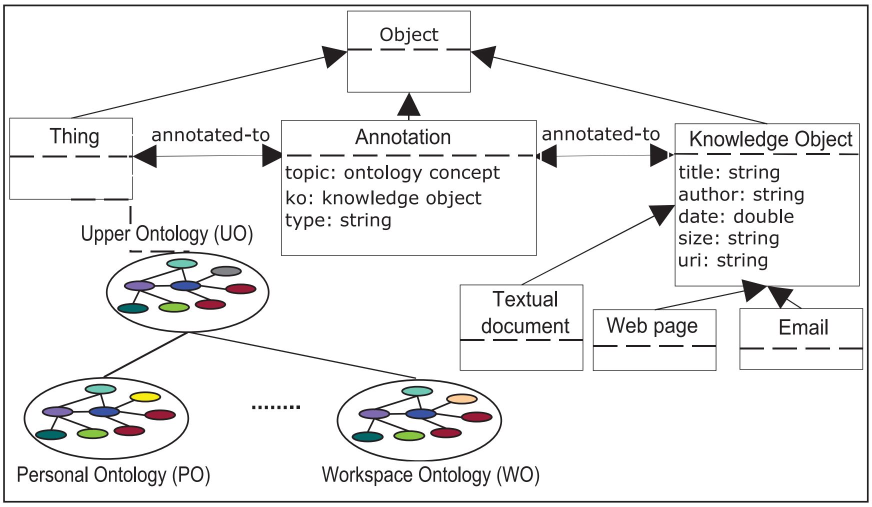


Fig. 3. Annotation of knowledge objects.

3.2. *Handling ontology drift*

Static or fully predefined ontologies cannot satisfy the ever-changing requirements of an organization in a dynamic distributed environment. In the proposed framework, IKWs are allowed to propose extensions or modifications of ontologies (i.e., the *UO* and *WO*) according to their needs. Upon acceptance of such proposals, ontologies evolve in a collaborative and emerging way. Ontology drift, i.e., the evolution of an ontology, is managed through a distributed voting mechanism [24]. In particular, for each voting procedure a voting chair is in charge of permitting or denying the voting process, collecting results and propagating them to participants. Before initiating a new voting procedure, an IKW obtains the authorization from the chair if there are no other voting procedures in progress. An update proposal related to the *UO* is accepted if, within a specified amount of time, the majority of all peer members, regardless of their workspace memberships, agree with the proposal. Similarly, to be approved, an update proposal related to a *WO* needs to be accepted by the majority of the workspace members. A voting process is divided into three phases:

(1) *Set-up phase*: in this phase the voting initiator contacts the voting chair who, if there are no pending voting procedures, forwards a "request for vote" message to all the involved IKWs. This message contains information about the update proposal along with the voting deadline.
(2) *Voting phase*: IKWs vote to confirm or reject the ontology update proposal, and send their vote to the chair.
(3) *Scrutiny phase*: when the deadline expires, the chair counts up the votes and sends the result to the involved IKWs. If the update proposal has been accepted, the *UO* or *WO* is modified accordingly.

When IKWs, which were previously offline, reconnect while a voting procedure is in progress, they are made aware of the voting proposal by the voting chair and can join the voting process. If they reconnect when the voting procedure has terminated, they receive from the chair a notification containing information about the updated version of the ontology.

3.3. *Ontology-based information retrieval*

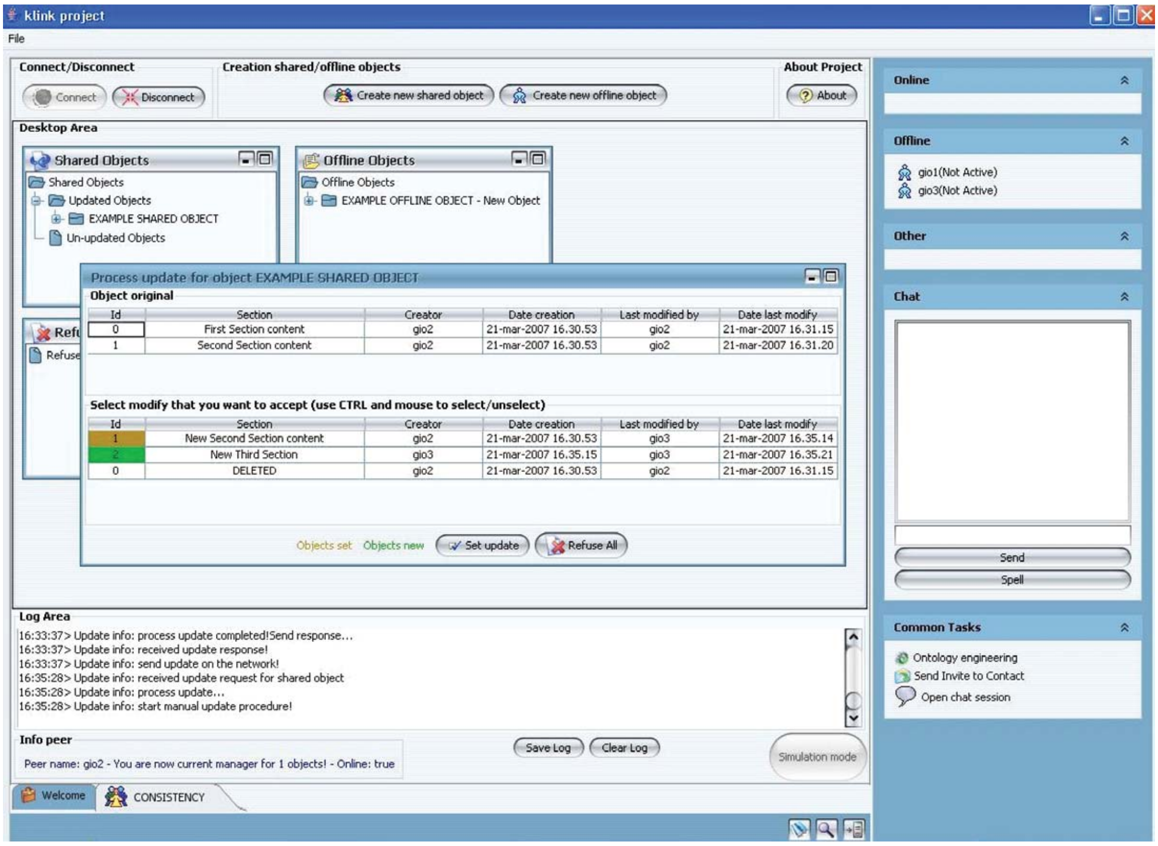
As stated in Section 3.1, annotations can be created between COKE instances and ontology concepts. These annotations are supposed to reflect the content of a particular instance and establish the foundation for its retrieval when requested. In general, semantics-driven information retrieval can be performed using specific tools able to retrieve specific kinds of COKE instances. In particular, we implemented the K-link+ File Sharing tool, which allows instances of Knowledge Objects (KOs) to be retrieved (see Section 4.2). Unstructured information constituting a KO (e.g., a textual file) can be semantically enriched through the annotations, and its retrieval can be performed by specifying ontology concepts instead of keywords. However, it is expected that the annotation process can be automated to decrease the burden of the IKWs. For this purpose, a method based on keyword extraction, as in [25], is adopted. Keywords extracted from the text of the KO can be viewed as descriptors of the KO content. In Fig. 3, the portion of the ontology framework exploited to manage the annotation of KOs is detailed.

