A Peer-to-Peer Architecture for Distributed Knowledge Management

Matteo Bonifacio
University of Trento,
via Sommarive, 18 38050 Trento (Italy)
Phone number +39 0461 314452
bonifaci@science.unitn.it

Roberta Cuel
University of Trento, Italy
via Inama, 5 38100 Trento (Italy)
Phone number +39 0461 882152
Fax number +39 0461 882124
rcuel@cs.unitn.it

Gianluca Mameli
ITC-IRST, Trento, Italy
via Sommarive, 18 38050 Trento (Italy)
Phone number +39 0461 314331
mameli@irst.itc.it

Michele Nori
ITC-IRST, Trento, Italy
via Sommarive, 18 38050 Trento (Italy)
Phone number +39 0461 314331
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), we argue that technological and social architectures must be reciprocally consistent and, 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), 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 Knowledge Nodes, is proposed.

Keywords: Knowledge management, Distributed Knowledge Management, Distributed and Federated Systems, Organizational Cognition, Peer to Peer Architecture

1 Introduction: 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 money and time in KM projects, whose typical outcome is the creation of computer based systems composed by a Knowledge Base (KB) and an Enterprize 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 exchanging 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 [17], [9], [16], [15], [14], [19]). They argue that knowledge is the result of different perspectival 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 performance, and the need of meeting different perspectives, in order to sustain innovation [4]. Thus, the management of knowledge should be based on supporting and balancing 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 inter-operation.

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).

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

2 Technological and Social Architectures

As already underlined in [6], 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 [18]) 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 underly some of the main features characterizing a KM system that assumes 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 [21], [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.

According to these organizational aspects centralized KM system architectures must:

- create and enable communication within formal and informal groups and communities. On-line interaction processes in terms of social/contextual cues [20] 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 Enterprize 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 this paragraph we will propose the main requirements of a DKM architecture (see Figure 1) which, in the next section, will be framed in the specific context of a knowledge exchange application developed within a Bank.

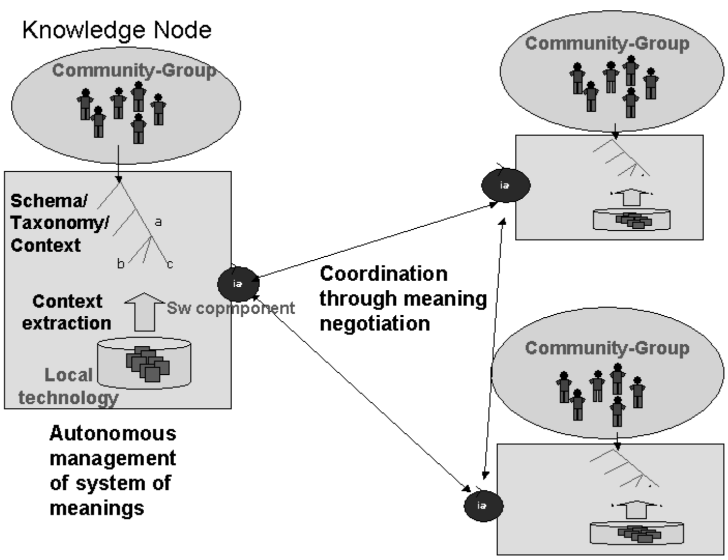

Fig 1. A DKM technological architecture

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 Knowledge Nodes (KNs) considered in [8] as the building blocks of a DKM architecture.

A Knowledge Node (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), technology 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 two logical elements that are relevant in order to support coordination processes among KNs:

- context: a representation schema of the system of meanings adopted within each KN. As proposed in [12], context is 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.

- connection table: a system that, using pointers such as URIs, allow us to relate content to an appropriate interpretative context. For example a connection table is the logical connection between a document and the label of the taxonomy under which the document is categorized.

