UNIVERSITY
OF TRENTO

**DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY**

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

# A PEER-TO-PEER ARCHITECTURE FOR DISTRIBUTED KNOWLEDGE MANAGEMENT

Matteo Bonifacio, Roberta Cuel,
Gianluca Mameli and Michele Nori

2002

Technical Report # DIT-02-0065

# A Peer-to-Peer Architecture for Distributed Knowledge Management

Matteo Bonifacio[1], Roberta Cuel[2], Gianluca Mameli[3], Michele Nori[3]

[1] University of Trento, Department of Information and Communication Technology
Via Sommarive, 18 -- 38050 Trento, Italy
bonifaci@science.unitn.it
[2] University of Trento, Department of Computer and Management Science,
Via Inama, 5 -- 38100 Trento, Italy
rcuel@cs.unitn.it
[3] ITC-IRST, Automated Reasoning Systems Division,
via Sommarive, 18 -- 38050 Trento, Italy
{mameli, nori}@irst.itc.it

**Abstract.** Most of the knowledge management systems of complex organizations are based on technological architectures that are in contradiction with the social processes of knowledge creation. In particular, centralized architectures are adopted to manage a process that is intrinsically distributed. In this paper, assuming a Distributed approach to Knowledge Management (DKM), is proposed that technological and social architectures must be reciprocally consistent. Moreover, in the domain of Knowledge Management, technological architectures should be designed in order to support the interplay between two qualitatively different processes: the autonomous management of knowledge of individuals and groups - here called Knowledge Nodes (KNs) -, and the coordination required in order to exchange knowledge among them. Finally a peer to peer architecture to support knowledge exchange across distributed and autonomous KNs is presented.

**Keywords.** Knowledge Management, Distributed Knowledge Management, Distributed and Federated Systems, Organizational Cognition, Peer to Peer.

## 1 Centralized versus Distributed Knowledge Management

In complex organizations, such as networked and multi-national firms - composed by a large number of units, divisions and offices -, knowledge management (KM) is increasingly recognized as an important discipline to enable processes of creating, codifying and disseminating knowledge.

In the last ten years, many companies have invested a lot of resources and efforts in KM projects, whose typical outcome is the creation of computer based systems composed by a Knowledge Base (KB) and an Enterprise Knowledge Portal (EKP) [6]. The aim of these projects is to create large, homogeneous repositories, in which corporate knowledge is made explicit, collected, represented, and organized, according to a single, supposedly shared, system of meanings. This centralized representation - often called knowledge map, ontology, categorization, or classification system - is meant to represent a shared conceptualization of corporate knowledge, and thus to enable communication and knowledge exchange across the entire organization. From an epistemological perspective, these KM systems reflect an objectivistic view of the world where meanings are assumed to be univocal pictures of world objects. It presupposes that all contextual, subjective, and social aspects of knowledge can be eliminated in favor of a unique, objective, and general codification.

Despite of this approach, a lot of theories of knowledge consider subjectivity and sociality as intrinsic dimensions of knowledge (see [20], [9], [16], [15], [14], [25]). They argue that knowledge is the result of different perspective and partial interpretations of world portions or domains, generated by individuals and groups through social interactions. Therefore, it is proposed that knowledge, rather than being viewed as an absolute monolithic matter, is better represented as a system of local "knowledges" continuously negotiated by communities of "knowers" [11].

This different epistemological paradigm leads to a different approach to KM - called Distributed Knowledge Management (DKM) - in which subjectivity and sociality are taken as irreducible aspects of knowledge, and are viewed as a potential source of value, rather than as a problem to overcome [5], [7]. In DKM, an organization is viewed as a "constellation" of local "knowledges" which reside within organizational communities, and live in the interplay between the need of sharing perspectives, in order to incrementally improve performances, and the need of meeting different perspectives, in order to sustain innovation [4]. Thus, the management of knowledge should support and balance to very general organizational principles:

- Principle of Autonomy: each community has a high degree of autonomy to manage its local knowledge. Autonomy can be allowed at different levels. We are mainly interested in what we call semantic autonomy, namely the possibility of choosing the most appropriate conceptualizations of what is locally known.
- Principle of Coordination: each community exchanges knowledge with other units not by imposing the adoption of a single, common interpretative schema (this would be a violation of the first principle), but by adopting a mechanism of meaning translation across different interpretative contexts, called semantic interoperation.

In this article we argue that, in designing KM systems, technological architectures need to be consistent with the social process of organizational cognition (see section 2). Moreover, assuming a distributed approach to organizational cognition, such a need is suggested as a major reason to explain the unsuccessful implementation of current KM systems since based on a centralized view (see section 2.1).