As can be noted, annotations of KO instances are handled by an *Annotation* class. This class has two properties, *topic* and *KO*, by which concepts and documents are related together. The property type is used to specify the kind of annotation (i.e., manual or automatic). Overall, the process of retrieving a KO can be summarized as follows. A user annotates its own KOs, thus creating instances of the *Annotation* class. Instances of the *Annotation* class have a property *topic* which indicates the ontology concept that describes the KO. The user can retrieve KOs by choosing a concept of the ontology and sending a request to the peers.

4. The K-link+ system

The architecture described in Sections 2 and 3 was implemented in the K-link+ [26] system, using the JXTA framework [15]. K-link+ features a user-friendly graphical interface both for personal and workspace knowledge management. The workspace perspective is depicted in Fig. 4. Through this graphical interface users have an immediate view on the shared work environment; a user can see the list of IKWs belonging to the workspace, the set of workspace tools (the user can switch among tools by selecting the corresponding tab at the bottom of the workspace interface), a workspace chat application and references to an ontology editor and to the *peer invitation* feature. The creation of a workspace can be done in a few steps. First, a user edits the workspace name and profile; then, he/she populates the workspace by inviting other IKWs belonging to his/her contact list. Afterwards, he/she can select one or more Upper Ontology concepts to which the workspace is related, and is this way starts creating the workspace ontology. Finally, the workspace administrator selects the set of tools useful for workspace members.

In the rest of this section we provide a detailed description of two fundamental features of the DOKM framework implemented in the K-link+ system: shared objects management and distributed search and retrieval. To support these features, we implemented the Consistency Management tool and the File Sharing tool, respectively. The implementations of the tools, described in the



Object original

Id	Section	Creator	Date creation	Last modified by	Date last modify
0	First Section content	gio2	21-mar-2007 16.30.53	gio2	21-mar-2007 16.31.15
1	Second Section content	gio2	21-mar-2007 16.30.53	gio2	21-mar-2007 16.31.20

Select modify that you want to accept (use CTRL and mouse to select/unselect)

Id	Section	Creator	Date creation	Last modified by	Date last modify
1	New Second Section content	gio2	21-mar-2007 16.30.53	gio3	21-mar-2007 16.35.14
2	New Third Section	gio3	21-mar-2007 16.35.15	gio3	21-mar-2007 16.35.21
0	DELETED	gio2	21-mar-2007 16.30.53	gio2	21-mar-2007 16.31.15

```
16:33:37> Update info: process update completed!Send response...
16:33:37> Update info: received update response!
16:33:37> Update info: send update on the network!
16:35:28> Update info: received update request for shared object
16:35:28> Update info: process update...
16:35:28> Update info: start manual update procedure!
```

Info peer

Peer name: gio2 - You are now current manager for 1 objects! - Online: true

Fig. 4. The K-link+ Consistency Management Tool.

Sections 4.1 and 4.2, were exploited to evaluate the system in a real P2P network, as discussed in Section 5.

4.1. The K-link+ consistency management tool

As mentioned in Section 2, an important feature of K-link+ is that it allows IKWs to work on their local copies of shared objects and at the same time it guarantees the consistency of such copies. To manage shared object consistency, we adopt the sequential consistency model proposed by Lamport [27], which ensures that all the updates performed on a shared object are seen in the same order by all the users. In K-link+, this model is implemented by defining three different roles that can be assumed by peers: Manager, Rendezvous and Broker. One or more Manager nodes are assigned to each object and are responsible for authorizing object updates and ordering them sequentially. In particular, each object is assigned a Version Number (VN), which is incremented after each update. A Rendezvous node maintains relevant metadata about the shared objects of a workspace; in particular it maintains, for each object: (i) the list of nodes that can assume the Manager role and (ii) the identity of the Current Manager node, i.e., the node that is currently responsible for the management of the object. Finally, a peer assumes the role of Broker whenever it obtains an updated replica of an object. A Broker can forward a replica to other nodes in a P2P fashion, thus speeding up the propagation of updated objects over the network.

The definition of the above-mentioned roles enables three different kinds of interactions among peers, as shown in Fig. 5. A centralized approach is adopted when Workers interact with the Rendezvous, for example to get information about the VN and the Current Manager of an object. A dynamic centralized paradigm is enabled by the presence of a number of interchangeable Managers associated to each object. Moreover, a decentralized paradigm is exploited by workers to obtain updated object copies by Brokers in a P2P fashion. These three paradigms help to reach a valuable trade-off among different ways to face distributed object management. In particular, two common issues are solved: (i) avoid the presence of a central bottleneck which would be originated if all objects were managed by a single node; (ii) cope with the volatile nature of P2P networks, in which a peer with Manager responsibilities can leave the network at any time. Further details about the techniques used to guarantee consistency in the proposed framework can be found in [28].

The users can create new shared objects (e.g., textual documents) through the consistency management tool. Each object can be composed by a set of sections described by corresponding metadata, e.g., the section name and the peer that created the section. The central part of Fig. 4 shows the user interface of the tool. When a new object is created, the peer chooses a list of peers that are responsible for the object (i.e., the Manager list). The first online peer contained in the list assumes the role of Current Manager (CM).

The peers request updates on a shared object directly to the CM, which can accept or reject the update proposals. In the central part of Fig. 4, some update requests received by the CM are shown. In particular, the CM has to decide whether or not to (i) allow modification of the object with ID 1 (first row); (ii) add a new section to the object with ID 2 (second row); (iii) delete the object with ID 0 (third row). When an update on a shared object is performed, the updated copy is sent to the interested peers.