Contexts and connection tables 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. 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. 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. Such a process of meaning assessment and comparison across different categorization schemas [1] [22] is here named "meaning negotiation process" (for an algorithm of meaning negotiation see [13]).

2.2.3 Cooperation

Given that KNs are semantically autonomous but coordinated through the capacity of negotiating meaning, some other minimal organizational elements are needed in order to design a sustainable distributed architecture. In general, these organizational features are necessary to avoid communication overflow while ensuring effectiveness, and configure a network of KN as a distributed cooperative system. Here we briefly propose some of these abilities:

- Federate: KNs should be able to federate creating groupings 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). Federations can simplify the 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 as presumably a certified quality.

- Discover: KNs should be able to find each others having the opportunity to discover who is available, what type of services it's committed to deliver, 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. In particular since "knowing who knows" is a knowledge itself, then, give a DKM approach, it is and must be distributed like any other form of 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, he 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, that emerges through interaction, can be also fruitfully used to redirect 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 Logical Architecture for a DKM System

In this section we briefly show how the non-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[1]. For more details see [8].

In the DKM approach, a great emphasis is given to autonomy and coordination aspects, so that every KN can access to external information and can allow others to access to its knowledge. As we already said, this meaning negotiation is mediated through the use of contexts [12] that are a personal explicit representation, typically in form of classification, of the KN's system of meanings. Compared to these aspects a peer-to-peer system (see [2]) can be depicted with the following capabilities:

- being autonomous: every peer can decide how to provide a service;
- being dynamic: peers and resources can be added or removed at any time;
- being decentralized: the community of peers is able to achieve its goal independently from any specific member or component;
- being composed by entities that have equal capabilities: every member must provide resources or services, as well as has a right to access the others'resources and services.

These features of a peer-to-peer system seem to match the spirit and the main non-functional aspects of a KN in a DKM application, suggesting this architectural solution as a logical choice. In particular autonomy is guaranteed by each peer which manage its knowledge in a personal way, using contexts that better suit its document classification. Coordination is guaranteed through enabling peers to collaborate with each other, and exchange information, without

[1] 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.

imposing a common interpretative schema, but through a meaning negotiation service that maps concepts among different system of meanings and contexts.

From an implementation point of view, we focus on *JXTA* [2], a set of open, generalized peer-to-peer protocols that allow devices to communicate and collaborate through a connecting network. This P2P framework provides also a set of protocols and functionality 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 common set of rules to publish, share and access "resources" (like services, data or applications), and communicate among each others. 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 peer-to-peer 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). The flexibility of this framework allows to design distributed systems that cover all the requirements of a DKM application, using the JXTA P2P capabilities, completed and enhanced through the implementation of user-defined services.

In the next section we'll describe the architectural solution adopted in a distributed knowledge exchange system of the Bank; and in this project we combine the P2P paradigm (characterizing a KN network as a network of distributed peers) and JXTA as an implementation infrastructure.

4 Knowledge Exchange System Architecture

The Knowledge Exchange system (KEx) has the goal to provide users with a new mechanism to search and provide information implementing an innovative exchanging approach based on the concept of semantic inter-operability. The KEx system gives to the user the capability to exchange information (searching and providing) in a distributed network where knowledge is maintained and managed by different (possibly independent) actors which use different systems of meanings. The DKM system is based on the Peer-To-Peer technology and is represented as a network of peers. Every peer has the same main systems functionalities that are provided by two principal roles: the *provider* role and the *seeker* role. Each peer can act as a provider, as a seeker, or can play simultaneously both roles (see Figure 4).