In order two balance the needs of autonomy and coordination across different KNs, namely organizational entities such as individuals and groups which own a "local knowledge", a DKM architecture is proposed (see section 2.2). Finally, based on the concrete setting of a Bank, a peer to peer architecture supporting knowledge exchange among autonomous KNs, is described (see section 3).

## 2 Technological and Social Architectures

As already underlined in [6] and in [24], technology and organizations are interrelated dimensions since each influences and structures the other. Here we underline that the more an organizational process involves high level human activities, such as organizational cognition, the stronger the interdependency between technological and organizational dimensions will be. In particular, since each approach to cognition makes specific assumption on the role of communication, a technology that strongly structures social communication implies a particular model on how cognition occurs [3]. Moreover, we agree with the structurationist approach (for more details see [21], [22]) which considers social and technological architectures as interdependent, rather than independent, dimensions which need to be reciprocally consistent.

In general, we argue that technological architectures can be designed according to two opposite approaches which assume two different social architectures of organizational cognition. The first one (centralized) views organizational cognition as a convergent process that collects peripheral "raw" knowledge, from various sources, and codifies it into a central repository. The second one (distributed), as described by [4], represents organizational cognition as a distributed process that balances the autonomous knowledge management of individual and groups (perspective making), and the coordination needed in order to exchange knowledge across different autonomous entities (perspective taking). In the first case, technology is required to enable central control, standardization, high capacity, and robustness. In the second one it should enable distributed control, differentiation, customization, and redundancy.

### 2.1 Centralized KM Architectures

Briefly we underline some of the main features that characterize a KM system based on a centralized social architecture of organizational cognition (for a more detailed description see [6]). Socially, organizational cognition is viewed as a process whereby people:

- generate knowledge through peripheral socialization in communities of practices [27], [11]: through work practice, employees generate an implicit knowledge in terms of working solutions that can be fruitfully made explicit and thus reusable;
- contribute with their knowledge through a codification process: knowledge is categorized and validated by experts according to a corporate language;
- retrieve knowledge using a unified access to the organizational memory: through the use of manuals, procedures, routines, or the access to formal training, people have access to corporate knowledge.

From a technological point of view, architectures display some common features that are consistent with the above social description of organizational cognition (for a more detailed overview on KM technologies see [13]). According to these organizational aspects centralized KM architectures must:

- create and enable communication within formal and informal groups and communities. On-line interaction processes in terms of social/contextual cues [26] are enabled by a large variety of tools. Indeed also through “virtual communities” and group-ware applications individuals interact among each others and produce their “raw” peripheral knowledge;
- collect “raw” peripheral knowledge through workers participation. They can contribute to create and feed knowledge using automatic document management

tools, clustering, text mining, and information retrieval applications to explicit and collect knowledge;
- categorize and store knowledge in databases and repositories according to a common and shared system of meaning, "distilling" knowledge that is useful for the whole organization;
- design a corporate system of meanings in the form of a common language, an ontology, a knowledge map, a categorization, or a classification system that is necessary to codify knowledge according to a shared interpretative schema;
- create an Enterprise Knowledge Portal (EKP), that provides a unique, simple point of access to corporate knowledge for the members of different organizational units. Most of the time, members access to the KB through various forms of personalization tools (e.g., individual or group profiles, views, chats, and so on).

As observed in a paradigmatic case study of a worldwide consulting firm described in [5], these centralized KM systems are often deserted by end-users. It is shown that any approach which disregards the plurality of interpretative schemas is bound to trouble, as the outcome will be perceived by users either as irrelevant (there is no deep understanding of the adopted and centralized schema) or as oppressive (there is no agreement on the unique schema, which is therefore rejected) [10].

### 2.2 Distributed KM Architectures

According to a distributed approach to KM, technologies should mainly support the autonomous creation and organization of knowledge locally produced by individuals and groups and, on the other hand, support coordination processes among autonomous entities, in order to exchange and share knowledge. In particular this means:
- give to each knowledge owner the possibility to represent and organize knowledge according to her goals and interpretative perspective. Here, a DKM system represents each knowledge owner as a Knowledge Node (paragraph 2.2.1);
- provide tools to support the exchange of knowledge across different individuals without assuming shared meanings but rather enabling the dynamic translation of different meanings. Here a DKM system supports meaning negotiation processes (paragraph 2.2.2);
- set mechanisms and protocols to enable, through cooperation, the emergent and bottom-up formation of informal communities and communication practices (such as finding or addressing people to trusted individuals/communities). Here a DKM system supports the formation of groups and knowledge discovery/propagation through social cooperation (paragraph 2.2.3).