4.2. The file sharing tool

Another remarkable feature of K-link+ consists in the possibility of performing two kinds of distributed document search:

- *Keyword-based search:* This kind of operation allows users to discover documents on the basis of their content. This is a significant improvement versus the classical P2P file sharing systems in which it is only possible to perform title-based search. To enable this feature, K-link+ exploits the Lucene

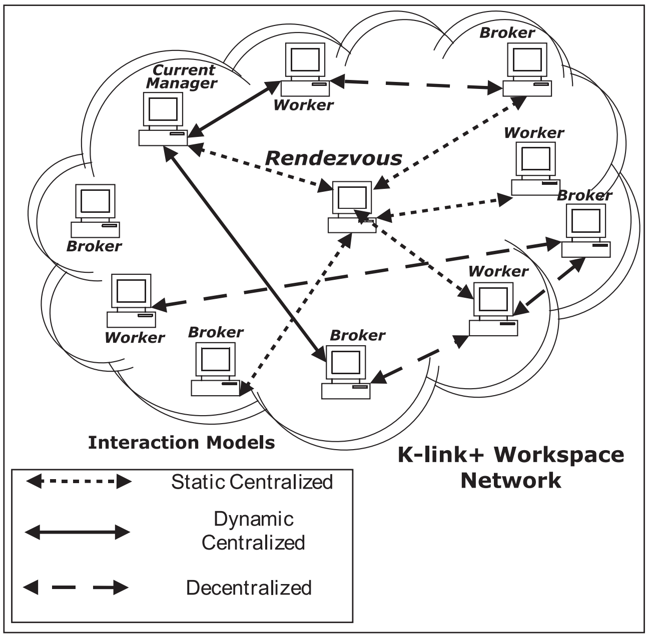


Fig. 5. The K-link+ consistency model.

search engine library [4] that manages the indexing of documents locally stored by a peer.
- *Semantic-based search:* K-link+ allows users to search for documents that have been semantically annotated to ontology concepts. In particular, a peer can select one or more concepts of the shared ontology and send the query to the group. The peers return the documents annotated to the concepts specified in the query.

These search features have been implemented in the File Sharing tool whose interface is shown in Fig. 6. The tool allows for sharing different kinds of files (e.g., doc, ppt, txt, pdf, etc.) and adding them to the workspace or personal knowledge base. The File Sharing tool also allows files to be annotated to both shared ontology concepts (incrementally built through the voting mechanism described in Section 3.2) and a locally defined structure as shown in the left part of Fig. 6. Annotations of documents to ontology concepts can be done by dropping their descriptors over the ontology concepts. Fig. 6 shows the document annotation perspective. The bottom right part of the figure shows the Semantic Table that maintains information about annotations in the form of couples *concept-document* (i.e., instances of the annotation class described in Section 3.3). A document can be annotated to several concepts, thus allowing it to be considered on different perspectives.

Fig. 7 shows how documents can be searched and retrieved by a keyword-based discovery procedure that also analyzes the content of document. The Search Result panel shows, for each discovered document, the relevance (in the *score* field) of the keyword within the document. Results are grouped by peer.

4.3. K-link+: A use case

This section presents a real use-case scenario with the aim to underline possible benefits of using K-link+ in an organization such as a small–medium enterprise. The DOKMS architecture and thereof the K-link+ system is based on the concept of workspace. A workspace can be viewed as a common work area, accessible at any time from everywhere, in which knowledge is represented under the form of COKE instances (e.g., human resources, knowledge objects) annotated to WO concepts. A workspace provides a set of tools for creating and storing knowledge objects and for using services useful for the workspace members. Each IKW can be a member of a workspace under the following profiles:

- A Workspace Manager is a workspace administrator endowed with full capabilities for adding tools, inviting other IKWs or modifying the WO settings.
- A Workspace Participant is a workspace member with reduced but extensible (under Manager control) capabilities.

A workspace set-up procedure can be performed whenever a new organizational task must be carried out. For instance, let us consider a software company that wants to develop a graphical interface, written in Java, able to support the design of business workflows in a distributed environment. To deal with this task and fulfill the commitment requirements, the project leader can set up a proper workspace. The workspace is associated to one or more concepts of the UO, and the WO represents a specialization of these concepts. A configuration of the ontology framework described in Section 3.1, for this example, is shown in Fig. 8.

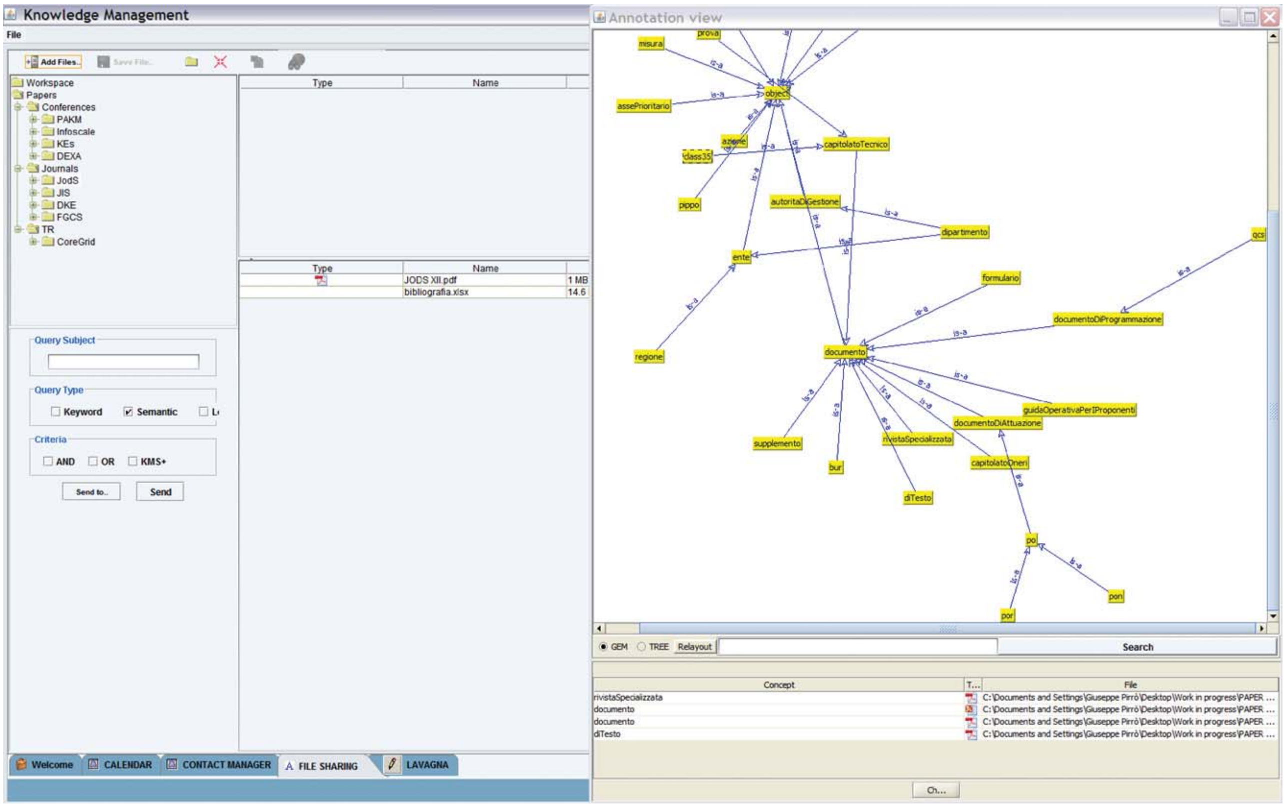
In this figure, the WO specializes the Java concept by adding child concepts related to useful Java graphical libraries. Moreover, the IKW also constructs a PO to which he/she can annotate its personal documents. The creation of a workspace is automatically followed by the creation of a group ontology instance in the COKE human resources ontology. This instance is semantically annotated to the Java UO concept. Hereafter, by using the K-link+ functionalities, the project leader:

- chooses the existing literature and document templates concerning the project topic. In this case it is valuable to populate the workspace document base with knowledge objects related to the concepts defined in the WO. Interesting knowledge objects can be discovered by the K-link+ File Sharing tool that handles keyword and semantic-based searches;
- defines an appropriate team of IKWs whose skills can be exploited to accomplish the commitments. For the above-mentioned example, the K-link+ system should be able to find, through the ontology support, at least the following IKW profiles: experts in Java programming, experts in graphical interface development, and experts in workflow systems. People having the selected profiles become members of the workspace after receiving invitation messages sent by the workspace manager through the K-link+ workspace and invitation service;
- designs an activity plan and assigns single activities to the IKWs by sending proper messages;
- chooses a set of services for supporting the project. For example, the services ontology should contain a reference to a concurrent versions system (CVS) dedicated to the project. Services can be directly embedded in the K-link+ workspace perspective that provides IKWs with a common work environment that gathers the needed applications.

Finally, the workspace manager or its delegates choose a set of tools that can be useful for the workspace members to perform their work. Such tools can be selected among a basic set of tools with which K-link+ is endowed (file sharing, shared calendar, shared browser, sketch pad, etc.). Through these tools, IKWs can set project deadlines, project meetings, exchange documents and so on. Furthermore, it is also possible to develop specific tools that can be plugged into the system as libraries at run time. When a new tool is added to a workspace, the workspace members will automatically be informed and local instances of the tool will be created. Afterwards, each tool update (a new project meeting in the shared calendar tool) will be forwarded to the workspace members that can store new information locally.

---

[4] http://lucene.apache.org.



☐ Keyword ☑ Semantic ☐ L

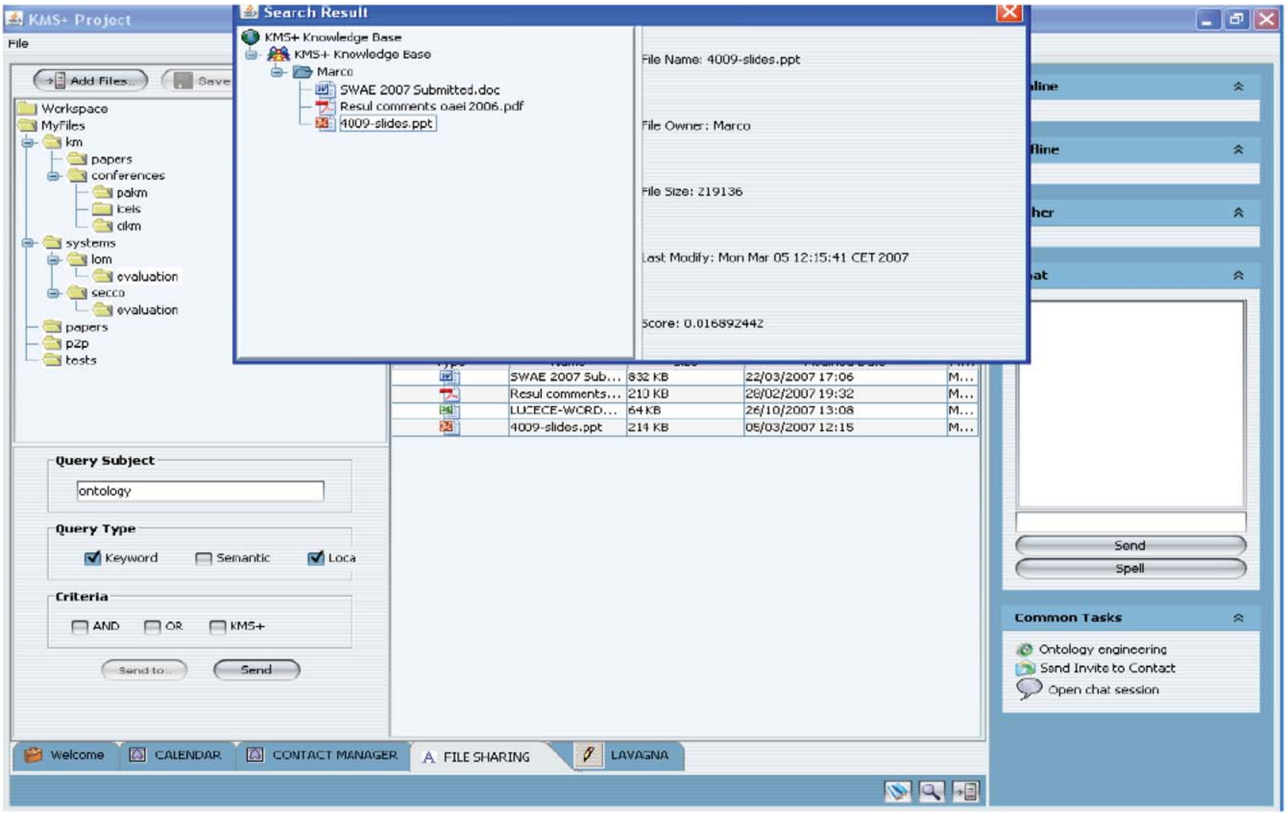
☐ AND ☐ OR ☐ KMS+

● GEM ○ TREE

Type	Name	
	JODS XII.pdf	1 MB
	bibliografia.xlsx	14.6

Concept	T...	File
rivistaSpecializzata		C:\Documents and Settings\Giuseppe Pirrò\Desktop\Work in progress\PAPER ...
documento		C:\Documents and Settings\Giuseppe Pirrò\Desktop\Work in progress\PAPER ...
documento		C:\Documents and Settings\Giuseppe Pirrò\Desktop\Work in progress\PAPER ...
diTesto		C:\Documents and Settings\Giuseppe Pirrò\Desktop\Work in progress\PAPER ...

Fig. 6. The K-link+ File Sharing Tool: annotation perspective.