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

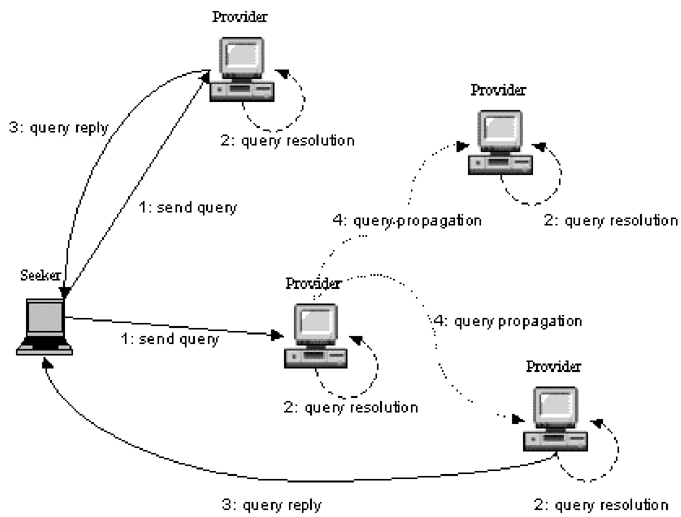

Fig 2. The KEx system: interaction between Seeker and Provider roles.

4.1 Peer Roles

4.1.1 Seeker

The seeker is one of the two principal roles in the KEx system. It allows the user to interact with other peers and federations in order to search information. The seeker supports the user in the definition of the query using the context browser in order to search for information in the network. 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 to the user, the discovery mechanism to find resources (peers or federations) in order to perform the query. The user decides to send the query to some of the available peers and federations. When the user submits the query, the seeker activates a session (there can be only one active session for a seeker). A session is associated to a query. In a session, a seeker can receive asynchronously several replies to his query coming from these providers that solved the query and contacted the seeker. The session is in charge to perform the query resolution using the meaning negotiation protocol. The active seeker session collects the results. The results returned to the user are composed by the aggregation of all the results received from the different answering providers; each result is made up of a list of document descriptors (name of the document, short description, and so on). Each result is proposed to the user with the indication of the part of context (focus) that the provider has used for the query resolution. This relationship between contexts can be used as a learning opportunity (see 4.2.5) that the seeker can store and reuse for other queries. Finally the seeker allows the user to access to the peer download service. If the user finds one or more interesting documents it can contact the peer that has the document and, if possible, download it.

4.1.2 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 peer receives a semantic query (composed by keywords and one focus), it instantiates a provider configured to use a set of contexts and some documents (a particular directory) in order to resolve the query in two ways:

Semantic search: using for each context a matching algorithm [13] that tries to find a correlation between a providers context and the querys focus. In particular the matching algorithm tries to find a focus in the providers context that has a relation with the querys focus. The focus that the provider has used for the query resolution with the matching algorithm and the documents linked with the focus are part of the result. If there are links to external information, the provider will propagate the query to those peers that own the information.

Syntactic search: using an indexer 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 from focus, the provider will use only the semantic search. If both are available, the final result will be the merge of the semantic and syntactic results that the provider will send to the requesting seeker.

4.2 Peer Capabilities

A peer is the software component that plays the two main roles and provides some capabilities such as create or join a 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.

4.2.1 Federation

A *Federation* is a group of K-Peer that agree to appear as a unique entity in respect to peers that perform a semantic search. In other words, a Seeker can send a query to a Federation, and the query will be forwarded to each federation member. As a consequence, the response of the Federation to a query sent by a

Seeker, is the same as if the query was sent directly to all the members of the Federation (even if in the returned result set is specified if the Provider answers as a member of a Federation). The Federation can be though as a "social" aggregation of peers that display some synergy in terms of content (provide synergic content), quality (certify content) or access policies (certify members).

To become a member of a Federation, a K-Peer must provide a Federation Service (quite similar to that required by the Provider role) that implements the required Federation's protocol (reply to query sent to the Federation) and observes the Federation's membership policy.

4.2.2 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. KEx avoids the need to create huge syntactical indexes and to transform knowledge and information in an homogeneous format; it gives to each 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. There are two component to manage contexts:

Context editor: gives to the user the possibility to edit and manage a context. Moreover when the user crates a context he can classify information (documents, address of other peers) into the context. In particular the user creates the links among a context's concept and the specific information stored in the local directories (or links to external information maintained in other peers).