In the following three paragraphs we will describe these elements in more detail proposing some of the main requirements that should be considered in a DKM architecture (see Figure 1). In the next section, such requirements will be framed in the specific context of a knowledge exchange application developed within a Bank.

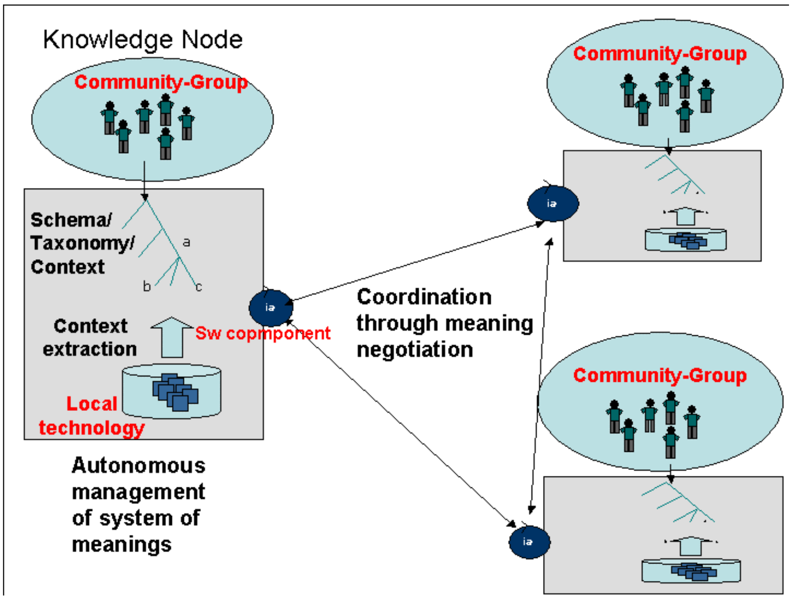


**Fig. 1.** A DKM technological architecture

### 2.2.1 Knowledge Nodes

As we said above, a complex organization can be seen as a "constellation" of groups and communities, and, from a designing point of view, as a "constellation" of KNs considered in [8] as the building blocks of a DKM architecture.

A KN, has been described as the reification of organizational units - either formal (e.g. divisions, market sectors) or informal (e.g. interest groups, communities of practices, communities of knowing) - which exhibit some degree of semantic autonomy. Semantic autonomy means the ability to develop autonomous interpretative schemas (perspectives on the world) to interpret, organize, and store useful information. In other words, each KN is a content interpreted within an interpretative context which is the autonomous manifestation of the node's perspective. Interpretative contexts are both individually held and shared, through negotiation, by groups and communities. Moreover, they become visible by looking at how individuals and groups use technology in order to store, organize and use information. Sometimes groups and communities develop shared repositories where content is stored and organized through common taxonomies (which are here considered as representations of the community's interpretative schemas) while, some other, individual technologies (such as PCs) are considered as the primary artifact for knowledge organization. Thus, when concretely looking at organizations (see the case of the bank after), existing technology and local patterns of use, can suggest us whether to consider as relevant KNs individuals rather than groups or both.

#### 2.2.2 Meaning negotiation

In a DKM approach, KNs are thus materialized by local technologies that represent a semantically autonomous expression of local knowledge owned by an individual or a group. In particular, within such local technologies, we observe, besides the specific features provided to autonomously manage knowledge (such as validation and contribution processes), the availability of a logical element that is relevant in order to support coordination processes among KNs. Such an element, namely a context [12], is a representation schema of the system of meanings adopted within each KN. As proposed in [12], a context is defined as a meta-data schema used in order to interpret the local system of meanings, and make it syntactically comprehensible to others. For example, a context could be the directory structure of a web site or the taxonomy contained in an EKP. Contexts can be used by a DKM system in order to enable knowledge exchange among KNs. In particular, through some technological component which is able to extract, use and compare interpretative contexts, a KN can retrieve information from other nodes even if stored according to a different semantic schema [1], [23], [19]. Moreover, given an information request (query) from a KN A, a software component should be able to contact KNs B and C and to compare the meaning of the query within the context of A with the meaning that emerges within the contexts of B and C[1]. Such a process of meaning assessment and comparison across different categorization schemas is here named meaning negotiation process (for an algorithm of meaning negotiation see [17]).