File Name: 4009-slides.ppt

File Owner: Marco

File Size: 219136

Last Modify: Mon Mar 05 12:15:41 CET 2007

Score: 0.016892442

Query Subject: ontology

☑ Keyword ☐ Semantic ☑ Loca

☐ AND ☐ OR ☐ KMS+

	SWAE 2007 Sub...	832 KB	22/03/2007 17:06	M...
	Resul comments...	210 KB	28/02/2007 19:32	M...
	LUCECE-WORD...	64 KB	26/10/2007 13:08	M...
	4009-slides.ppt	214 KB	05/03/2007 12:15	M...

Fig. 7. The K-link+ File Sharing Tool: search perspective.

Each workspace is described by a workspace profile that includes the UO concept used to specialize the WO, information about the participant IKWs, and services and tools of the workspace. After creating the WO, its concepts can be used

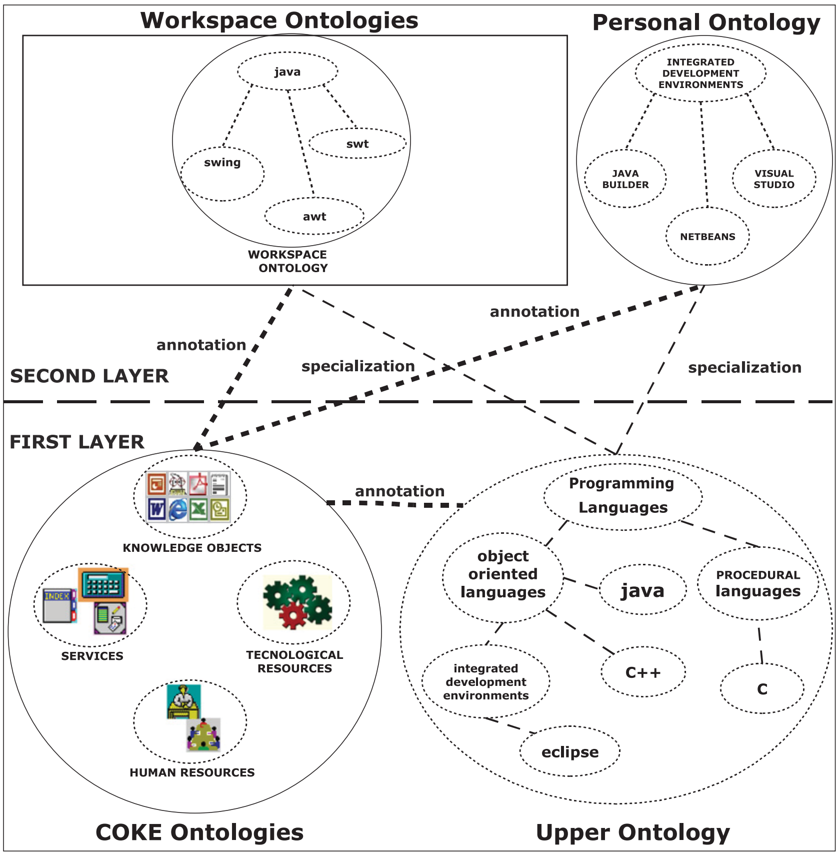


Fig. 8. An example of ontology framework.

for semantically annotating new COKE instances created within the workspace. For example, a tutorial on the use of the SWT Java library can be annotated to the SWT WO concept. K-link+ can be used profitably as an effective cooperative platform in organizations because:

(1) it enables cooperation between IKWs by offering them an integrated and shared work environment in which they can concurrently work on the same shared objects and handle different sources of knowledge within the same environment. This way, the system avoids users to run several applications that cannot exchange data;
(2) it allows contents (described by COKEs) to be provided with an immediate semantic meaning, by means of annotations. The principle followed by annotations is aimed at providing information with a sort of superimposed meaning. This aspect is particularly important since today information is for the most part in unstructured form and its retrieval mainly relies on statistical approaches (e.g., information retrieval approaches) that are not able to “interpret” its semantic meaning;
(3) it enables the reusability of organizational knowledge. For instance, in the described example, if the company will deal in the future with a similar commitment, such as the development of a new Java application, a search can be issued for a workspace that in its profile contains concepts such as Java, SWT, Swing and so forth. Thereafter the project leader can select the documents, templates and human resource profiles that can profitably be reused for the new project.

5. Performance evaluation

In this section we provide a performance evaluation of K-link+ in terms of: (i) distributed semantic search, performed by exploiting the *File Sharing* tool described in Section 4.2; (ii) shared object update, performed with the *Consistency Management* tool described in Section 4.1.

For each evaluation experiment, we set up a real P2P network and simulated the events related to each kind of evaluation, which are normally generated by real users, through event generators. In this way, we simulated the behavior of users in the real system: this allowed us to test different scenarios on varying the number of peers and the activeness of users. Specifically, in the case of distributed search, *send query* events were generated, whereas *request update* events were issued in the case of shared object update. For the exchange of messages we used the JXTA sockets that ensure reliability and automatic message queuing. Although the network overhead in terms of communication time is higher than that experienced with JXTA endpoints, we decided not to adopt the latter technique owing to its unreliability: with JXTA endpoints, we noticed that a considerable amount of messages get lost. The experiments were performed on Windows XP PCs with 3.0 GHz CPU and 1 GB RAM, connected through a 100 Mbps LAN.

5.1. Evaluating distributed search

K-link+ is based on a super-peer (SP) [29] architecture in which peers of the same group/workspace are connected to one SP (a JXTA Rendezvous), and each SP is connected to a set of neighbor SPs. SPs of different groups can communicate each other in a pure P2P fashion. Each SP is responsible for handling requests issued by the peers of its group and forwarding these to the other peers of the group and to the neighbor SPs.

We evaluated the query response time (QRT) as a function of the peer group size and the query generation interval (QIG), that is, the rate at which a peer generates queries. The following table summarizes the parameters of the evaluation. The experiments were performed with a number of peers ranging from 3 to 27 and a query rate ranging from 1 query every 6 s to 1 query every 24 s.

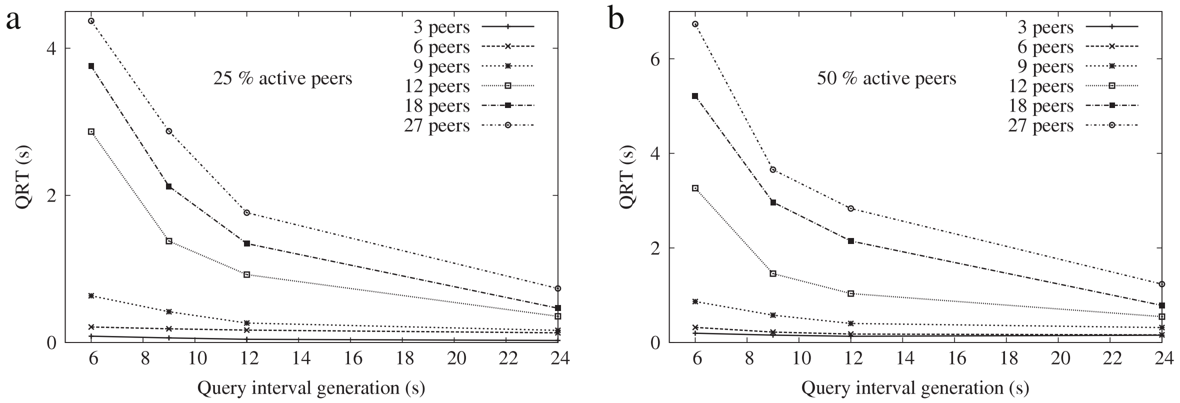


Fig. 9. Evaluating semantic search. Query response time versus the query generation interval time, for different network sizes: (a) 25% of the peers issue queries; (b) 50% of the peers issue queries.