Context browser: allows the user to specify a query in order to search for 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 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 other contexts.

4.2.3 Discovery

The discovery is a mechanism that allows the user to discover resources in the peer to peer network. The user discovers peers or federations available in the network in order to contact them and send them the queries. A peer advertises the existence of resources publishing a XML document (advertisement). In the KEx system we advertise two type resources:

Peers have a provider service to solve queries. The main elements of the advertisement are a description of a peers contexts and how to contact the peer in order to send it a query or retrieve documents.

Federations are set of peers that have a provider service to solve similar queries. The federation assures that a query sent to a federation is propagated to all active peers that are member of the federation. In this case the main elements of the advertisement are the federation themes, how to contact, how to join.

In order to discover resources in a peer to peer network, a peer sends a discovery request to another known 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 peer or federation description.

4.2.4 Query Propagation

This functionality allows the KEx system to satisfy the requirements of a better and more specific result set and, contemporaneous, showing a adaptability to a high dynamic environment.

When a Provider receives a query, it might propagate that query to another Provider it considers "expert" or "fair" to answer the query. To decide to which forward a query a peer has two possibility:

- a "proximity" criteria: the query will be sent to a known peer (i. e. with the discovery functionality); in this way peers or providers that are non directly reachable from the Seeker or have just joined the system can advertise their presence and contribute to the resolution of the query.

- a semantic criteria: if the Provider compute some matching between a query and concepts in its own context, the system should look for addresses of other peers that the K-Owner has associated to that concept as expert in the query's topic. If one o more address are found, the Provider forward the query to those peer, so the query will be propagated only to relevant Provider and the answers should be more accurate.

The propagation algorithm is based upon a cost function allows choosing peers that are regarded to provide more relevant information (assigning a higher value to peer discovered through the semantic criteria than peers reached through the proximity criteria).

Obviously, there are several parameters and mechanism controlling the scope of the search and prevent the message "flooding": set a time to live (TTL), limit the number of hop, store in the query the name of peers that have still received the query, and so on.

4.2.5 Learning

When the matching algorithm find a semantical correspondence between concepts of different context, the Provider can store this information for a future reuse. These informations are represented as semantical "mapping" between concepts, and can be used in three way:

- when the K-Peer receive a query from a seeker can reuse the corresponding stored mapping to facilitate (or eventually don't perform) the matching algorithm;
- a Provider can use the existing mapping to forward the query to other peers that present a semantic relation with the topic of the query (see 4.2.4);
- the Seeker could search into the mapping relations in order to suggest to the user a set of Provider which has already had a previous interaction and has been considered qualified respect the semantic meaning of the concept selected as query.

With this mechanism the K-Peer network will define and increase the number and quality of the semantic relation between its members, so that it becomes a dynamic web of knowledge link.

5 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 peer to peer 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.

building communities: if we consider communities as network of people that, to some extent, tend to share a common interpretative schema [4], 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 cognitive networks. Then, 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, 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 *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 approch to KM *Journal for Universal Cumputer 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] 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

[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] J. McCarthy. Notes on Formalizing Context. In *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pages 555–560, Chambery, France, 1993.

[18] W. J. Orlikowsy, and D. Robey Information Technology and the Structuring of Organizations *Information Systems Research, 2, 2 (June)*, 1991.

[19] W. V. Quine. Propositions. In *Ontological Relativity and other Essays*. Columbia University Press, New York, 1969.

[20] 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 Kaufmann Publishers.*

[21] E. Wenger Communities of Practice. Learning, Meaning and Identity *Cambridge University Press.*

[22] S. Prasad, Y. Peng, and T. Finin. A Tool For Mapping Between Two Ontologies Using Explicit Information authors. *Submitted to the AAAI Workshop on Meaning Negotiation*, August 2002.