#### 2.2.3 Cooperation

Given that KNs are semantically autonomous and able to communicate through meaning negotiation, some other organizational capabilities are needed in order to sustain the distributed social process of knowledge creation. According to research on informal communities [11] [27] [18], these organizational features provide a set of social filters based on elements such as trust, membership or shared interests that are necessary to avoid communication overflow while ensuring effectiveness. Here we briefly propose some of these abilities:

- Federate: KNs should be able to spontaneously federate creating groupings and communities of nodes that display a common interest. Such an interest could be given by the goal of maximizing the opportunity to be found by other KNs (being part of a visible group), by the need to certify the type or the quality of a knowledge (through the filtering of members), or the issue to protect content and secure knowledge access from unauthorized KNs (through access policies). K-Federations can simplify interaction processes because a request can be sent to a group rather than to individual KN, decreasing the number of interactions, or because knowledge retrieved from a KN that belongs to a group has presumably a certified quality.

[1] For example, as presented in [8], the query composed by the keyword sequence "antivirus software" in the context of a directory structure available in A "/office-activities/software/antivirus/antivirus-up-date/manuals" has a meaning which can be considered as similar to a document containing the same keywords and stored under the directory of B "/documents/security-system/antivirus/antivirus-up-date/manuals" and not to one stored in C under "/products/soft-ware/anti-virus/marketreports/last" even if keywords are still compatible.

- Discover: KNs should be able to find each other having the opportunity to discover who is available, what type of services are accessible, and what kind of conditions are necessary to establish communication. Such type of capacity should be distributed according to the distributed nature of organizational knowledge.
- Propagate: a KN that looks for information, should be able not to directly interact with all the KNs in order understand who has the required knowledge. Rather, she should be able to trust that a limited number of interactions with neighbors give sufficient chances to find the right counterpart. Such type of capacity becomes possible if KNs are able to propagate requests to other nodes or redirect requesting KNs to potential targets.
- Learn: a KN, that maintains semantic relations with other KNs, should be able to remember such relations in order not to re-negotiate the meaning of a request. Such a knowledge can be also fruitfully used to redirect requests to other nodes or propagate their request to appropriate targets.

In the following section we will propose how these requirements can be implemented in a concrete business case adopting peer to peer technologies.

## 3 Peer to Peer Architecture for a DKM System

In this section it is shown how the functional requirements of a DKM system drive the choice of a particular architectural pattern design (a peer to peer system) and an underlying technology framework (the Jxta Project).

In particular this knowledge exchange system is under development within the business setting of an Italian National Bank[2]. For more details see [8].

In the DKM approach, a great emphasis is given to autonomy and coordination aspects, so that every KN can manage her knowledge, exchange knowledge through meaning negotiation, and cooperate with other KNs in order to achieve her goals. Compared to these aspects a peer to peer system is particularly suitable (see [2]) since depicted with the following capabilities:

- supports autonomy: every member of the system is seen as a peer that manages and has control over a set of local technologies, applications and services;
- is dynamic: peers and resources can be added or removed at any time;
- is decentralized: the community of peers is able to achieve its goal independently from any specific member or component;
- is cooperative: in order to join and use the system, every member must provide resources or services to the others.

From an implementation point of view, we focused on JXTA[3], a set of open, generalized peer to peer (P2P) protocols that allow devices to communicate and collaborate through a connecting network. This P2P framework provides also a set of protocols and functionalities as a decentralized discovery system, an asynchronous point-to-point messaging system, and a group membership protocol. A peer is a software component that runs some or all the JXTA protocols; every peer has to agree upon a

[2] This architecture is under development as part of EDAMOK, a joint project of the Institute for Scientific and Technological Research (IRST, Trento) and of the University of Trento.

[3] A P2P open source project started in 2001 and supported by Sun. http://www.jxta.org/

common set of rules to publish, share and access “resources" (like services, data or applications), and communicate among each other. Thus, a JXTA peer is used to support higher level processes (based, for example, on organizational considerations) that are built on top of the basic P2P network infrastructure; they may include the enhancement of basic JXTA protocols (e.g. discovery) as well as user-written applications. JXTA tackles these requirements with a number of mechanisms and protocols: for instance the publishing and discovery mechanisms, together with a message-based communication infrastructure (called “pipe”) and peer monitoring services, supports decentralization and dynamism. Security is supported by a membership service (which authenticates any peer applying to a peer group) and an access protocol (for authorization control).