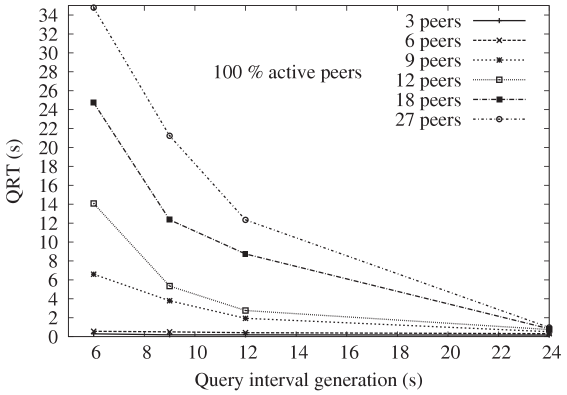


Fig. 10. Evaluating semantic search. Query response time versus the query generation interval time, for different network sizes, in the case that all the peers issue queries.

In the first test of experiments we assumed that 25% of the peers are active and send queries. Results are reported in Fig. 9. It is noticed that the QRT never exceeds 5 s even when the QIG is low (i.e., 1 query every 6 s), which implies that a high number of query messages circulate in the network. The response time decreases as fewer queries are generated by the users. For instance, with a QIG of 24 s, the QRT decreases up to 1.5 s in a network with 27 peers. In the second set of experiments, we doubled the percentage of peers sending queries (i.e., 50% of the peers are active). Fig. 9 reports the results obtained in this case. As expected, the QRT increases, since the number of queries and related messages circulating in the network increases. However, it never exceeds 7 s. Note that with a number of peers not larger than 12, the measured QRTs are comparable in the two above-mentioned scenarios. Finally, we carried out a further evaluation test in which all the peers actively send queries. Fig. 10 shows that the QRT noticeably increases in this scenario. In particular, in the configuration with 27 peers and QIG equal to 6 s, the QRT increases up to 35 s.

These experiments show that the QRT increases with the query rate and the number of participants in a group. The main reason is that in a larger peer group, the SP needs more time to forward the query to all the peers. Moreover, when the number of queries and related messages increases, the SP is more inclined to be overloaded in processing the queries coming from the peers of its group. The configuration with 100% of the peers sending queries and low QIG is not actually realistic, as we also experienced during the KMS-Plus project, but the purpose of the tests was to stress the system and evaluate it in a critical scenario.

In particular, in a network with 27 peers and each peer generating a query every 6 s, the QRT increases up to 35 s, which is a quite large value. However, in more realistic scenarios, in which only a percentage of the peers are active and send queries concurrently, the QRT is limited and never exceeds 6 s.

Table 1
Parameter values adopted to evaluate shared object update.

Number of peers	3 to 27
Number of shared objects	800
Number of Current Managers (CM)	1 to 3
Update request rate	1 each 5 s
Size of the update	1 to 10 kb
Average update request per peer	250
Number of measures for each simulation	10000

5.2. *Evaluating shared object update*

The second evaluation we performed is related to the shared object management approach described in Section 4.1. In particular, we evaluated the average time required for updating an object as a function of the number of available *Current Managers* (CM). The parameters of the evaluation are summarized in Table 1. In order to collect and plot results, we adopted the following approach: we started by considering a single CM (complete centralization) and then incrementally added new peers, up to 27. The peers submit object update requests at the rate of 1 every 5 s. Each update request is related to one of the 800 shared objects, chosen randomly, and the size of each request is randomly selected in the range from 1 to 10 kb. Each evaluation test was stopped after 45 min. We obtained sufficient data to plot the average update time (i.e., the time elapsed from the update request to the time in which the reply from the CM is received and stored) as a function of the number of peers. Further evaluations were performed by considering an increasing number of CMs, which equally share the management of the objects. For instance, if 4 CMs are available, each CM is responsible for 200 objects. Fig. 11 shows the results of the experiments.

It is noticed that the number of CMs does not affect the response time if no more than 18 peers are connected: the time is always equal to about 0.3 s. However, when the number of peers further increases, there is a significant difference between the configuration with a single CM and the configurations with more than one CMs. In particular, in the configuration with 27 peers, the update time is reduced by about 30% by using 2 CMs. Similarly, in a configuration with 3 CMs, the update time is reduced by another 20%. However, if the fourth CM is added, the update time decreases in a much less relevant fashion. We can conclude that, in a configuration with about 30 peers, 2 or 3 CMs are suitable to handle a number of 800 shared objects.

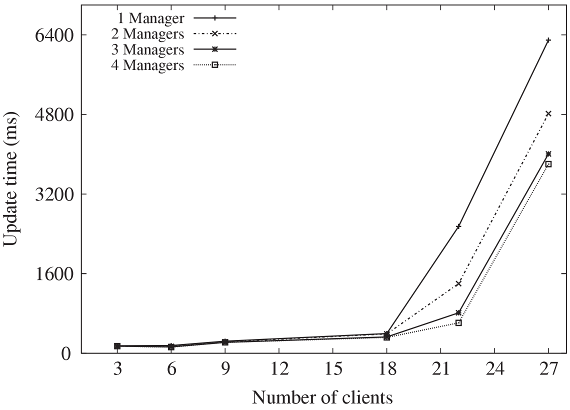


Fig. 11. Average time for updating 800 shared objects as a function of the number of Current Managers and peers.

6. Related work

Recently, some collaborative systems implementing the virtual office paradigm have been developed, e.g., Zoho (http://www.zoho.com) and ThinkFree (http://www.thinkfree.com). However, they are based on a client/server approach and do not include semantic features. The system closest to K-link+ is Groove (http://www.groove.net), an integrated environment for creating distributed virtual offices. Collaboration activities in Groove take place in a shared application space, which is accessed from an application client called transceiver. A shared space, including tools and persistent data, is replicated on every member's computer. Data within a shared space is encrypted, both on disk and over the network, to assure confidentiality and integrity. Both data and commands are transformed, stored and transmitted as XML documents. Every modification made in a shared space is propagated to the other peers. Though its approach is very promising, Groove, differently from K-link+, does not feature semantic functionalities nor exploit ontology mechanisms to cope with knowledge.