These features of a P2P system, and in particular those provided by JXTA, seem to match the spirit and the main functional aspects of a KN in a DKM application, suggesting this architectural solution as a logical choice. In particular:

- KNs can be seen as peers that own a knowledge which is stored and organized through local technologies and applications;
- meaning negotiation can be viewed as a peer query resolution service that involves the use of a local context in order to retrieve or provide knowledge through interaction protocols;
- cooperation can be implemented through the set of dynamic and heterogeneous services that each peer provides to the others in order to, for example, support the discovery of other peers.

In this section we propose a P2P architecture (called Kex: Knowledge Exchange System) which is coherent with the vision of DKM. In Kex (i) each peer (called K-Peer in KEx) provides all the services needed by a knowledge node to create and organize its own local knowledge (autonomy), and (ii) social structures and protocols of meaning negotiation are defined in order to achieve semantic coordination (e.g., when searching documents from other peers).

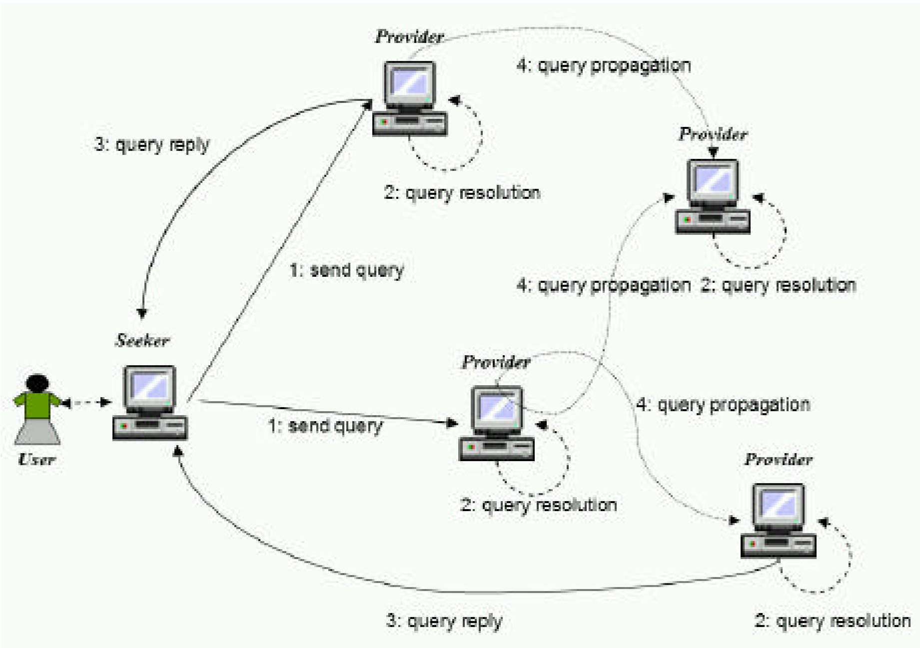


**Fig. 2.** The KEx system: interaction between Seeker and Provider roles.

### 3.1 K-Peer Roles

In KEx every K-Peer has the same main system's functionalities that are provided by two main roles: the *provider* and the *seeker* role. Each K-Peer can act as a provider, as a seeker, or can play simultaneously both roles. Figure 2 shows the interaction between a K-Peer that plays the seeker role and some K-Peers that play the provider role. The following sections explain in details the two roles and their interaction.

#### 3.1.1 Seeker

The seeker component allows the user to interact with other K-Peers and K-Federations (see 3.2.1) in order to search information. The seeker has access to a contexts repository (3.2.2) that contains the local semantic representations of the user's knowledge in the form of taxonomies, category structures or ontologies. Through a context browser, the seeker supports the user in the formulation of the query by selecting the context (or part of it ) which is relevant in defining the meaning of the query. In particular a query is composed by a query expression and one focus. A query expression is a list of one or more keywords entered by the user; the query expression could be empty. The focus is a selection of a portion of a context the user has made. The seeker provides the user with a discovery mechanism (see 3.2.3) to find resources (K-Peers or K-Federations) that could be targeted in retrieving information. The user decides to send the query to some of the available K-Peers and

K-Federations (as showed in step 1 in figure 2). When she submits the query, the seeker activates a session that is associated to the query and is in charge to resolve it using the meaning negotiation protocol (see 2.2.2). When a provider receives a query it performs the query resolution through both a syntactic and a semantic algorithm (see 3.1.2) in order to retrieve relevant documents (steep 2 in figure 2). The provider sends back to the seeker the result set (step 3 in figure 2).

A provider could propagate the query received to other providers that are considered as "experts" about the query's topic (step 4 in figure 2). The active seeker session can receive asynchronously several incoming replies from those providers that have been selected/suggested by other peers, and collects results that are composed by the aggregation of all those that have been received; each result is made up of a list of document descriptors (name of the document, short description, and so on) and the indication of the part of the providers' context (focus) that has been used in order to interpret the meaning of the query and provide a resolution. This relationship between contexts can be used as a learning opportunity (see 3.2.5) that the seeker can store and reuse for other queries. Finally the seeker allows the user to access to the K-Peer download service; if the user finds in the result set one or more interesting documents, she can contact the providing K-Peer to download it.

#### 3.1.2 Provider

The provider is the second main role in the KEx system. It contains all the features required to accept and resolve a query identifying results that have to be sent back to the seeker. When a K-Peer receives a semantic query (composed by keywords and one focus), it instantiates a provider configured to use a set of local contexts and documents (contained in a particular directory or repository) in order to resolve the query in two ways:

- Semantic search: uses for each context a matching algorithm [17] that tries to find a correlation between a provider's context and the query focus. In particular the matching algorithm tries to find in the provider's context the focus that has a relevant semantic relation with the one sent by the seeker. The focus that the provider has used for the query resolution and related documents are returned as results. If the focus points to other K-peers, the provider will propagate the query.
- Syntactic search: uses an indexer (or any locally available keyword index) to search for the occurrence of specific keywords into the set of documents owned by the provider.

If the semantic query is composed only from keywords, the provider will use only the syntactic search, otherwise if it is composed only by a focus, the provider will use only the semantic search. If both are available, the final result will be the merge of semantic and syntactic results.

### 3.2 K-Peer Capabilities

Besides playing the above roles, a K-Peer has some capabilities such as create or join a K-Federation, manage contexts, and use a distributed discovery mechanism, a query propagation functionality and a learning algorithm. All these capabilities will be explained in the following paragraphs.

#### 3.2.1 K-Federation

A K-Federation is a group of K-Peers that agree to be considered as a sole entity by other K-Peers when these are requesting services (such as semantic search) to the former. In other words, a Seeker can send a query to a K-Federation, and the query will be forwarded to each K-Federation member. As a consequence, the response of the K-Federation is the same as if the query was sent directly to all the members of the K-Federation (even if in the returned result set is specified if the Provider answers as a member of a K-Federation). The K-Federation can be though as a "social" aggregation of K-Peers that display some synergy in terms of content (provide synergic content), quality (certified content) or access policies (certified members).

To become a member of a K-Federation, a K-Peer must provide a K-Federation Service (for example the query resolution service), implement the required K-Federation's protocol (for example the commitment to reply to queries sent to the K-Federation) and observes the K-Federation's membership policy (for example reply only to queries coming from federation members).

#### 3.2.2 Context Management

The context management module allows users to manage contexts in order to search or provide information. Contexts provide a semantic classification of the knowledge and data. At the system level, KEx avoids the need to create huge syntactical indexes and to transform knowledge and information into an homogeneous format; it gives to each K-Peer the autonomy to store and maintain its own information in the way it prefers, providing a tool to enable autonomous semantic classification of content. Multiple views can be built over the local information, each corresponding to different contextual representations and semantic classifications of content. Two components are provided to manage contexts:

1. Context editor: gives to the user the possibility to edit and manage a context. Moreover when the user crates a context she can classify information (documents, addresses of other K-Peers) into the context. In particular the user creates the links between a context's concept and the specific information stored in local repositories (or links to other peers or directly to external information maintained in other K-Peers repositories). This activity (relating context to content) can be also automatically performed by any text mining tool if integrated with the peer system (e.g. able to transform the documents representation schema provided by the mining engine into a context using a standard syntax such as CTXML [12]).
2. Context browser: allows the user to specify a query in order to search information in the network. The user can select a context from a set of available contexts and browse the structure of the selected one in order to focus on a particular contextual topic (focus). The focus represents a set of contextual information that the providing K-Peers can use to interpret the meaning of a query (that can be expressed as a traditional set of keywords) and to compare it with their own contexts.

#### 3.2.3 Discovery

Discovery is a mechanism that allows the user to discover resources in the peer to peer network. The user discovers K-Peers or K-Federations available in the network in order to contact them and send them queries. A K-Peer advertises the existence of resources publishing an XML document (advertisement). In the KEx system two type resources are advertised:

- K-Peers that have a provider service to solve queries. The main elements of the advertisement are a description of a K-Peers contexts and how to contact the K-Peer in order to send a query or retrieve documents.
- K-Federations that are set of K-Peers that have a provider service to solve similar queries. The K-Federation assures that a query sent to a K-Federation is propagated to all active K-Peers that are member of the K-Federation. In this case the main elements of the advertisement are the K-Federation themes, how to contact, how to join.