A number of systems implement the DKM paradigm and support semantics through the use of ontologies. The KEEx system [5] combines P2P technologies with semantic functionalities. This system, implemented with JXTA, allows a set of K-nodes to exchange information on a semantic basis. Semantics in KEEx is supported through the notion of context, which represents a personal point of view about reality. Users can autonomously create a context to organize their personal knowledge. The system relies on an automatic mapping algorithm that finds the correspondences among the concepts included in the personal contexts of different users. As opposed to K-link+, KEEx leaves the users completely free in the process of context creation and management, without providing them with any organizational background. This may be a weak point in a structured organization. We think that the three level ontology exploited by K-link+ is more suited to guarantee both the autonomy of users and the coordination of their knowledge in a corporate environment.

SWAP (Semantic Web and Peer-to-Peer) [11] aims at combining ontologies and P2P for knowledge management purposes. SWAP enables local knowledge management through a component called LR (Local node Repository), which gathers knowledge from several sources and represents it in RDF schemas. SWAP enables knowledge discovery by using a query language called SeRQL, which is an evolution of RQL. Castano et al. [30] proposed a general framework, called Helios, for ontology-based KM in P2P systems. Ontology matching can be dynamically performed at different levels of accuracy by exploiting the ontologies of peers. Helios also provides a communication infrastructure called Hermes that allows peers to dynamically join community of interests and share their knowledge. Differently from these approaches, K-link+ does not specifically tackle the problem of ontology mapping, which can be time consuming thus affecting the requirement of quickness that is mandatory in a P2P environment. Conversely, K-link+ supports a "mutual agreement" mechanism among the participants, since ontologies are built in a democratic way through a distributed voting mechanism.

The OntoZilla system [31] combines ontologies and P2P technology, with a vision of improving the information search process and facilitating integration as well as interoperability. In OntoZilla, peers supporting the same ontology concept are grouped into the same cluster, and the relations between clusters are modeled according to the concepts in which they are specialized. Therefore, a query belonging to a specific concept can be routed to a suitable group of peers in a systematic way, thus supporting efficient concept search. Moreover, since peer relations are based on peers' expertise that may change over time, the semi-structured system adopted by OntoZilla can flexibly cope with the characteristics of an ever-changing environment.

Another interesting approach is presented in [32,33]. A semantic network is built among the peers, by establishing relations between "semantic nodes". A semantic node can be an entity, a concept, a schema, or a semantic community. A semantic link represents a semantic relation between semantic nodes, such as *instance-of*, *subtype*, *similar-to*, *cause-effect*, and so on. The construction of the semantic network helps to improve the efficiency of discovery procedures, since a query message can be forwarded with higher probability to neighbor nodes having semantic characteristics that match with the target information specified in the query.

The solution presented here for content consistency management in a P2P environment is better detailed in [34]. Current approaches to consistency management in P2P networks depend on the scale of the system. In a large-scale and dynamic system, it is complex and cumbersome to guarantee full consistency among replicas, so researchers designed algorithms to support consistency in a best-effort way. In [35], a hybrid push/pull algorithm is used to propagate updates, where flooding is substituted by rumor spreading to reduce communication overhead. SCOPE [13] is a P2P system that supports consistency among a large number of replicas, at the cost of maintaining a sophisticated data structure. By building a replica-partition-tree (RPT) for each key, SCOPE keeps track of the locations of replicas and then propagates update notifications. In [14], an algorithm for file consistency maintenance through virtual servers in unstructured and decentralized P2P systems is proposed. Consistency of each dynamic file is maintained by a virtual server (VS). A file update can only be accepted through the VS to ensure the one-copy serializability. The K-link+ consistency approach, based on the definition of different peer roles (Rendezvous, Manager, Broker), aims to combine the flexibility of decentralized algorithms with the reliability of the centralized architecture. We think that this is the most equilibrate approach in a small- or medium-scale system, which is the context of interest of K-link+.

7. Concluding remarks

This paper addressed the problem of distributed organizational knowledge management faced in the context of the Italian research project KMS-Plus. Our investigation revealed that in a complex organization, enabling the autonomy of workers and coordinating their knowledge is more effective than superimposing predefined knowledge organization procedures. Hence, a reference framework based on a P2P architecture, which preserves users' autonomy but fosters their collaboration, was presented. The framework presents an alternative way to knowledge management as compared to current centralized architectures based on corporate infrastructures. A striking feature of the framework is represented by the use of formal domain theories, based on ontologies, to model organizational knowledge at the personal, group and organizational levels.

A concrete implementation of the abstract framework is K-link+, a fully fledged system centered upon the virtual office model. The paper showed how the P2P paradigm, the multi-level ontology infrastructure, and the workspace functionalities and tools can be exploited in K-link+ to efficiently handle a specific organization commitment, and tackle important issues such as knowledge drift, object sharing and consistency management. Moreover, the possibility of associating semantic-free content with ontology concepts allows unstructured and semantic-enriched knowledge to be retrieved efficiently.

We are working on extending the semantic capabilities of K-link+ in order to allow multiple and independent ontologies to coexist. However, this will bring in problems of semantic interoperability, since it will become necessary to discover correspondences among the ontologies in order to foster the semantic information flow. In this respect, we are devising an algorithm for matching distributed ontologies that will exploit different kinds of ontological information (e.g., linguistic, structural). Further technical details about K-link+ can be found at the K-link+ project website.[5]