In order to discover resources in a peer to peer network, a K-Peer sends a discovery request to another known K-Peer or sends a multi-cast request on the network, and receives responses (list of advertisements) that describe the available services and resources. It is possible to specify a searching criteria (such as a keyword or textual expression) that is matched against the contents provided by the advertisement related to each K-Peer or K-Federation description.

#### 3.2.4 Query Propagation

This functionality allows the KEx system to "use" trust and social relations as a mean to find information in highly dynamic, heterogeneous and complex environment. In fact a K-peer can trust not just the "content" knowledge of another peer, but also its "relational" knowledge in terms of the capability to redirect a request to other trusted K-peers. Thus, when a Provider receives a query, it might propagate that query to another Provider it considers "expert" on what it believes is the meaning of the request. In order to decide where to propagate a query a K-Peer has two possibilities:

– a "proximity" criteria: the query will be sent to known K-Peers (i. e. by using the discovery functionality) and selection will be done according to some quantitative criteria (number of peers, number of possible re-routings –hops-, etc.); this way K-Peers or providers that are non directly reachable by the Seeker or that have just joined the system, can advertise their presence and contribute to the resolution of the query;
– a semantic criteria: if the Provider computes some matching between a query and concepts in its own context, the query resolution mechanism might look for addresses of other K-Peers that have been associated to the matching concept. Here propagation is done on the base of an explicit trust since the provider defines other peers as experts about the query topic.

The propagation algorithm combines the two possibilities and is based upon a cost function that has the goal to ensure quality and reduce overflow. This goal is achieved privileging semantic re-routing, and restricting the possibility to perform proximity based hops the more semantic based hops are done.

Other parameters and mechanism controlling the scope of the search and prevent the message "flooding" are provided: the seeker can set a time to live (TTL), manually limit the number of hops, store in the query the name of K-Peers that have already received the query, and so on.

#### 3.2.5 Learning

When the matching algorithm finds a semantic correspondence between concepts belonging to different providers contexts, the Seeker can store this information for a future reuse. This information is represented as a semantic "mapping" between concepts, and can be used in three ways:

1. when the K-Peer receives a query from another seeker, it can reuse the corresponding stored mapping to facilitate (or eventually don't perform) the matching algorithm;
2. a Provider can use the existing mapping to forward the query to other K-Peers that present a semantic relation with the topic of the query (see 3.2.4);
3. in order to send the query to a trusted set of providers, the Seeker could search into the mapping relations in order to select those that have been recognized as "experts" on a particular topic thanks to previous interactions. Such a system could be viewed as a sort of semantic "bookmark" of preferred providers.

## 4 Conclusions and Research Issues

In this article we argued that technological architectures, when dealing with processes deeply characterized by human communication, must be consistent with the social architecture of the process itself. In particular, in the domain of KM, technology must embody a principle of distribution that is intrinsic to the nature of organizational cognition.

As a consequence, we propose some requirements for designing and implementing technological architectures for KM, based on P2P technologies. Moreover, a number of research issues emerge in order to map aspects of distributed cognition into technological requirements. Here we propose some of these:

- social discovery and propagation: in order to find knowledge, people need to discover who is reachable and available to answer to a request. On the one hand, broadcasting messages generates communication overflow, on the other talking just to physically available neighbors reduces the potential of a distributed network. A third option could be for a seeker to ask his neighbors who they trust on a topic and, among these, who is currently available. Here the question is about the social mechanisms through which people find, on the base of trust and recommendation, other people to involve in a conversation. A similar approach could be used in order to support the propagation of information requests. In this work we propose an hypothesis of social/semantic discovery and propagation. Nonetheless further research needs to be done in order to match theoretical and practical analysis of informal information flows and to map new models of bottom-up coordination mechanisms into distributed technologies.
- building communities: if we consider communities as network of people that, to some extent, tend to share a common interpretative schema, mechanisms will be needed to support the bottom-up emergence of semantic similarities across interacting KNs. Through this process, that could be based on meaning negotiation protocols, people could discover and form virtual communities, and within organizations, managers might monitor the evolving trajectories of informal cogni-

tive networks. Than such networks can be viewed as potential neighborhoods to support social discovery and propagation.

## References


1. S. Bergamaschi, F. Guerra, and M. Vincini. A Data Integration Framework for e-Commerce Product Classification *Proceedings of 1st International Semantic Web Conference (ISWC'2002), Ian Horrocks James Hendler Eds., Springer Press,* 2002.
2. D. Bertolini, P. Busetta, A. Molani, M. Nori, A. Perini. Peer-to-peer, agents, and knowledge management: a design framework *IRST Technical Report No. 0207-15, Istituto Trentino di Cultura,* 2002.
3. R. J. Boland, R. V. Tenkasi, and D. Te'eni. Designing Information Technology to Support Distributed Cognition, *Organizational Science,* 5, 3 (August), 1994.
4. R. J. Boland, and R. V. Tenkasi Perspective Making and Perspective Taking in Communities of Knowing, *Organization Science,* 6, 4 (July-August), 1995.
5. M. Bonifacio, P. Bouquet, and A. Manzardo. A distributed intelligence paradigm for knowledge management. In *AAAI Spring Symposium Series 2000 on Bringing Knowledge to Business Processes. AAAI,* 2000.
6. M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. Managerial and technological implications. *Informatik - Informatique,* 1/2002.
7. M. Bonifacio, P. Bouquet, and R. Cuel. The Role of Classification(s) in Distributed Knowledge Management. To appear in the Proceedings of 6th International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies (KES'2002), Special Session on Classification. September 16-18, 2002. Crema (Italy). IOS Press.
8. M. Bonifacio, P. Bouquet, and R. Cuel. KNs: the building blocks of a distributed approach to KM *Journal for Universal Computer Science,* 2002.
9. P. Bouquet. *Contesti e ragionamento contestuale. Il ruolo del contesto in una teoria della rappresentazione della conoscenza*. Pantograph, Genova (Italy), 1998.
10. G. C. Bowker and S. L. Star. *Sorting things out: classification and its consequences*. MIT Press., 1999.
11. J. S. Brown and P. Duguid. Organizational learning and Communities of Practices: Toward a Unified View of Working, Learning and Innovation. *Organization Science,* 2(1), 1991.
12. P. Bouquet, A. Donà, L. Serafini, and S. Zanobini Contextualized local ontologies specification via CTXML To appear in the working notes of the AAAI-02 workshop on Meaning Negotiation. Edmonton (Canada). July 28, 2002.
13. U.M. Borghoff and R. Pareschi *Information Technology for KM* Springer, Berlin, 1999.
14. J. Dinsmore. *Partitioned Representations*. Kluwer Academic Publishers, 1991.
15. G. Fauconnier. Conceptual Integration Networks. *Cognitive Science*, 22(2):133-187, 1998.
16. C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence,* 127(2):221-259, April 2001.
17. B. Magnini, L. Serafini, and M. Speranza. Linguistic based matching of local ontologies. To appear in the working notes of the *AAAI-02 workshop on Meaning Negotiation*. Edmonton (Canada). July 28, 2002
18. J. Lave and E. Wenger. *Situated Learning: Legitimate Peripheral Participation*. New York, Cambridge University Press. 1990.

19. J. Madhavan, P. Bernstein, P. Domingos, and A. Haley, Representing and Reasoning about Mappings between Domain Models at the *Eighteenth National Conference on Artificial Intelligence (AAAI'2002)*, Edmonton, Canada.
20. J. McCarthy. Notes on Formalizing Context. In *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pages 555-560, Chambery, France, 1993.
21. W.J. Orlikowski and D.C. Gash. Technological Frames: Making Sense of Information Technology in Organization. *ACM Transactions on Information Systems*, 1994 Vol.12/2, April, pages 174-207.
22. W. J. Orlikowsy, and D. Robey Information Technology and the Structuring of Organizations *Information Systems Research*, 2, 2 (June), 1991.
23. S. Prasad, Y. Peng, and T. Finin. A Tool For Mapping Between Two Ontologies Using Explicit Information. Notes of the *AAAI-02 Workshop on Meaning Negotiation*, Edmonton, Canada, August 2002.
24. R.L. Purvis, V. Sambamurthy, and R.W. Zmud. The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation. *Organization Science*, 2001, Vol 12, No 2, March-April, pages 117-135.
25. W.V. Quine. Propositions. In *Ontological Relativity and other Essays*. Columbia University Press, New York, 1969.
26. L. Sproull and S. Kiesler. Reducing Social Context Cues: Electronic Mail in Organizational Communication. Irene Greif (Eds.), *Computer Supported Cooperative Work: A Book of Readings*, San Mateo CA, Morgan Kaufman Publishers, 1988.
27. E. Wenger. *Communities of Practice. Learning, Meaning and Identity*, Cambridge University Press, 1998.