References

[1] Larry Kerschberg, Knowledge management in heterogeneous data warehouse environments, in: Proc. of DaWaK, Springer, 2001, pp. 1–10.
[2] Daniel E. O'Leary, Using AI in knowledge management: Knowledge bases and ontologies, IEEE Intelligent Systems 13 (3) (1998) 34–39.
[3] M. Alavi, D. Leidner, Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues, MIS Quarterly 25 (2001) 107–136.
[4] Fausto Giunchiglia, Chiara Ghidini, Local models semantics, or contextual reasoning = locality + compatibility, Artificial Intelligence 127 (2001) 2001.
[5] M. Bonifacio, Paolo Bouquet, G. Mameli, Michele Nori, KEx: A peer-to-peer solution for distributed knowledge management, in: Proc. of PAKM'02, Springer-Verlag, London, UK, 2002, pp. 490–500.
[6] Jean Lave, Etienne Wenger, Situated Learning. Legitimate Peripheral Participation, University of Cambridge Press, 1991.
[7] D.S. Marshak, Groove virtual office enabling our new models of work, Technical Report, Patricia Seybold Group. 2004.
[8] C. Axton, R. Gear, N. Macehiter, E. Woods, Peer-to-peer computing: Applications and infrastructure. Technical Report, Ovum. 2002.
[9] J. Frazer, A representation of context for computer supported collaborative design, Automation in Construction 10 (15) (2001) 715–729.
[10] John Davies, Miltiadis D. Lytras, Amit P. Sheth, Guest editors' introduction: Semantic-web-based knowledge management, IEEE Internet Computing 11 (5) (2007) 14–16.
[11] M. Ehrig, C. Tempich, J. Broekstra, F. van Harmelen, M. Sabou, R. Siebes, S. Staab, H. Stuckenschmidt, SWAP—Ontology-based knowledge management with peer-to-peer technology, 2003.
[12] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz, Ontologies for enterprise knowledge management, IEEE Intelligent Systems 18 (2) (2003) 26–33.
[13] Xin Chen, Shansi Ren, Haining Wang, Xiaodong zhang, SCOPE: Scalable consistency maintenance in structured P2P systems, in: Proc. of INFOCOM 2005, vol. 3, 2005, pp. 1502–1513.
[14] Zhijun Wang, Mohan Kumar, Sajal K. Das, Huaping Shen, File consistency maintenance through virtual servers in P2P systems, in: Proc. of ISCC'06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 435–441.
[15] B. Traversat, M. Abdelaziz, E. Pouyol, Project JXTA: A loosely-consistent DHT rendezvous walker, http://www.jxta.org, March 2003.
[16] T.R. Gruber, A translation approach to portable ontologies, Knowledge Acquisition 5 (2) (1993) 199–220.
[17] N. Guarino, P. Giaretta, Ontologies and knowledge bases: Towards a terminological clarification, in: Towards Very Large Knowledge Bases, 1995, pp. 25–32.
[18] D. Fensel, Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer, 2004.
[19] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, M. Sintek, Toward a technology for organizational memories, IEEE Intelligent Systems and Their Applications 13 (3) (1998) 40–48.
[20] Fabien Gandon, Rose Dieng-Kuntz, Olivier Corby, Alain Giboin, Semantic web and multi-agents approach to corporate memory management, in: Intelligent Information Processing, 2002, pp. 103–115.
[21] Steffen Staab, Rudi Studer, Hans-Peter Schnurr, York Sure, Knowledge processes and ontologies, IEEE Intelligent Systems 16 (1) (2001) 26–34.
[22] Natalya Fridman Noy, Abhita Chugh, William Liu, Mark A. Musen, A framework for ontology evolution in collaborative environments, in: Proceedings of ISWC, 2006, pp. 544–558.
[23] David Maier, Lois M.L. Delcambre, Superimposed information for the internet, in: Proc. of WebDB (Informal Proceedings), 1999, pp. 1–9.
[24] Yanfeng Ge, Yong Yu, Xing Zhu, Shen Huang, Min Xu, OntoVote: A scalable distributed vote-collecting mechanism for ontology drift on a P2P platform, Knowledge Engineering Review 18 (3) (2003) 257–263.
[25] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, Miroslav Goranov, KIM—Semantic annotation platform, in: Proceedings of ISWC, 2003, pp. 834–849.
[26] Enrico Le Coche, Carlo Mastroianni, Giuseppe Pirrò, Massimo Ruffolo, Domenico Talia, A peer-to-peer virtual office for organizational knowledge management, in: Proc. of PAKM 2006, Vienna, Austria, 30 November–1 December 2006, in: LNCS, vol. 4333, Springer-Verlag, 2006, pp. 166–177.
[27] Leslie Lamport, How to make a multiprocessor computer that correctly executes multiprocess programs, IEEE Transactions on Computers 28 (9) (1979) 690–691.
[28] Carlo Mastroianni, Giuseppe Pirrò, Domenico Talia, Data consistency in a P2P knowledge management platform, in: Proc. of UPGRADE'07, ACM, New York, NY, USA, 2007, pp. 17–24.
[29] Beverly Yang, Hector Garcia-Molina, Designing a super-peer network, in: Proceedings of ICDE, 2003, pp. 49–60.
[30] S. Castano, A. Ferrara, S. Montanelli, D. Zucchelli, HELIOS: A general framework for ontology-based knowledge sharing and evolution in P2P systems, in: Proc. of DEXA'03, IEEE Computer Society, Washington, DC, USA, 2003, p. 597.
[31] Joung Yuh-Jzer, Chuangb Feng-Yuan, OntoZilla: An ontology-based, semi-structured, and evolutionary peer-to-peer network for information systems and services, Future Generation Computer Systems (2008) (in press).
[32] Hai Zhuge, Xiang Li, Peer-to-peer in metric space and semantic space, IEEE Transactions on Knowledge and Data Engineering 19 (6) (2007).
[33] Hai Zhuge, Communities and emerging semantics in semantic link network: Discovery and learning, IEEE Transactions on Knowledge and Data Engineering 21 (3) (2009).
[34] Carlo Mastroianni, Giuseppe Pirrò, Domenico Talia, A hybrid architecture for content consistency and peer synchronization in cooperative P2P environments, in: Proc. of Infoscale 2008, ICST, Vico Equense, Italy, 2008.
[35] Anwitaman Datta, Manfred Hauswirth, Karl Aberer, Updates in highly unreliable, replicated peer-to-peer systems, in: Proceedings of ICDCS, 2003, pp. 76–88.

Giuseppe Pirró is a research fellow at University of Calabria, Italy. He received his Ph.D. from the same university in January 2009. His research interests include semantic web technologies, ontology mapping in open environments and peer-to-peer systems.

Carlo Mastroianni is a researcher at the Institute of High Performance Computing and Networks of the Italian National Research Council (ICAR-CNR) in Cosenza, Italy, since 2002. He received his Ph.D. in Computer Engineering from the University of Calabria in 1999. His research interests focus on distributed systems and networks, in particular on grid computing, peer-to-peer networks, content distribution networks, multi-agent systems. He has published more than 60 scientific papers in international journals and conferences. He currently lectures Computer Networks at the University of Calabria.

Domenico Talia is a professor of Computer Engineering at the University of Calabria, Italy, and the Director of ICAR-CNR in Rende, Italy. He received the Laurea degree in Physics from the University of Calabria. His research interests include Grid computing, distributed knowledge discovery, parallel data mining, parallel programming languages, and peer-to-peer systems. Talia has published four books and more than 200 papers in international journals and conference proceedings. He is a member of the editorial boards of IEEE TKDE, the Future Generation Computer Systems journal, the International Journal on Web and Grid Services, the Parallel and Distributed Practices journal, and the Web Intelligence and Agent Systems International journal. He is a member of the Executive Committee of the CoreGRID Network of Excellence. He is serving as a program committee member of several conferences and is a member of the ACM and the IEEE Computer Society.

---

[5] http://grid.deis.unical.it/K-link.